

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи



Баженова Ирина Васильевна

**МЕТОДИКА ПРОЕКТИВНО-РЕКУРСИВНОГО ОБУЧЕНИЯ
ПРОГРАММИРОВАНИЮ
СТУДЕНТОВ МАТЕМАТИЧЕСКИХ НАПРАВЛЕНИЙ ПОДГОТОВКИ**

13.00.02 – Теория и методика обучения и воспитания
(информатика, уровень профессионального образования)

Диссертация на соискание учёной степени
кандидата педагогических наук

Научный руководитель:
доктор педагогических наук,
профессор
Пак Николай Инсебович

Красноярск – 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1 ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ОБНОВЛЕНИЯ МЕТОДИЧЕСКОЙ СИСТЕМЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ СТУДЕНТОВ МАТЕМАТИЧЕСКИХ НАПРАВЛЕНИЙ ПОДГОТОВКИ.....	17
1.1. Анализ проблемной области, существующих подходов и современных требований к обучению программированию.....	17
1.2. Возможности проективно-рекурсивной стратегии в обучении программированию	31
1.3. Повышение эффективности обучения программированию за счет использования когнитивных технологий.....	44
1.4. Моделирование результативно-оценочного компонента методической системы обучения программированию студентов-математиков	57
ВЫВОДЫ ПО ГЛАВЕ 1	69
ГЛАВА 2 ПРОЕКТИРОВАНИЕ МЕТОДИКИ ОБУЧЕНИЯ СТУДЕНТОВ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ПРОЕКТИВНО-РЕКУРСИВНОЙ СТРАТЕГИИ И КОГНИТИВНЫХ ТЕХНОЛОГИЙ.....	71
2.1. Структурно-логическая модель обучения программированию будущих бакалавров математических направлений подготовки	71
2.2. Процессуальная модель обучения программированию студентов- математиков.....	96
2.3. Планирование, проведение и результаты педагогического эксперимента.....	109

ВЫВОДЫ ПО ГЛАВЕ 2.....	123
ЗАКЛЮЧЕНИЕ	127
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	131
ПРИЛОЖЕНИЯ	152

ВВЕДЕНИЕ

Актуальность исследования. В настоящее время перед высшей школой стоит задача эффективной, практико-ориентированной и системной подготовки специалистов, социально и профессионально адаптированных к вызовам информационного общества и требованиям работодателей, особенно в высокотехнологичных и наукоёмких сферах труда.

Важность этой задачи отмечена во многих государственных документах, например, в дорожной карте «Развитие отрасли информационных технологий на 2013-2018 гг.» указано на приоритетное «развитие человеческого капитала за счет повышения уровня образования в области информационных технологий, включая развитие физико-математического и профильного образования» [115].

Базовый курс программирования традиционно включается в учебные планы подготовки бакалавров математических направлений. С введением новых Федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО), ориентированных на компетентностную модель подготовки и оценивания результатов обучения выпускников вуза, значимость данного курса повышается в силу того, что он способствует формированию у них профессиональных и общекультурных компетенций. Так, в ФГОС ВПО по направлению подготовки 01.03.02 «Прикладная математика и информатика» указана «способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения» как базовый структурный элемент компетентностной модели выпускника (ОПК-3, ПК-7), а «изучение и разработка языков программирования, алгоритмов, библиотек и пакетов программ» является профессиональной задачей, которую выпускник готов решать [118]. Важным аспектом обучения программированию является то, что программирование, как системная интеллектуальная деятельность, оказывает существенное влияние на развитие когнитивных способностей обучающихся, в том числе алгоритмического и системного мышления. Академик А.П. Ершов указывал на

такие качества программиста, как «способность планировать собственные действия, вырабатывать общие правила и способ их применения к конкретной ситуации, организовывать эти правила в осознанную и выразимую структуру» [58, с.99].

Вопросы теории и практики обучения программированию в школьном и вузовском образовании были поставлены в трудах многих видных отечественных и зарубежных ученых (А.П. Ершов, С. Пейперт, Э. Соловей, Л.А. Робертсон, В.М. Монахов, А.А. Кузнецов, А. Г. Гейн, И.Г. Семакин, Ю.А. Первин, Г.А. Звенигородский, Н.Д. Угринович, М.В. Швецкий, В.Е. Жужжалов, А.Р. Есаян, А. Шень и др.). Вместе с тем, следует отметить, что, в условиях смены парадигмы высшего образования, реализации новых ФГОС ВПО и профессиональных стандартов, традиционные методические системы обучения программированию не отвечают требованию стать метакогнитивным средством формирования профессиональных и общекультурных компетенций. Ряд исследований и учебных пособий посвящены методике обучения программированию студентов-математиков (А.Г. Кушниренко, Г.В. Лебедев, С.С. Лавров, И.А. Кудрявцева, А.Е. Люлькин, С.А. Григорьев, О.В. Смирнова, А.Н. Салангина и др.) [49], [78], [80], [83], [94], [144], [155]. Тем не менее, следует отметить, что основное внимание в них уделено содержательному компоненту методической системы – отбору специфических математико-ориентированных задач, что, несомненно, имеет практическую ценность, но недостаточно для полноценной методической системы обучения программированию в современных условиях. Как указывает И.В. Роберт, в условиях информатизации образования одним из направлений развития дидактики становится «разработка методических систем обучения, реализующих идеи конвергенции педагогической науки и наукоемких технологий и ориентированных на изменение структуры представления учебного материала, обеспечивающих отбор изучаемого материала адекватно личностным предпочтениям обучающегося ...» [140, с.118].

Эффективная и продуктивная методика в любой предметной области, позволяющая повысить качество обучения, должна учитывать изменения,

происходящие в самих обучающихся. Современное поколение школьников и студентов сейчас принято называть «сетевым» или «цифровым» поколением. На формирование мышления этого поколения оказывают влияние экспоненциально растущий поток информации, зачастую агрессивная визуальная среда, использование цифровых технологий с самого детства, постоянный доступ в Интернет с помощью компьютеров и мобильных устройств. В психологических исследованиях используется понятие «клиповое» мышление». Это образное наименование реально фиксируемых фактов фрагментарности восприятия и обработки информации, утраты её целостности, ухудшения способности обучаемых к обобщению и систематизации. В этой связи необходимо пересмотреть методы и средства обучения программированию, которые отвечали бы возможностям и интересам современных студентов. Следует ожидать дидактического эффекта от удачных сочетаний традиционных технологий обучения с инновационными, опирающимися в большей степени на когнитивный подход в обучении, целью которого является развитие «всей совокупности умственных способностей и стратегий, делающих успешным процесс обучения и адаптации к новым ситуациям» [4]. Концепция когнитивного обучения в психолого-педагогической теории и практике сформировалась на основе классических работ отечественных и зарубежных ученых: Л.С. Выготского, С.Л. Рубинштейна, Э.Ч. Толмена, Дж.С. Брунера, Р.Ч. Аткинсона, Г.Э. Гарднера, Б.М. Теплова, М.С. Шехтера, Г. С. Альтшуллера, Б.М. Величковского и др.

В процессе обучения программированию первокурсников математических направлений подготовки возникает ряд проблем, обусловленных как объективными, так и субъективными причинами. Некоторые факторы, оказывающие влияние на процесс обучения программированию, способствуют этому процессу (например, совершенствование вычислительной техники и программного обеспечения, разнообразие технологий и платформ программирования), другие имеют негативный характер воздействия (например, сложность предметной области программирования). К тому же, фактическое отсутствие конкурса на математические специальности среди абитуриентов

определяет низкий уровень практических навыков программирования, неразвитые алгоритмическую культуру и системное мышление у многих студентов 1 курса. Серьёзную проблему представляет недостаточная мотивация студентов к учебной деятельности, обусловленная их слабой профессиональной осведомленностью и сложностью как собственно математических дисциплин, так и теории и практики программирования. Существенно повысить мотивацию студентов к обучению программированию, обеспечить качество обучения и достижение планируемых образовательных результатов возможно на основе проективно-рекурсивной стратегии (Н.И. Пак). Суть этой стратегии – в реализации двух базовых принципов. Принцип проективности можно сформулировать как «будущее определяет настоящее», т.е. будущая профессиональная деятельность обучающихся проецируется на настоящий учебный процесс, в ходе которого моделируются условия для решения возможных профессиональных задач с помощью реализации личного проекта. Принцип рекурсивности предполагает, что необходимым компонентом процесса обучения является создание самими обучающимися образовательных ресурсов, которые используются в этом же учебном процессе.

Таким образом, в процессе исследования проблемной области были выявлены следующие **противоречия**:

– на социально-педагогическом уровне: между требованиями современного общества, выраженными в ФГОС ВПО, профессиональных стандартах, к выпускникам математических направлений подготовки, обладающих сформированными компетенциями в области программирования, развитыми когнитивными способностями, и неготовностью традиционных методических систем обучения программированию обеспечить эти требования;

– на научно-педагогическом уровне: между необходимостью обновления методов и средств обучения программированию, учитывающих личностный потенциал и индивидуальные ментальные характеристики каждого студента и недостаточной теоретико-практической базой исследований в области

использования инновационных и когнитивных технологий в обучении студентов программированию;

– на научно-методическом уровне: между возможностью повысить эффективность подготовки студентов в предметной области программирования, развить их алгоритмическое и системное мышление за счет потенциала информационно-коммуникационных и когнитивных технологий, и отсутствием подобных методик обучения программированию в условиях реализации федеральных государственных образовательных стандартов.

Перечисленные противоречия определяют **проблему исследования**: какой должна быть современная методика обучения программированию студентов математических направлений подготовки, обеспечивающая достижение требуемых результатов обучения в условиях использования информационно-коммуникационных и когнитивных технологий и реализации ФГОС ВПО.

Объект исследования – процесс обучения программированию студентов математических направлений подготовки.

Предмет исследования – методика обучения программированию студентов-первокурсников математических направлений подготовки с использованием когнитивных технологий обучения на основе проективно-рекурсивной стратегии.

Цель исследования:

Теоретически обосновать и применить проективно-рекурсивную стратегию к обновлению компонентов методической системы обучения программированию студентов математических направлений подготовки, разработать методику их обучения с использованием когнитивных технологий, обеспечивающую достижение соответствующих ФГОС ВПО результатов обучения.

Постановка и анализ проблемы исследования, указанные объект и предмет исследования, а также поставленная цель позволяют сформулировать **гипотезу исследования**: проектируемые результаты обучения у студентов математических направлений подготовки при обучении программированию будут достигнуты,

если применить методику, опирающуюся на проективно-рекурсивные компоненты методической системы их обучения, в которой:

- целевой компонент определяется путем переориентации курса в средство достижения планируемых результатов обучения на трех уровнях: дисциплинарном, профессиональном и метадисциплинарном;

- содержательный компонент формируется на основе разноуровневых учебных элементов;

- технологический компонент содержит традиционные и когнитивные средства и методы, нацеленные на: а) смещение представления учебной информации от текстовой к образной форме за счет визуализации; б) вовлечение студентов в разработку электронного курса программирования в среде LMS MOODLE, программных продуктов для проектов межвузовской кооперации, ИТ-бизнеса, исследовательских проектов учебной практики; в) разработку и применение информационных ресурсов ментального типа (ментальные карты, концептуальные карты) как средств обучения и средств диагностики и контроля результатов обучения программированию;

- результативно-оценочный компонент содержит комплекс измерителей уровней алгоритмического и системного мышления, критериальных и порядковых оценок качества программных продуктов, оценок дисциплинарных, профессиональных и метадисциплинарных результатов обучения.

Для достижения поставленной цели и проверки сформулированной гипотезы исследования определим следующие **задачи**:

1. Выявить методические особенности, принципы и требования к процессу обучения программированию студентов математических направлений подготовки в современных условиях.

2. Обосновать проективно-рекурсивную стратегию и использование когнитивных технологий обучения студентов программированию.

3. Уточнить понятия «результаты обучения программированию», «алгоритмическое мышление», «системное мышление» в контексте когнитивного подхода.

4. Разработать структурно-логическую и процессуальную модели обучения программированию будущих бакалавров-математиков и на их основе осуществить обновление компонентов методической системы.

5. Разработать методику обучения программированию студентов-математиков с позиций проективно-рекурсивной стратегии обучения и использования когнитивных технологий.

6. Провести педагогический эксперимент по апробации методики проективно-рекурсивного обучения программированию студентов-математиков в реальном учебном процессе и оценить эффективность разработанной методики.

Теоретико-методологические основы исследования:

– концепция деятельностного подхода (Л.С. Выготский, А.Р. Лурия, А.Н. Леонтьев, П.Я. Гальперин, Д.Б. Эльконин, В.В. Давыдов, Л.В. Занков, А.В. Запорожец, Л.И. Божович, В.П. Зинченко и др.);

– концепция системного подхода (Б.Г. Ананьев, Б.Ф. Ломов, В.Г. Афанасьев, И.В. Блауберг, В.Н. Садовский, Э.Г. Юдин, Ю.К. Бабанский, П.И. Пидкасистый, В.В. Краевский, И.Я. Лернер, Г.П. Щедровицкий, А.Г. Асмолов, В.П. Беспалько, З.А. Решетова и др.);

– концепция компетентностного подхода (И.А. Зимняя, В.И. Байденко, Ю.Г. Татур, А.В. Хуторской, А.А. Вербицкий, А.П. Тряпицина, В.В. Сериков, Л.В. Шкерина и др.);

– концепция проективной стратегии в обучении (Н.И. Пак, Г.Л. Ильин, В.С. Безрукова, И.А. Колесникова, И.В. Богомаз и др.);

– фундаментальные труды и научные исследования в области когнитивной психологии (У. Найссер, Ж. Пиаже, Дж. Брунер, Дж. Андерсон, Р. Солсо, Ж.Ф. Ришар, Г.А. Саймон, А. Ньюэлл, Ф.Ч. Бартлетт, Ф. Джонсон-Лэйрд,

Б.М. Величковский, Л.М. Веккер, М.А. Холодная, К.А. Абульханова-Славская, К.В. Анохин, Т.В. Черниговская и др.);

– исследования в области теории, практики и методики обучения программированию (Д. Кнут, Э. Дейкстра, Н. Вирт, Б. Страуструп, А.П. Ершов, Г.С. Цейтин, А.Г. Гейн, А.Г. Кушниренко, И.Г. Семакин, Е.К. Хеннер, Н.Д. Угринович, М.П. Лапчик, К.Ю. Поляков, Е.А. Ерёмин, А.П. Шестаков, В.Е. Жужжалов, С.М. Окулов, Л.З. Шауцукова, А.А. Дуванов и др.).

Для проверки выдвинутой гипотезы и решения поставленных задач в процессе работы над диссертационным исследованием использовались следующие методы:

– теоретические (изучение, анализ и обобщение научной, методической, психолого-педагогической и специальной литературы в проблемной области; сравнительный анализ Федеральных государственных стандартов высшего профессионального образования 3 поколения и версии ФГОС ВПО 3+, учебных пособий и методических разработок по программированию; изучение материалов научно-практических конференций и материалов по теме исследования, представленных в сети Интернет; обобщение и систематизация педагогического опыта и научных положений по теме исследования; моделирование и педагогическое проектирование;

– эмпирические (наблюдение, опрос, анкетирование, тестирование, педагогический эксперимент, проектирование, реализация и апробация контента электронного курса, визуализация результатов эксперимента);

– статистические методы (количественный и качественный анализ результатов).

Организация и этапы исследования. Экспериментальная работа по теме исследования проводилась с 2011 по 2015 годы на базе Института математики и фундаментальной информатики Сибирского федерального университета (ИМиФИ СФУ). В педагогическом эксперименте участвовали 124 студента 1-го курса направлений подготовки 01.03.01 «Математика» и 01.03.02 «Прикладная математика и информатика», обучавшихся по дисциплинам «Основы

информатики», «Информатика и программирование», «Языки и методы программирования».

Этапы проведения эксперимента:

Первый этап (2011-2012 гг.) – концептуально-констатирующий. Включал в себя изучение предметной области, анализ литературы по теме исследования, проведение констатирующего эксперимента для выявления проблемы, целей, задач и методов исследования.

Второй этап (2012-2014 гг.) – поисково-формирующий: уточнение и корректировка теоретического обоснования методики, проектирование электронного обучающего курса, проведение формирующего эксперимента.

Третий этап (2014-2015 гг.) – заключительный: анализ и обобщение результатов педагогического эксперимента, формулирование основных положений исследования, оформление диссертации.

Научная новизна исследования заключается в следующем:

1. Выполнено обоснование, моделирование и обновление компонентов методической системы обучения студентов программированию с позиции проективно-рекурсивной стратегии, обеспечивающее:

- индивидуализацию обучения;
- высокую мотивацию студентов к учебной и исследовательской деятельности;
- ответственность студентов за собственные результаты обучения и продукты учебной и проектной деятельности;
- профессионально-ориентированное обучение.

2. На этой основе теоретически обоснована, разработана и экспериментально апробирована в реальном учебном процессе методика проективно-рекурсивного обучения программированию студентов математических направлений подготовки классических университетов, которая обеспечивает достижение результатов обучения:

- повышение уровней алгоритмического и системного мышления за счет использования когнитивных технологий;

- высокий уровень обученности программированию в виде знаний, понимания и умений в предметной области.

Теоретическая значимость исследования:

- уточнены понятия «алгоритмическое мышление», «системное мышление» на основе информационного и когнитивного подходов в современных условиях;
- предложена модель планируемых результатов обучения программированию студентов-математиков;
- сформулированы принципы отбора содержания методической системы на основе проективно-рекурсивной стратегии и когнитивных технологий обучения;
- с учетом данных принципов построена структурно-логическая и процессуальная модели обучения базовому курсу программирования студентов математических направлений подготовки;
- определены критерии оценки уровня сформированности алгоритмического и системного мышления и освоения базового курса программирования на основе предложенной модели результатов обучения программированию и компетентностного подхода.

Практическая значимость исследования:

- разработан и внедрен в реальный учебный процесс электронный обучающий курс «Программирование на языке C/C++» для студентов 1 курса Института математики и фундаментальной информатики СФУ. Курс разработан в системе дистанционного обучения Moodle, доступ по учетным записям сервисов СФУ после авторизации по ссылке: <http://study.sfu-kras.ru/course/view.php?id=557>;
- предложенные методические приемы и формы, разработанные визуальные средства обучения программированию могут быть использованы во вводных и базовых курсах программирования по другим направлениям подготовки бакалавров.

Достоверность и обоснованность результатов и выводов диссертационного исследования обеспечиваются теоретико-методологическими основами исследования, опорой на современные научные достижения в области педагогики и когнитивной науки, анализом и обобщением педагогического опыта

преподавателей программирования, соответствием теоретических и эмпирических методов исследования поставленным целям и задачам, апробацией результатов исследования в реальном учебном процессе.

Личный вклад соискателя в получении результатов, изложенных в диссертации, состоит в самостоятельной работе по исследованию проблемной области, моделированию обновленных компонентов методической системы обучения программированию, разработке и апробации в реальном учебном процессе методики проективно-рекурсивного обучения программированию с использованием когнитивных технологий студентов-математиков, проектированию и реализации электронного обучающего курса «Программирование на C/C++», который используется для обучения первокурсников Института математики и фундаментальной информатики СФУ с 2012 г. по настоящее время. В процессе диссертационного исследования и реализации предложенной методики были опубликованы полученные результаты, в том числе, в журналах из перечня ВАК, разработаны и реализованы студенческие проекты под руководством автора в рамках межвузовской кооперации студентов и преподавателей разнопрофильных вузов.

Положения, выносимые на защиту:

1. Обновление компонентов методической системы обучения студентов программированию с целью достижения планируемых результатов обучения и удовлетворения социальным и профессиональным требованиям к выпускникам математических направлений подготовки целесообразно осуществлять с позиций проективно-рекурсивной стратегии обучения.

2. Проективно-рекурсивные содержательный и технологический компоненты методической системы обучения студентов программированию, включающие традиционные и когнитивные средства и методы, обеспечивают:

- индивидуализацию процесса обучения;
- высокую мотивацию студентов к обучению;
- ответственность за собственные результаты обучения и продукты учебной деятельности;

– профессионально-ориентированное обучение.

3. Реализация в учебном процессе методики проективно-рекурсивного обучения программированию студентов математических направлений подготовки, использующая информационно-коммуникационные и когнитивные технологии, обеспечивает достижение планируемых результатов обучения.

Апробация результатов исследования. Материалы и результаты исследования обсуждались на межвузовском научно-исследовательском семинаре-вебинаре «Информационные технологии и открытое образование» на базе КГПУ им. В.П. Астафьева, на научно-методических семинарах базовой кафедры вычислительных и информационных технологий СФУ, отражены в публикациях и выступлениях на международных научных и научно-практических конференциях («Информационные ресурсы в образовании» – Нижневартовск, 2012 г.; «Современные материалы, техника и технология» – Курск, 2012 г.; «Человек, семья и общество: история и перспективы развития» – Красноярск, 2013 г.; «Решетневские чтения» – Красноярск, 2013 , 2014 гг.; «Фундаментальная информатика, информационные технологии и системы управления: реалии и перспективы. ФИТМ-2014» – Красноярск, 2014 г.), на Всероссийских конференциях с международным участием («Открытое образование: опыт, проблемы, перспективы» – Красноярск, 2011 г., «Информатика и информационные технологии» – Челябинск, 2013 г.), на Всероссийской научно-практической конференции «Интегрированная система профессионального образования: проблемы и пути развития» – Красноярск, 2012 г. По теме исследования опубликовано 14 работ ([6]–[19]), в том числе 4 статьи в изданиях, рекомендованных ВАК, и опубликованы 3 работы, выполненные студентами под руководством автора [39], [46], [75].

Результаты диссертационного исследования внедрены в учебный процесс Института математики и фундаментальной информатики Сибирского федерального университета в виде электронного обучающего курса. Данный курс находится в открытом доступе, им могут воспользоваться студенты других направлений подготовки и преподаватели.

Структура диссертации. Работа состоит из введения, двух глав, которые включают семь параграфов, заключения, библиографического списка и приложений. Текст диссертации содержит 30 рисунков, 7 таблиц.

ГЛАВА 1

ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ОБНОВЛЕНИЯ МЕТОДИЧЕСКОЙ СИСТЕМЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ СТУДЕНТОВ МАТЕМАТИЧЕСКИХ НАПРАВЛЕНИЙ ПОДГОТОВКИ

1.1. Анализ проблемной области, существующих подходов и современных требований к обучению программированию

Курс программирования играет важную роль в подготовке будущих бакалавров-математиков, хотя в учебных планах по направлению подготовки 01.03.01 «Математика» он не является базовой профессиональной дисциплиной. Очевидно, что этот курс оказывает значительное влияние на развитие профессиональных качеств будущего математика. Умения и навыки, приобретенные в процессе обучения программированию, выступают в роли эффективного инструментального средства для решения задач в последующей научно-исследовательской, производственно-технологической или преподавательской деятельности бакалавра-математика. Выдающийся теоретик программирования Д. Кнут так говорил о связи математики и программирования: «Составление вычислительной программы из некоторого множества базисных команд очень сходно с построением математического доказательства, исходя из заданного множества аксиом» [74, с.13].

Необходимость учета многочисленных факторов, влияющих на процесс обучения программированию и разработку соответствующей методики, требует их систематизации. Кроме проблем, непосредственно относящихся к области программирования и вопросов обучения программированию, предварительно следует сказать о проблемах математического образования в целом, поскольку это имеет отношение к рассматриваемой проблеме. Эти проблемы были обобщены в Концепции развития математического образования в Российской Федерации [110], где указано, что они имеют:

а) мотивационный характер (низкая учебная мотивация школьников и студентов, обусловленная общественной недооценкой значимости математического образования);

б) содержательный характер (устаревшее, формальное и оторванное от жизни содержание математического образования на всех уровнях образования);

в) кадровый характер (подготовка студентов по направлениям математических и педагогических специальностей не способствует их интеллектуальному росту и требованиям педагогической деятельности в образовательных организациях).

Далее приведена таблица 1, в которой сделана попытка классификации наиболее существенных, на наш взгляд, факторов, влияющих на процесс обучения программированию будущих бакалавров-математиков.

Таблица 1.

Факторы влияния на обучение программированию студентов-математиков

	описание фактора		характер влияния (позитивный, негативный)
объективные факторы	общие	Интенсивное развитие вычислительной техники и программного обеспечения	+ –
		Разнообразие технологий и платформ программирования	+ –
		Сосуществование разных парадигм программирования (необходимость выбора базовой парадигмы при обучении)	+ –
		Наличие интегрированных сред разработки в современных системах программирования	+
		Сложность изучения теории и практики программирования	–

	специфические	Математические основы программирования	+
		Пересечение тезаурусов математики и программирования	–
		Несогласованность курса программирования с математическими дисциплинами в учебных планах	–
субъективные факторы	общие ментальные характеристики обучающихся	Постоянный доступ в Интернет с помощью компьютеров и мобильных устройств	+ –
		Улучшение визуального восприятия и ориентации в пространстве	+
		Способность к быстрому принятию решений	+
		Способность к сетевой коммуникации и взаимодействию	+
		«Клипное» мышление	–
		Неготовность к самостоятельной учебной деятельности	–
	специфические характеристики обучающихся	Слабая профессиональная осведомлённость → низкая мотивация к обучению	–
		Разнородность студенческой группы в опыте практического программирования	–
		Неразвитые алгоритмическая культура и системное мышление у многих студентов-первокурсников	–

Дадим пояснения к таблице. Специфическими факторами названы те, которые имеют отношение к обучению программированию студентов-математиков, общими – характерные для процесса обучения в целом. Ряд факторов оказывает одновременно и позитивное, и негативное влияние, что помечено в таблице символами «+» и «–». Так, интенсивное развитие вычислительной техники и программного обеспечения – важный фактор

технического прогресса, но, в то же время, требует ускоренного усвоения большого объема новых знаний, необходимости учиться в течение всей жизни не только самих студентов, но и преподавателей. Отметим двойственный характер тесной связи программирования и математики. С одной стороны, программирование, имеющее корни в математических дисциплинах, должно быть проще для понимания и усвоения студентами-математиками. С другой стороны, пересечение тезаурусов математики и программирования (например, понятия «функция», «переменная», «целое») вызывает определённые затруднения у обучающихся.

Обращаясь к общим ментальным характеристикам, свойственным многим современным студентам, необходимо указать тот факт, что, наряду с таким негативным фактором, как наличие «клипового» мышления, многие исследователи отмечают и положительные тенденции. Мультимедийная виртуальная среда, в которой общаются, играют, учатся современные школьники и студенты, развивает их зрительное восприятие, ориентацию в пространстве, координацию движений, способность к сетевому взаимодействию и быстрому принятию решений в условиях многозадачности и параллельности [200]. Очевидно, что эти особенности сенсорного восприятия и мышления современного «цифрового» поколения должны быть учтены в методике обучения программированию.

Среди объективных причин, обуславливающих неуспешность освоения курса программирования многими студентами, наиболее существенной причиной является сложность теории и практики программирования, которая, в первую очередь, вытекает из семантического разрыва между естественным языком человека (программиста) и искусственным языком программирования. Для большинства людей не представляет сложности сформулировать последовательность инструкций для других людей на естественном языке. Трудности возникают, когда нужно выразить такие инструкции в элементарных операциях, которые может выполнить компьютер. По выражению В.Ш. Кауфмана, «традиционное для компьютеров управление посредством указания действий, а не

целей требует учета мельчайших нюансов всех обстоятельств, в которых может оказаться исполнитель» [71, с. 36]. В подтверждение этого, можно привести тот факт, что при изучении темы «Сортировки массивов», студенты (не знакомые с классическими алгоритмами), успешно демонстрируют и формулируют на естественном языке алгоритмы сортировки выбором и сортировки вставками при решении следующей задачи: расположить по алфавиту подписанные листы с контрольными работами, которые они выполнили. В то же время, реализация этих же алгоритмов на языке программирования, вызывает серьёзные затруднения и непонимание.

В многочисленных зарубежных исследованиях приведены факты о том, что порядка 30% приступивших к изучению курса программирования, терпят неудачу [187], [189], [199], [201], [203]. Наиболее цитируемой работой среди исследований, посвященных вопросам изучения и преподавания программирования, является отчет группы ученых, объединившихся в 2001 г. на конференции ITICSE (Innovation and Technology in Computer Science Education) под руководством М. Маккракена, которые проанализировали ответы студентов из разных учебных заведений нескольких стран на одну и ту же задачу [195]. Средний показатель составил всего 21% – крайне низкий результат, неприятно удививший, в первую очередь, самих преподавателей, ожидавших более высокие показатели. Исследование способностей к программированию студентов, изучающих базовый курс программирования, было вновь проведено той же группой в 2013 г. Результаты тестирования по основам программирования (базовые понятия) в среднем составил 42% [202]. Как видим, результаты улучшились в 2 раза, но, по-прежнему, не удовлетворяют ни преподавателей, ни самих студентов. К сожалению, подобные масштабные исследования не проводились в отечественной системе высшего образования.

Важным фактором, усложняющим работу, в первую очередь, преподавателя, является различный предшествующий опыт программирования студентов-первокурсников. Таким опытом обладают немногие абитуриенты. Статистика ЕГЭ по предмету «Информатика и ИКТ» показывает, что лишь около 8% выпускников

средней школы выбирает этот предмет для сдачи экзамена [123]. И хотя средний балл сдавших экзамен достаточно высок – примерно 63% (данные 2013 г.), более четверти экзаменуемых не приступает к решению заданий уровня «С», что говорит о значительных пробелах в знании предметной области программирования и несформированности практических навыков программирования. Следует также отметить, что, при поступлении абитуриентов на некоторые математические направления подготовки, в качестве вступительного экзамена вузы принимают результаты ЕГЭ по физике, а не по информатике. Как уже было отмечено ранее, практическое отсутствие конкурса на математические направления подготовки, обуславливает низкие стартовые показатели первокурсников в обучении программированию. Это подтверждается результатами входного тестирования по программированию среди студентов-первокурсников Института математики и фундаментальной информатики Сибирского федерального университета (задание на входном тестировании приведены в приложении 1). Рисунок 1 иллюстрирует результаты входного тестирования, которые автор проводил в 2012-2014 гг.

Гистограмма показывает, что более половины студентов (от 49% до 61%), поступивших на 1 курс, не справляется с входным тестированием и, к сожалению, имеется тенденция к увеличению этого показателя. Отметим, что неудовлетворительным считается результат менее 40% выполненных заданий в соответствии с минимальным пороговым баллом ЕГЭ по предмету «Информатика и ИКТ» – 40 баллов.

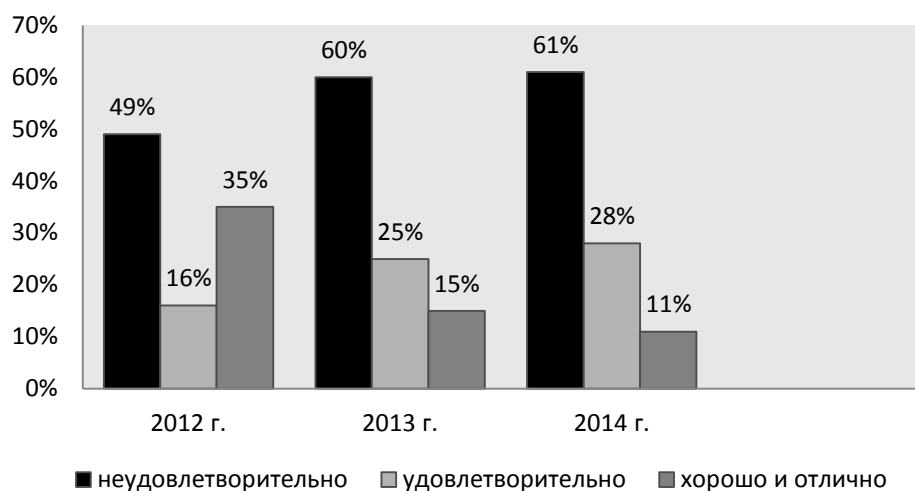


Рис. 1. Динамика результатов входного тестирования с 2012 по 2014 годы

Одновременно с входным тестированием, первокурсникам предлагается заполнить анкету. Один из вопросов анкеты – выяснение того, изучались ли в школе языки программирования и умеют ли респонденты использовать формальное описание алгоритма (блок-схемы, алгоритмический язык). Данные анкетирования, приведенные в [12], были дополнены результатами 2012-2014 гг. (рисунок 2). Сравнение полученных данных с результатами входного тестирования говорит об их несоответствии: в разные годы от 10 до 38% первокурсников, изучая программирование в школе, не могут показать удовлетворительные результаты во входном тестировании (имея, как правило, оценки «4» и «5» в аттестате по информатике).

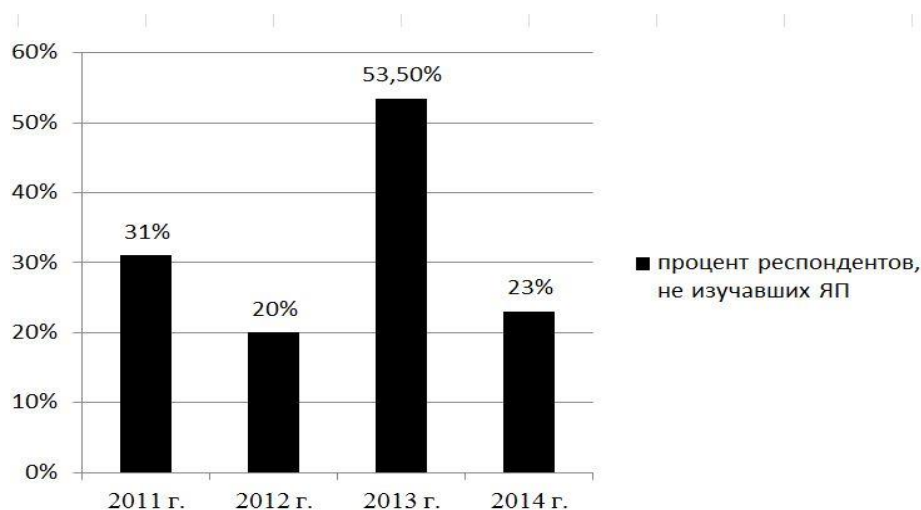


Рис. 2. Динамика результатов анкетирования с 2011 по 2014 годы

В набор заданий, предлагаемый во входном тестировании, были включены простые задания на определение уровня развития алгоритмического и системного мышления, которые не требуют знания какого-либо языка программирования. Как показывает практика, такие задания также вызывают значительные затруднения у тестируемых. Подавляющее большинство испытуемых представляет алгоритм как линейную последовательность каких-либо действий, имеет нечеткое понимание системы как совокупности взаимосвязанных элементов, образующих единое целое, затрудняется с выявлением связей между элементами системы и их категоризацией. Результаты тестирования 64 студентов по трёхбалльной шкале оценивания показаны на рисунке 3.

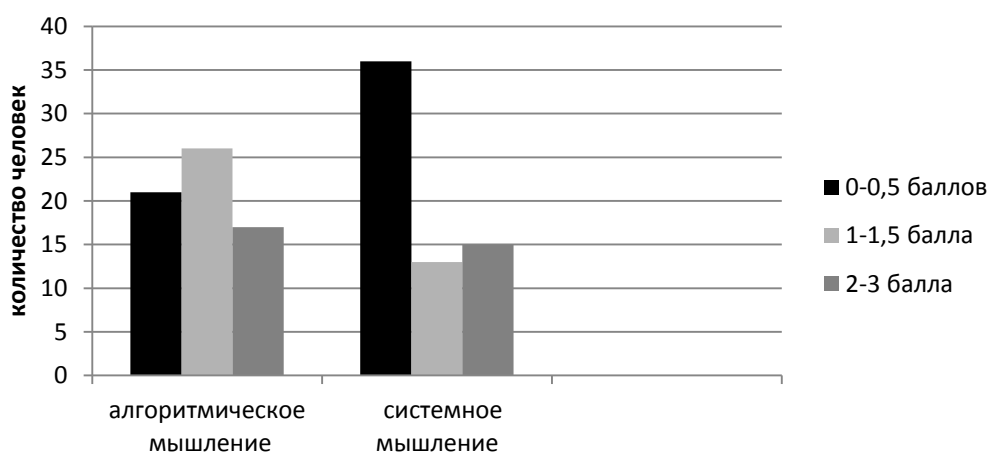


Рис. 3. Результаты тестирования уровня развития алгоритмического и системного мышления

Таким образом, комплекс проблем, возникающий в процессе обучения программированию, требует их разрешения (в идеале) или сглаживания при разработке соответствующей методики обучения.

Для достижения поставленной цели диссертационного исследования необходим предварительный анализ существующих подходов к обучению программированию. Несмотря на недавнее становление программирования как научной дисциплины, разработано большое количество методик обучения программированию. Большинство исследователей рассматривает собственно методики обучения программированию в двух аспектах:

- использующие формальный (дедуктивный) подход, когда изложение материала происходит в строгой последовательности от формального описания конструкций программирования к программной реализации и далее к примерам использования;

- использующие неформальный (индуктивный) подход, суть которого в обучении программированию на примерах готовых программ с постепенным объяснением синтаксиса и семантики программных конструкций.

Оба подхода считаются традиционными, имеют свои достоинства и недостатки. В контексте обучения программированию студентов-математиков, на первый взгляд кажется, что более предпочтительным выглядит первый подход, т.к. именно он используется при изучении классических математических дисциплин. Признавая этот факт, тем не менее, отметим важность повышения мотивации и заинтересованности студентов в обучении, что более эффективно обеспечивается с помощью второго подхода. Вывод очевиден – необходимо разумное сочетание обоих методических подходов.

Методические системы обучения должны быть выстроены на концептуально-методологической основе общедидактических подходов к обучению. Следует отметить, что не существует единой классификации возможных подходов к обучению программированию, сгруппированных по общей методологической позиции. В данном вопросе существуют разные точки зрения.

Так, В.Ш. Кауфман [71] выделяет пять основных позиций рассмотрения и изучения языка программирования:

- технологическая (программирование рассматривается с точки зрения производства программных услуг);

- семиотическая (язык программирования – это знаковая система, акцент ставится на строгом формализованном изложении учебного материала);

- авторская (автор, создавая новый язык программирования с определённой целью, как правило, создаёт учебно-методические материалы для его изучения со своей позиции);

- математическая (обучение на основе формальной грамматики, основа – создание математической модели языка программирования);
- реализаторская (подход к языку программирования как средству практического программирования).

Д.М. Гребнева рассматривает следующие подходы к обучению программированию: системный, деятельностный, семиотический, когнитивный, проблемный [48].

По мнению В.Е. Жужжалова [63], В.Ю. Нефедовой [103] методика обучения программированию зависит от выбора парадигмы программирования: процедурной, объектно-ориентированной, логической, функциональной.

Системный и деятельностный подходы являются базисом любой методической системы. Системный подход к научному исследованию и проектированию в разных сферах социальной практики заключается в применении принципа системности, согласно которому, все предметы и явления мира представляют собой системы разной степени целостности и сложности [183]. В то же время, согласно А.Г. Асмолову, «деятельность выступает в качестве основания таких систем как «психика», «сознание», «бессознательное» и «личность», порождаемых движением деятельности человека в природе и обществе», что делает правомерным использование понятия «системно-деятельностный подход» как конкретно-научной методологии психологии и педагогики [3].

В проектировании методической системы обучения программированию также должен быть применен системный подход, в то же время, обучение программированию невозможно без практического компонента, т.е. деятельности.

В предметной области методики обучения программированию целесообразно выделить следующие подходы: *проблемный, дифференцированный, когнитивный, конструктивно-игровой*. Обращаясь к классификации методических систем академика А.М. Новикова, следует добавить в этот список *контекстный* и *проективный* подходы [106].

Проблемное обучение (Дж. Дьюи, Дж. Брунер, А.М. Матюшкин, М.И. Махмутов, И.Я. Лернер, Н.Г. Дайри и др.) основано на систематическом создании и разрешении проблемной ситуации в процессе обучения, побуждающей обучаемого к самостоятельной поисково-исследовательской деятельности под руководством преподавателя. И.Я. Лернер, М.Н. Скаткин [65, с.72] выделили следующие виды проблемного обучения: проблемное изложение (когда задачу ставит и решает преподаватель, а обучаемые сотрудничают с ним, высказывают свои мнения, предположения); частично-поисковый (эвристический) метод обучения (активное вовлечение обучаемых в процесс решения под руководством преподавателя); исследовательский метод (полная самостоятельность обучаемых, осуществляющих научно-исследовательскую работу, консультируясь с преподавателем). По мнению В.И. Загвязинского, проблемное обучение, несмотря на многие достоинства, имеет ограничение на область применимости: это должен быть материал высокого уровня значимости (методологический, общенаучный) и допускающий альтернативное решение, оценку, толкование [65]. В предметной области программирования проблемный подход (А.В. Диков, Е.В. Касьянова, К.Ю. Поляков [53], [133]) во многом пересекается с индуктивным (неформальным) подходом и может применяться непосредственно на практических занятиях: обучающиеся методом проб и ошибок создают и отлаживают программный код, пользуясь возможностями современных интегрированных сред программирования, реализующих интеллектуальный интерфейс. На лабораторных (семинарских) занятиях также целесообразны элементы проблемного обучения. Достоинством этого подхода является поддержание интереса к программированию у обучающихся через практическое решение задачи и получение видимого результата, но, при этом желательно наличие некоторого опыта программирования. Также могут потребоваться значительные временные затраты.

Дифференцированный подход в обучении основан на учете интересов и способностей обучаемых, которые объединяются в группы по некоторому признаку. В предметном обучении программированию дифференцированный

подход рассмотрен в исследованиях И.Н. Фалиной, Т.Б. Захаровой, Н.Д. Есиповой, В.В. Бобкова, Ю.А. Петровой, Л.А. Внуковой и др. [26], [64], [169].

Конструктивно-игровой подход в обучении программированию целесообразно применять, в первую очередь, в образовательной практике для дошкольников и младших школьников, чтобы в доступной, интересной для детей форме дать некоторые основы программирования. Этот подход связан, в первую очередь, с именем психолога и математика С. Пейперта, создателя языка LOGO [128]. Отечественными разработками на основе конструктивно-игрового подхода являются системы программирования КуМир и Пиктомир (А.Г.Кушниренко, А.Г.Леонов, М.Г.Эпиктетов и др.) [81], [82], Роботландия (А.А. Дуванов, Н.Д. Шумилина) [56].

Контекстное обучение (А.А. Вербицкий) предполагает динамическое моделирование предметного и социального содержания профессионального труда с помощью семиотической обучающей модели (передача и усвоение студентами знаковой информации), имитационной модели (квазипрофессиональная деятельность в аудиторных условиях) и социальной обучающей модели (учебно-профессиональная деятельность) [35]. Методики обучения информатике на основе контекстного подхода предложены М.В. Мащенко, Л.Д. Ситниковой, обучения программированию – В.М. Гриняк, Н.Л. Слугиной [157]. Контекстное обучение в зарубежном образовании понимается в более широком смысле: «это образовательный процесс, цель которого помогать обучающемуся увидеть смысл в изучаемом материале, находить его связи с контекстом своей личной, социальной, профессиональной и культурной жизни» [36]. Достаточно успешное применение контекстного подхода в обучении программированию констатировали М. Гуздьял, Ч. Фаулер, Б. Саймон, R.H. Sloan, P. Troy, A.E. Tew [68], [198]. Например, М. Гуздьял с коллегами проводил эксперимент по внедрению вводного курса программирования в контексте обработки цифровых материалов (монтаж, редактирование цифровых фотографий с помощью языка программирования). Отмечая положительную реакцию студентов на курс и рост среднего процента успешной сдачи экзаменов, тем не менее, автор делает вывод, что «исследования

контекстной методологии доказывают лишь то, что для разных групп студентов и преподавателей возможно улучшение успеваемости на вводных курсах программирования, но не гарантируют, что это улучшение обязательно произойдет» [68, с. 130–144].

Когнитивный подход, который более подробно будет рассмотрен в §1.3, достаточно широко применяется в практике обучения программированию, особенно зарубежной. Перечислим следующие имена: J.R Anderson, C. Bishop-Clark, C. Boyle, A. Corbett, R.E. Brooks, E. Soloway, J. Sorva, J.C. Spohrer, T. Bajo, P. Gonzalvo, J. М. Нос, R.S. Rist, M.B. Швецкий, И.Г. Семакин, С.М. Окулов, А.Р. Есаян, Е.А. Евсюкова, В.Е. Зюбин, А.В. Бабич.

Необходимо отметить, что практически все авторские методики обучения программированию (Ю.А. Первин, И.Р. Дединский, В.Е. Жужжалов, И.Н. Фалина, В.Ю. Нефедова, В.М. Гриняк, Е.А. Евсюкова, Н.Г. Саблукова, А.И. Лапо, Д.М. Гребнева, И.В. Рожина, Н.С. Толстова, А.Н. Петров, М.С. Орлова [51], [62], [103], [122], [129], [166], [169] и др.) базируются на интеграции нескольких подходов, что даёт синергетический эффект для достижения результатов обучения программированию. Тем не менее, смена парадигмы высшего образования, студентоцентрированный подход к обучению, профессиональные требования работодателей к выпускникам высшей школы, определяют необходимость дальнейшего совершенствования и модернизации компонентов методической системы обучения программированию, которую можно эффективно применять в процессе подготовки бакалавров-математиков. Современная методика обучения программированию должна не просто помочь студенту овладеть некоторыми практическими навыками программирования, но и способствовать развитию определённых стилей мышления, свойственных математикам, умению и желанию учиться самостоятельно и в течение всей жизни.

Востребованность общества в высоком уровне квалификации программистов, анализ новых образовательных стандартов по математическим направлениям подготовки, определяют необходимость развивать когнитивные способности обучающегося. Успешность формирования компетенций, связанных

с планированием деятельности и применением стандартных методов и средств в определённых ситуациях, зависит от уровня развития алгоритмического мышления субъекта. Высокий уровень развития системного мышления обеспечит способность разрабатывать и создавать новые математические и программные решения, рефлексировать над результатами деятельности, работать совместно с профессионалами в других областях.

Подводя итог обзору существующих проблем и методических подходов в обучении программированию, сформулируем *требования*, предъявляемые к методике обучения программированию в современных условиях:

- 1) необходимость системного (комплексного) подхода;
- 2) необходимость принятия во внимание социального контекста развития общества;
- 3) учет компетентностной модели системы высшего образования, выраженной в новых образовательных стандартах третьего поколения и переход к ФГОС ВПО 3+;
- 4) соответствие современному уровню развития вычислительной техники, программного обеспечения, теории и практики программирования;
- 5) эффективная организация самостоятельной работы студентов;
- 6) обеспечение трёхкомпонентного взаимодействия между студентами, преподавателем и средствами ИКТ [139], т.е. субъект-объект-субъектных отношений;
- 7) учет когнитивных особенностей обучаемых, предоставление широкого спектра образовательных ресурсов;
- 8) акцент на понимании сложного учебного материала;
- 9) обеспечение практико-ориентированного характера обучения;
- 10) включение обучающихся в научно-исследовательскую и профессиональную деятельность;
- 11) формирование ценностно-мотивационного отношения к учебной деятельности в виде осознания необходимости постоянного пополнения знаний и умений и желания стать квалифицированным специалистом.

1.2. Возможности проективно-рекурсивной стратегии в обучении программированию

Концепция проективного подхода к обучению сформулирована в работах Г.Л. Ильина, Н.И. Пака, В.С. Безруковой, Л.И. Гурье, И.В. Богомаз. По словам Г.Л. Ильина, «центральным понятием проективного образования личности является проект – замысел решения проблемы, имеющей для его носителя жизненно-важное значение» [70, с. 34]. Г.Л. Ильин видит в проективном образовании одну из форм непрерывного образования (образования в течение всей жизни). В настоящее время именно в этом направлении – удовлетворении жизненной потребности личности в образовании в условиях стремительно меняющегося информационного общества, происходит реформа высшей школы. Осознание личностью необходимости в получении нового знания для реализации поставленных перед собой целей, становится движущей силой проективного образования. По мнению автора концепции проективного образования, это образование персонифицированное, более того, «элитарное», рассчитанное на творческих, одарённых и целеустремлённых людей [69]. Ю.Г. Татур, анализируя концепцию проективного образования, отмечает, что в ней «наиболее отчетливо фиксируется роль студента как ведущего субъекта процесса образования. В ходе проектировочной деятельности он не просто поглощает кем-то добытые знания, но добывает их сам, творит истину, чем обогащает интеллектуальную и духовную культуру общества...» [164].

Л.И. Гурье рассматривает проективное образование в двух аспектах: как форму личностного развития (может выступать в качестве основной и исходной формы образования, возможной на любой стадии развития) и как социальную систему (наличие определённых ограничений, связанных с реализацией идей, заложенных в личных проектах) [50]. Таким образом, условием и основой развития проективного образования является школьное и высшее образование.

Хотя Г.Л. Ильин, четко разделяет понятия «образование» (процесс, протекающий в течение всей жизни человека) и «обучение» (целенаправленный

процесс, протекающий в специально организованных условиях), принцип проективности можно корректно перенести на обучение. Цель проективного обучения – формирование у обучающегося потребности к саморазвитию и самообразованию, способности к креативному инновационному мышлению через реализацию собственного профессионального или жизненного проекта, отличающегося от готовых решений. Задача педагога при таком обучении – оказать квалифицированную помощь обучающемуся в проектировании и реализации личного замысла, достижении поставленных целей.

В.С. Безрукова в [23] сформулировала принципы и задачи проективной педагогики на примере проективной модели личности и деятельности инженера-педагога. Позиция автора заключается в следующем: возможно не только проектирование отдельных элементов педагогического процесса (урока, педагогических ситуаций) и всего процесса в целом, но и педагогическое проектирование личности обучающегося. Под таким проектированием автор понимает разработку и описание возможного варианта развития обучающегося с учетом требований времени, личных возможностей и конкретных условий, созданных в учебном заведении.

Н.И. Пак в [124, с. 42] рассмотрел возможности применения проективной стратегии к организации и проектированию сложноорганизованных динамических открытых систем, к которым относятся образовательные системы. Автором было введено понятие проективной образовательной системы, организованной по принципу «многие – ко – многим», т.е. создаваемой многими участниками для совместного использования, допускающей модернизацию, корректировку, эволюционное развитие. Проективная образовательная система – это открытая гибкая система, моделируемая и развивающаяся в виде проекта с определёнными характеристиками и поведением в настоящем и будущем. Такая система функционирует на следующих принципах:

- 1) открытая архитектура – возможность «безболезненной» реконструкции организационной структуры, обеспечивающая универсальность системы;

- 2) рекурсивное проектирование системы – проект системы предполагает проектирование новых и отдельных его компонентов;
- 3) информационная открытость – вся нормативно-правовая, методическая, научная и учебная информация открыта для всех субъектов образовательного процесса;
- 4) оптимальность и простота использования;
- 5) свободный, но ответственный доступ к участию в разработке системы;
- 6) непрерывность и эволюционность – система не ждет завершения проекта, она эксплуатируется, непрерывно развивается, не предполагая революционных изменений.

Примером проективной образовательной системы, построенной на данных принципах, является открытая образовательная среда openSEE (авторская разработка Д.Н. Буторина) [30], [31].

Таким образом, инициирование, моделирование и реализация проективной образовательной системы (системы обучения) предполагает проектирование и динамичное развитие всех её компонентов с привлечением всех участников проекта: студентов, преподавателей, заказчиков (например, возможных работодателей или студентов и преподавателей других вузов).

И.В. Богомаз на основе проективно-информационного подхода определила проективную методическую систему обучения студентов в условиях информационно-коммуникационной предметной среды [28]. В исследованиях С.Л. Фоменко, Т.А. Султановой, О.Е. Ломакиной, П.Н. Пестрякова, С.А. Виденина, Ю.С. Баранова, С.В. Светличной [21], [37], [147], [156], [161] рассмотрены перспективы и приведены примеры использования проективного подхода в обучении в различных предметных областях.

Среди характеристик проективной методической системы особую важность имеет принцип рекурсивности. Как известно, рекурсия – это самоподобие. В применении к методике обучения, принцип можно трактовать следующим образом: в процессе обучения некоторой дисциплине обучающийся создаёт учебные элементы (ресурсы), которые использует сам и предоставляет другим

участникам образовательного процесса. Естественным следствием того, что проектируется методика обучения программированию студентов-математиков, будет необходимость усиления этого принципа, т.к. рекурсия (рекурсивные алгоритмы, рекурсивные функции) относятся к категориальному аппарату и математики, и программирования. Поэтому мы предлагаем назвать методику обучения программированию студентов-математиков методикой *проективно-рекурсивного обучения*. В отличие от работ Ю.С. Баранова [21], С.В. Светличной [147], где проективно-рекурсивное обучение предполагает создание обучающимися различных образовательных ресурсов с помощью пакетов прикладных программ, в предлагаемой методике обучение программированию происходит, в том числе, через ЭОР, созданные средствами языка программирования.

Необходимым элементом проективно-рекурсивной стратегии в обучении является педагогическое проектирование. По мнению И.А. Колесниковой, педагогическое проектирование – это «практико-ориентированная деятельность, целью которой является разработка новых, не существующих в практике образовательных систем и видов педагогической деятельности» [76, с. 17].

Согласно В.С. Безруковой, процесс проектирования включает несколько этапов, среди которых можно выделить [23, с. 117]:

- моделирование: разработка целей (общей идеи) создания педагогических систем, процессов или ситуаций и основных путей их достижения;
- собственно проектирование: дальнейшая разработка созданной модели и доведение ее до уровня практического использования;
- конструирование: дальнейшая детализация созданного проекта, приближающая его для использования в конкретных условиях реальными участниками учебно-воспитательного процесса.

И.А. Колесникова рассматривает следующие этапы проектирования:

- предпроектный (стартовый) этап: исследование, проблематизация, концептуализация, целеполагание, ценностно-смысловое самоопределение, форматирование проекта;

- реализация проекта: уточнение цели, функций, задач, пошаговое выполнение проектных действий, коррекция, получение и внутренняя оценка проекта, презентация результатов работы и их внешняя экспертиза;
- рефлексивный этап (оценка хода и результатов проекта, системы отношений внутри проекта, осмысление перспектив использования проектного продукта);
- постпроектный этап (апробация, распространение результатов и продуктов проектной деятельности, выбор вариантов продолжения проекта, объединение своего проекта с другими и т.д.).

В определении А.М. Новикова и Д.А. Новикова, проект представляет собой завершённый цикл продуктивной деятельности отдельного человека, коллектива, организации или нескольких предприятий [105]. В жизненном цикле проекта (совокупности последовательных этапов развития проекта) авторы указывают следующие фазы:

- фаза проектирования, включающая концептуальную стадию, моделирование, конструирование и стадию технологической подготовки;
- технологическая фаза;
- рефлексивная фаза.

Рассматривая педагогическое проектирование с позиций проективно-рекурсивной стратегии, мы утверждаем, что рефлексивный и постпроектный этапы проектирования проективной методической системы становятся циклически замкнутыми с этапами моделирования и реализации (технологической фазой). Субъекты проектирования (студенты, преподаватели, возможные работодатели), как личности со своими потребностями, целями, способностями, генерирующие новые идеи, будут оказывать влияние на содержательный и технологический компоненты методической системы. Таким образом, представим этапы проектирования проективной методической системы с помощью следующей схемы (рисунок 4):

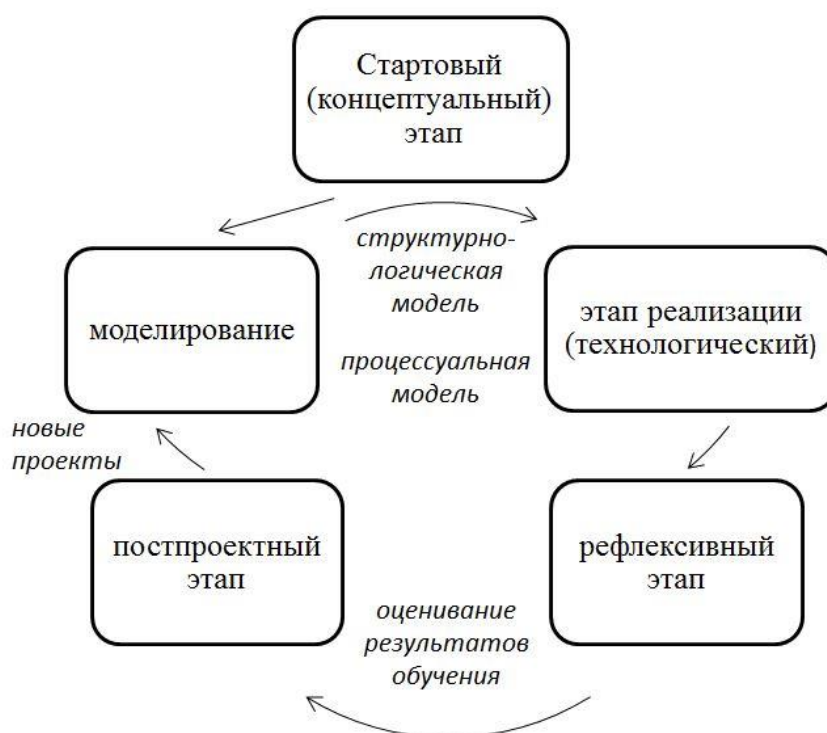


Рис. 4 Жизненный цикл проективной методической системы

Сформулируем *принципы*, на основе которых возможно *обновление компонентов методической системы обучения* программированию с позиций проективно-рекурсивной стратегии (опираясь на [124]):

- открытость и доступность для всех участников (включая информационный и архитектурный компоненты);
- динамичность (гибкая архитектура, коллективное проектирование всех компонентов);
- рекурсивность (проектирование новых элементов как самостоятельных проектов);
- модульность (каждый модуль системы представляет собой результат научно-исследовательской проектной работы одного или нескольких участников);
- эволюционность (поступательное и непрерывное развитие компонентов системы);
- адаптивность (наличие обратной связи в системе, позволяющей оперативно реагировать на внешние воздействия и изменение состава и взаимоотношений участников).

Рассмотрим возможности применения проективно-рекурсивной стратегии к обновлению всех компонентов методической системы обучения программированию. Предварительно следует уточнить структуру (компонентный состав) методической системы. Анализ научно-педагогической литературы по данному вопросу показал, что к определению методической системы как совокупности целей, содержания, методов, средств и форм обучения, данному А.М. Пышкало [135] (вслед за ним Н.В. Кузьминой, Ю.С. Брановским, М.В. Швецим, Н.И. Рыжовой и др.) ряд исследователей добавляет результативный и оценочный компоненты (Г.И. Саранцев, В.Г. Крысько, Н.Л. Стефанова др.), деятельностный или процессуальный компоненты (М.В. Рыжаков, Н.А. Черникова, М.А. Шаталов и др.). Будем рассматривать методическую систему обучения программированию, состоящую из следующих компонентов:

- целевой компонент (постановка целей обучения);
- содержательный компонент (отбор содержания курса на основе определённых принципов);
- технологический компонент (включающий методы, средства и формы обучения);
- результативно-оценочный компонент (включающий планируемые результаты обучения, методы и формы диагностики и контроля обучения).

Взаимосвязанная совокупность этих компонентов образует структурно-логическую модель обучения программированию. Рассмотрение методики обучения как целенаправленного процесса, совместной деятельности субъектов образования и их взаимодействия с объектами образовательной системы, вызывает необходимость построения процессуальной модели обучения программированию, что и будет сделано в дальнейшем.

Каким образом проективно-рекурсивная стратегия может быть применена к обновлению вышеперечисленных компонентов методической системы?

Анализ требований, предъявляемых к современной методике обучения программированию студентов-математиков, привел нас к выводу, что главная цель

разработки такой методики – достижение результатов обучения, которые можно представить в виде следующей трёхуровневой схемы (рисунок 5):

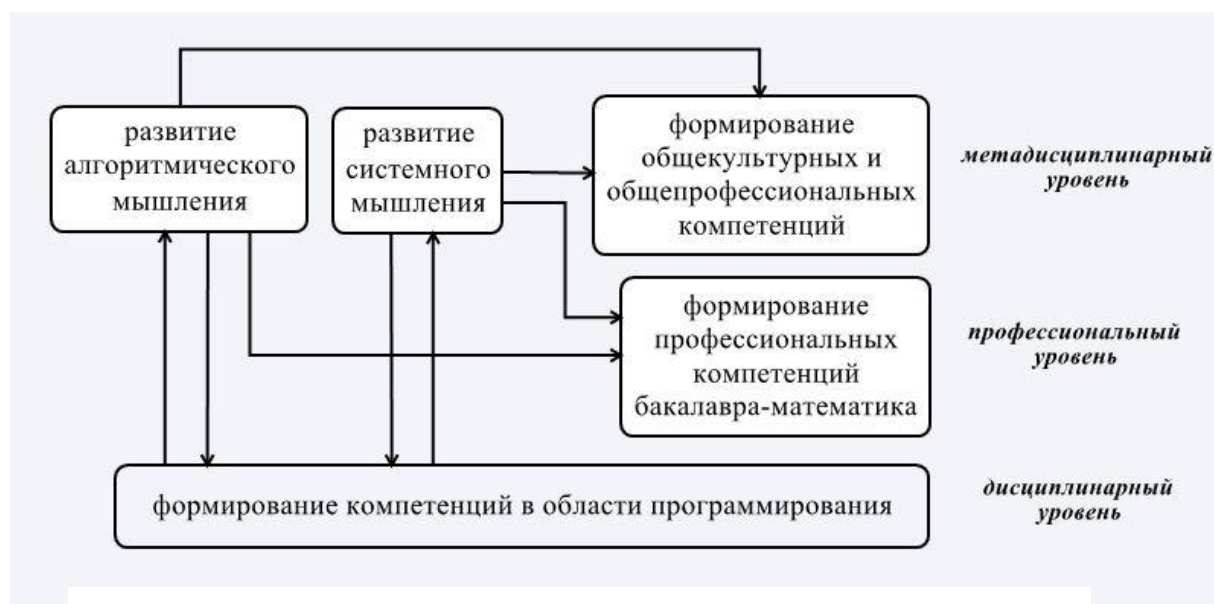


Рис.5. Целевая иерархия методики обучения программированию будущих бакалавров-математиков

Из представленной схемы видно, что развитие алгоритмического и системного мышления связано с формированием ряда компетенций из всех компетентностных категорий: общекультурных, общепрофессиональных и профессиональных. Так, анализ ФГОС ВПО по направлению подготовки 01.03.01 «Математика», вступившего в силу с 1.09.2014 [117], показал, что способность к алгоритмическому и системному мышлению оказывает непосредственное влияние на развитие таких компетенций, как

- способность к самоорганизации и самообразованию (ОК–7);
- способность к самостоятельной научно-исследовательской работе (ОПК–3);
- способность находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем (ОПК–4);
- способность к определению общих форм и закономерностей отдельной предметной области (ПК–1);
- способность строго доказать утверждение, сформулировать результат, увидеть следствия полученного результата (ПК–3);

- способность использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач (ПК–5);
- способность использовать методы математического и алгоритмического моделирования при анализе управленческих задач в научно-технической сфере, в экономике, бизнесе и гуманитарных областях знаний (ПК–7);
- способность представлять и адаптировать знания с учетом уровня аудитории (ПК–8);
- способность к планированию и осуществлению педагогической деятельности с учетом специфики предметной области в образовательных организациях (ПК–10).

Изучение ФГОС ВПО 3+ по направлению подготовки 01.03.02 «Прикладная математика и информатика» [118] выявило необходимость развития алгоритмического и системного мышления с целью формирования следующих компетенций:

- способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой (ОПК–1);
- способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей ... (ОПК–3);
- способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям (ПК–1);
- способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности (ПК–3);
- способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности (ПК–4);
- способность формировать суждения о значении и последствиях своей профессиональной деятельности с учетом социальных, профессиональных и этических позиций (ПК–6);

- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК–7);
- способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК–9);
- способность применять существующие и разрабатывать новые методы и средства обучения (ПК–13).

Подводя итоги анализа новых образовательных стандартов по математическим направлениям подготовки, можно утверждать, что высокий уровень развития алгоритмического и системного мышления у выпускников математических направлений подготовки во многом будет способствовать их успешности в профессиональной карьере и будет показателем общего уровня интеллекта.

В то же время, очевидно, что взаимозависимо развитие специфических компетенций в области программирования и алгоритмического и системного мышления, представляя собой систему вложенных (рекурсивных) целей и одновременно результатов обучения. А.П. Ершов по этому поводу высказался следующим образом: «программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с Эдисоновским талантом сооружать все что угодно из нуля и единицы [58].

Реализация обучения программированию на языке C/C++ позволяет охватить две базовые парадигмы программирования: процедурную (развитие в первую очередь алгоритмического мышления) и объектно-ориентированную (развитие системного мышления). С целью успешного и безболезненного перехода обучающихся с процедурной на объектно-ориентированную парадигму программирования, одновременного развития таких разных стилей мышления, как алгоритмический стиль и системный стиль, мы предлагаем в процессе обучения программированию акцентировать внимание на сквозной содержательной линии «Тип данных» (более подробно рассматривается в §2.1).

Планируемые результаты обучения программированию должны проецироваться (отображаться) на содержательный компонент методической системы. В ходе учебного процесса моделируются условия для решения возможных профессиональных задач. Согласно ФГОС ВПО по направлениям подготовки 01.03.01 «Математика» и 01.03.02 «Прикладная математика и информатика» область профессиональной деятельности бакалавров этих направлений включает разнообразные виды деятельности: научно-исследовательскую, производственно-технологическую, организационно-управленческую, педагогическую [117], [118]. В Концепции развития математического образования в Российской Федерации отмечено, что система профессионального математического образования должна готовить кадры для нужд математической науки, экономики, научно-технического прогресса, безопасности, медицины [110].

В силу этого, существует широкий класс задач из сферы будущей профессиональной деятельности, который может быть реализован в процессе обучения программированию. Традиционное решение типовых задач по программированию необходимо дополнить следующими направлениями:

- Решение профессионально-ориентированных задач, причем с элементами опережающего обучения, т.е. задач, не только из предметных областей параллельно изучаемых математических дисциплин (математического анализа, аналитической геометрии, алгебры и др.), но и изучаемых на втором-третьем курсах: численных методов, теории вероятностей, методов оптимизации, компьютерной графики и др. Подобная практика будет осуществлять проективное обучение.

- Создание студентами программ-тестов на знание и понимание синтаксических, семантических и прагматических особенностей языка программирования, при этом тесты кодируются на C/C++ (пример рекурсивности). Принципиально важным, на наш взгляд, является привлечение к созданию тестов не только сильных студентов, но и тех, кто не показывает удовлетворительного результата в ходе промежуточного контроля, т.к.

самостоятельное формулирование тестовых вопросов способствует пониманию и усвоению пройденного учебного материала.

– Вовлечение студентов в практическую работу над проектами межвузовской кооперации, реализующими сотрудничество студентов и преподавателей разнопрофильных вузов, и проектами IT-бизнеса, дающими возможность студентам сделать первые шаги в профессиональной сфере. Работа в этом направлении также является примером проективного обучения.

Ю.К. Бабанский сформулировал следующие дидактические правила, которые следует соблюдать при отборе содержания обучения [5]:

- подчинять содержание обучения актуальным жизненным задачам;
- обеспечивать соответствие содержания реальным учебным возможностям обучающихся;
- выделять в содержании главное: основные знания, умения, нравственные идеи;
- выбирать оптимальную структуру обучения, устанавливать логические взаимосвязи между элементами содержания.

Дополним перечисленные правила **дидактическими принципами для отбора содержания** проективно-рекурсивной методической системы обучения программированию студентов-математиков:

- принцип проективности (проекция будущей профессиональной деятельности на педагогический процесс в виде реализации личного образовательного проекта);
- принцип рекурсивности (создание и использование образовательных ресурсов самими обучающимися в учебном процессе);
- принцип итеративности (возвращение к задаче на более высоком уровне сложности, с использованием нового учебного материала);
- принцип вариативности (возможности выбора уровня сложности задач, последовательности выполнения и т.д.);
- принцип междисциплинарности (взаимосвязь с курсами математических дисциплин).

Обратимся к обновлению технологического компонента методической системы. Приведенный пример написания студентами программ-тестов является не только элементом содержательного компонента методической системы, но выступает в роли средства обучения, формирующего положительную мотивацию к овладению знаниями и умениями в области программирования. Наиболее качественные и релевантные заявленной теме программы-тесты размещаются (с указанием авторства) на электронном курсе в качестве обучающих электронных ресурсов. Ими могут воспользоваться не только сокурсники, но в будущем следующее поколение студентов. Создаваемые студентами средства обучения можно отнести к педагогическим программным средствам (ППС). Согласно определению И.В. Роберт, ППС представляет собой прикладную программу, предназначенную для организации и поддержки учебного диалога пользователя с компьютером, имеющую обратную связь с пользователем и учитывающую индивидуальные возможности и предпочтения обучаемого [138, с. 127].

Другими важными для рассматриваемого курса средствами обучения являются ментальные и концептуальные карты, которые более подробно рассматриваются в параграфе 1.3. Существенное значение для повышения эффективности обучения программированию будет иметь использование подобных карт в следующих аспектах:

- как средства визуализации знания, способствующие пониманию и лучшему усвоению учебного материала;
- как средства оценивания результатов обучения;
- как программные продукты, создаваемые самими студентами.

В создании ментальных и концептуальных карт также может быть применена проективно-рекурсивная стратегия. Представляются возможными следующие способы реализации:

1. Ментальная карта по определенной теме создаётся путем обращения к ранее разработанным картам с помощью гиперссылок.
2. Модернизация концептуальной карты путем встраивания нового понятия (алгоритма, раздела курса) в существующую карту.

3. Объединение концептуальных и ментальных карт: концептуальная карта создается как иерархия понятий, каждое из которых будет представлено ментальной картой.

Таким образом, все вышесказанное позволяет сделать вывод о возможности и целесообразности применения проективно-рекурсивной стратегии при модернизации всех компонентов методической системы обучения программированию.

1.3. Повышение эффективности обучения программированию за счет использования когнитивных технологий

Когнитивный подход в обучении связан со становлением и бурным развитием когнитивной науки, интегрировавшей успехи многих наук: нейрофизиологии, биофизики, психологии, информатики, лингвистики. Когнитивная наука представляет собой междисциплинарную область исследований, направленных на понимание природы мышления, моделирование и изучение работы мозга и мыслительных процессов. Фундаментом для развития концепции когнитивного обучения в педагогической науке послужили труды крупнейших отечественных ученых Л.С. Выготского, А.Р. Лурия, С.Л. Рубинштейна, Б.М. Теплова, Г.С. Альтшуллера, М.С. Шехтера, П.Я. Гальперина, Б.М. Величковского. Среди основоположников когнитивного обучения за рубежом необходимо упомянуть А. Бине, Э.Ч. Толмена, Дж.С. Брунера, Н.А. Бернштейна, Ж. Пиаже, Г.А. Саймона, современных психологов Ф. Джонсона-Лэйрда, Г.Э. Гарднера, Р. Аткинсона, Э. Спелке, М. Кай и др. Когнитивный подход в обучении связан, в первую очередь, с целенаправленным и последовательным развитием мышления обучающегося на основе его ментального опыта. Научить думать, размышлять, и, следовательно, учиться эффективно – сущность когнитивного обучения.

Отметим следующие научные исследования и примеры практического применения когнитивного подхода в обучении: М.Е. Бершадский,

М.В. Ядровицкая, А.А. Гилев, Т.Н. Колодочка, М.Б. Уразова, С.В. Шикунова, Н.В. Волосухина, Т.С. Дорохова, Д.В. Черникова, Ю.О. Швецова, А.Ю. Нагорнова.

Изучение и анализ научной литературы по когнитивной психологии (У. Найссер, Д. Андерсон, Р. Солсо, Ж.Ф. Ришар, Л.М. Веккер, Б.М. Величковский, М.А. Холодная, В.М. Аллахвердов и др.) показал, что центральное понятие когнитивной психологии – когнитивная схема (также называемая когнитивная структура, ментальная схема) должно быть рассмотрено в предметной области программирования.

Один из основоположников когнитивной психологии У. Найссер когнитивными («предвосхищающими» – у автора) схемами называл обобщенно-визуальные образования, возникающие как результат интеграции зрительных, слуховых и тактильно-осязательных впечатлений [102].

По определению М.А. Холодной, когнитивная схема представляет собой «обобщенную и стереотипизированную форму хранения прошлого опыта относительно строго определенной предметной области... ». При этом М.А. Холодная различает когнитивные структуры двух типов: когнитивные структуры как фиксированные формы опыта с «горизонтальным» принципом формирования и когнитивные структуры – интегрированные формы опыта с «вертикальным» принципом формирования, т.е. это структуры более высокого уровня (порядка) [171].

Н.И. Пак определяет ментальную схему как «форму существования образа объекта в памяти в виде ощущений и их модельных представлений, определяющих его свойства в пространстве и поведение во времени» [126].

Наиболее близкой к предметной области программирования является теория фреймов М. Минского (создавшим свою теорию в ходе работ над искусственным интеллектом), рассматривавшем фрейм как когнитивную схему с устойчивым каркасом и вариативными узлами (слотами). М. Минский разделял фреймы на «фрейм-визуальный образ» и «фрейм-сценарий» [99].

Другие основополагающие понятия когнитивной психологии, на которых базируется когнитивный подход в обучении, – понятия концепта и ментальной

репрезентации. Концепт – это понятийная структура, «базовая когнитивная сущность, связывающая смысл со словом» (Ж.-Фр. Ришар). В теории психических процессов Л.М. Веккера, концепт – форма интегральной работы интеллекта [33]. В [183] ментальная репрезентация определяется как «внутренний (психический, ментальный) образ или формат кодирования» объектов и явлений реального мира. Ментальная репрезентация представляет собой ситуативную форму ментального опыта индивидуума и, по мнению Ж.-Фр. Ришара, это специфическая детализированная конструкция, построенная в конкретном индивидуальном контексте. Ментальная репрезентация хранится в оперативной памяти человека. Этим свойством ментальная репрезентация отличается от знания, обладающего постоянством и хранящегося в долговременной памяти [137]. Ж.-Фр. Ришар выделяет следующие формы знания:

- знания об объектах – концепты, образующие семантическую сеть;
- знания о ситуациях и событиях, выраженные посредством когнитивных схем;
- знания о действиях (процедурные знания).

Отметим, что понятийный аппарат когнитивной науки – молодой и бурно развивающейся отрасли научного знания, находится в формирующейся стадии, поэтому у разных исследователей можно встретить различные термины. Но все ученые сходятся во мнении, что когнитивные (ментальные) схемы имеют сложную иерархическую структуру, образуют системы (сети) схем. От количества, разнообразия, насыщенности когнитивных схем, прочности связей между ними, зависит интеллект личности, успешность в обучении и дальнейшей профессиональной деятельности. Основываясь на этом положении, будем считать, что обучение программированию потребует формирования у обучающихся ментальных схем следующего вида:

- **понятийные ментальные схемы** (включающие схемы-образы и схемы-действия, например, «переменная» – схема-образ и «присваивание» – схема-действие);

– **алгоритмические ментальные схемы**, образующие систему – иерархию схем от самых базовых (базовые алгоритмические структуры – следование, ветвление, цикл) до схем – шаблонов (паттернов) программирования.

Наиболее важным представляется формирование схем второго вида, на их основе (ментальном опыте) возможно дальнейшее продвижение обучающегося от программиста-новичка до программиста-профессионала.

Наглядное представление о формировании ментальных схем у начинающего программиста можно получить с помощью рисунка 6.

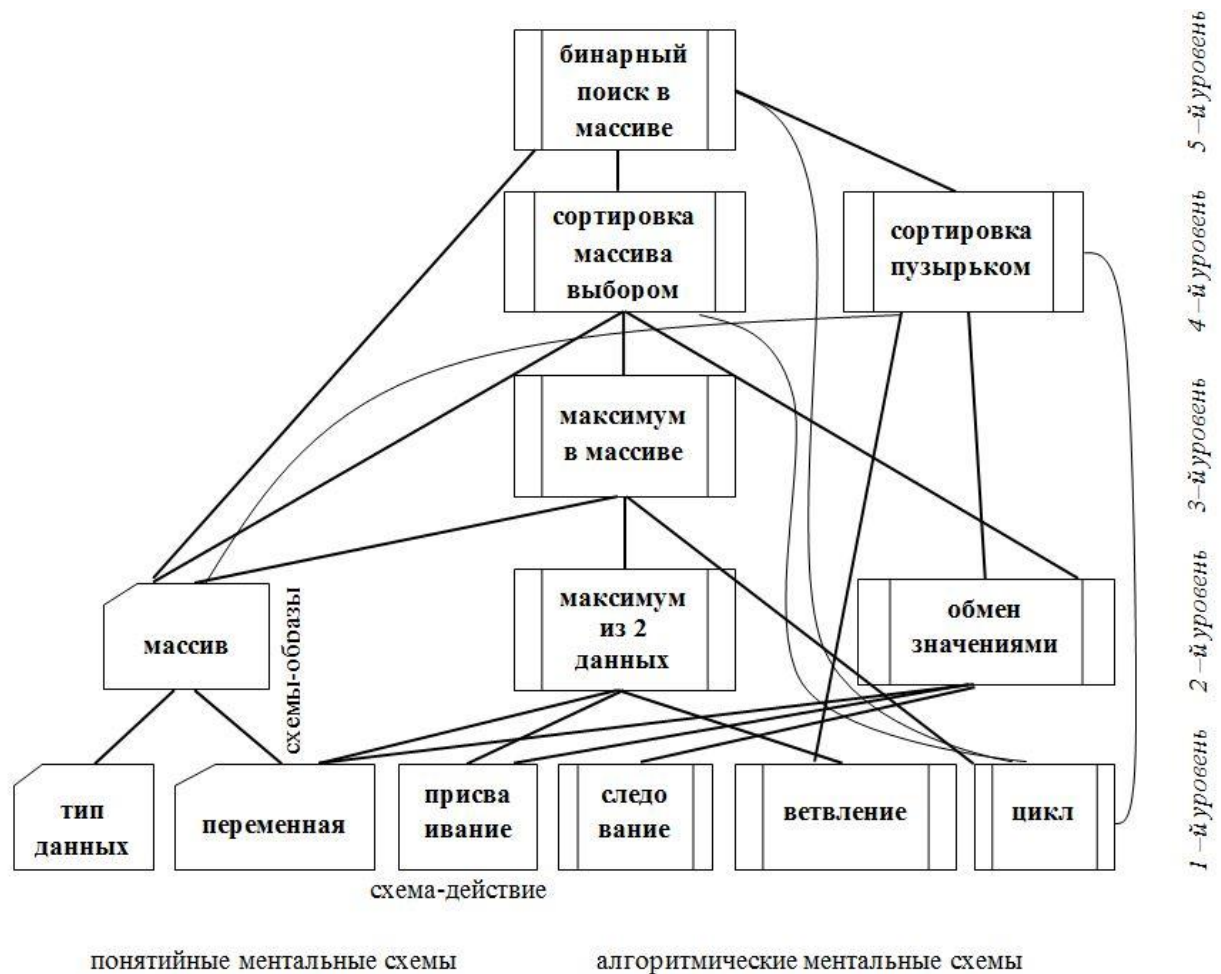


Рис. 6. Пример иерархии понятийных и алгоритмических ментальных схем

На рисунке представлен фрагмент иерархической сети нескольких ментальных схем, включающий как понятийные ментальные схемы (концепты), так и алгоритмические ментальные схемы. На основе сформированных схем базовых алгоритмических конструкций – следования, ветвления, цикла и

концептов «переменная», «тип данных» (схемы 1-го уровня), можно сформировать алгоритмические ментальные схемы «нахождение максимального значения из двух данных (переменных)» и «обмен значениями двух данных (переменных)» (схемы 2-го уровня). Включая в эту сеть концепт «массив», возможно сформировать алгоритмическую ментальную схему «нахождение максимального значения в массиве». На её основе и всех ментальных схем нижнего уровня возможно встроить в иерархию алгоритмическую ментальную схему «сортировка массива выбором максимального элемента» (схема 4-го уровня), далее «бинарный поиск в массиве» (для реализации алгоритма необходимо предварительно отсортировать массив). Включение в иерархию новых концептов и алгоритмических ментальных схем происходит последовательно, путем организации новых связей с предшествующими ментальными схемами.

В качестве экспериментального подтверждения такой организации ментальных схем, приведем следующий пример. При проведении промежуточного контроля по темам «Циклы» и «Операторы ветвления» группе студентов была предложена задача:

Реализовать вывод на экран квадрата, границы которого состоят из символов «». Длина стороны квадрата вводится пользователем с клавиатуры.*

Таким образом, экран консольного приложения в среде Microsoft Visual Studio должен выглядеть, как на рисунке 7.

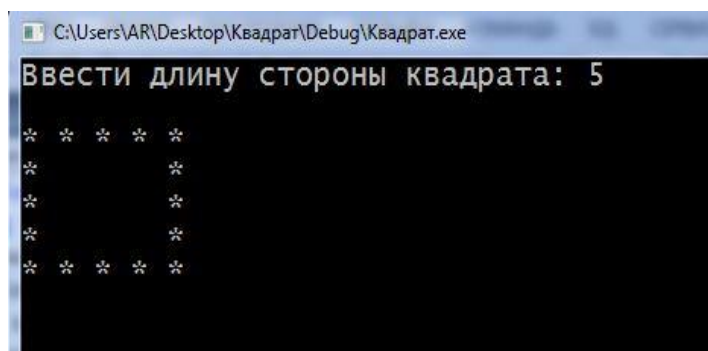


Рис. 7. Результат программного решения задачи о выводе на экран квадрата из символов «*»

Обладая разным практическим опытом программирования до поступления в вуз, студенты по-разному справились с задачей. Соотнеся варианты решения с наличием необходимых для решения ментальных схем, можно прийти к следующим выводам. Не справились с задачей (не предложили работоспособное решение) те студенты, у которых ещё не сформированы прочные связи между ментальными схемами «ветвление» и «цикл» (хотя студенты могут реализовать программы с использованием только условного оператора, или с использованием только цикла) и нет сформированной ментальной схемы «вложенный цикл». Один из вариантов работоспособной программы представлен на рисунке 8.

```
#include <iostream>
using namespace std;
int main()
{setlocale (LC_ALL, "");
  int size,i,j;
  cout<<"Ввести длину стороны квадрата: ";
  cin>>size;
  for (i=0; i<size; i++) cout<<"*";
  cout<<endl;
  for (i=0; i<size-2; i++)
  {cout<<' ';
    for (j=0; j<size-2; j++)
      cout<<' ';
    cout<<'*<<endl;
  }
  for (i=0; i<size; i++) cout<<"*";
  cout<<endl;
  system ("pause");
}
```

Рис. 8. Код решения задачи о квадрате на языке C++ (используются четыре цикла)

В этой программе отдельно выводятся на экран верхняя и нижняя границы квадрата и применен вложенный цикл для печати левой и правой границ. Такие программы реализовали студенты, у которых отдельные ментальные схемы уже сформированы, но нет прочных, устойчивых связей между ними.

Самое оптимальное решение предложили студенты, имеющие достаточный опыт программирования, уже умеющие работать с массивами, т.е. их ментальный опыт позволил сформировать концепт «массив». Репрезентируя понятийную

ментальную схему «массив» к условиям данной задачи, они увидели решение с помощью вложенного цикла, аналогичное использованию двумерного массива (хотя в условиях задачи нет явного указания на это). Код решения представлен на рисунке 9.

```
#include <iostream>
using namespace std;
int main()
{setlocale (LC_ALL, "");
  int size,i,j;
  cout<<"Ввести длину стороны квадрата: ";
  cin>>size;
  cout<<endl;
  for (i=0; i<size; i++)
  {for (j=0; j<size; j++)
    if (i==0 || i==(size-1) || j==0 || j==(size-1))
      cout<<"*";
    else cout<<" ";
    cout<<'\\n';
  }
  system ("pause");
}
```

Рис. 9. Код решения задачи о квадрате на языке C++ (с одним вложенным циклом)

Таким образом, качество программного кода во многом зависит от имеющегося практического опыта и тех шаблонов программирования, которые уже усвоены начинающим программистом.

Реализовать когнитивный подход к обучению программированию возможно при активном использовании инновационных технологий, к числу которых относят когнитивные технологии. Г.Г. Малинецкий определяет когнитивные технологии как «способы и алгоритмы достижения целей субъектов, опирающиеся на данные о процессах познания, обучения, коммуникации, обработки информации» [97]. Используя моделирование познавательных способностей человеческого мозга, когнитивные технологии дают решение многих конкретных прикладных задач – проектирования систем поддержки принятия решений на основе нечеткой логики, распознавания речи, сигналов, изображений, обработки больших массивов данных и т.д. Ученые исследуют

феномен конвергенции NBIC-технологий, определяющих прогресс современного общества (взаимопроникновение и взаимовлияние нанотехнологий, биотехнологий, информационных и когнитивных технологий). Ряд ученых (Д.И. Дубровский, В.А.Лекторский) настаивают на включении в эту группу также социальных технологий.

Когнитивные технологии находят все большее применение в сфере образования. В этом контексте нужно говорить о когнитивных технологиях обучения – комплексе методов, приемов, средств, учитывающих индивидуальные ментальные характеристики обучающихся, и обеспечивающих, в первую очередь, понимание учебного материала и интеллектуальное развитие обучающихся. У педагога-практика М.Е. Бершадского дано следующее определение: «Когнитивная образовательная технология является общепедагогической предметно-независимой индивидуально ориентированной образовательной технологией, обеспечивающей понимание ребёнком окружающего мира путём формирования системы когнитивных схем, необходимых для успешной адаптации к жизни в современном информационном обществе» [25].

Главной целью применения когнитивных технологий в учебном процессе и самообразовании служит существенное улучшение восприятия и понимания, и, следовательно, усвоения учебного материала. Важная роль в развитии и активизации когнитивных процессов обучаемых принадлежит средствам и методам визуализации.

Главные причины возрастания роли визуализированной и хорошо структурированной учебной информации заключаются в следующем. Во-первых, современные нейрофизиологические и психофизиологические исследования мозга человека показывают, что 80-90% информации подавляющее большинство людей получают через органы зрения, кроме того, у большей части взрослых людей ведущим каналом восприятия является зрение, т.е. они являются визуалами. Во-вторых, в эпоху Интернета возникает необходимость учета особенностей сенсорного восприятия и мышления «цифрового» поколения школьников и

студентов, для которых использование мультимедийных технологий становится наиболее естественным способом получения информации.

Необходимо отметить появление многих новых терминов, имеющих отношение к классическому дидактическому принципу наглядности: «визуальное мышление», «визуализация», «когнитивная визуализация», «визуальная грамотность», «визуальная коммуникация», «визуальное обучение».

В.П. Зинченко предложил следующее определение понятия «визуальное мышление»: «Это человеческая деятельность, продуктом которой является порождение новых образов, создание новых визуальных форм, несущих определенную смысловую нагрузку и делающих значение видимым. Эти образы, как и слова языка, отличаются автономностью и свободой по отношению к объекту восприятия» [67]. Термин «визуальное мышление» ввел американский психолог Р. Арнхейм [2]. Он определил его как мышление посредством визуальных операций. При этом визуальные образы существуют не как иллюстрации к мыслям автора, а становятся конечным проявлением самого мышления. Работы Арнхейма положили начало современным исследованиям о роли образных явлений в когнитивной деятельности.

Визуальная грамотность представляет собой целое направление современной педагогики, в котором исследуются проблемы развития навыков пользования визуальной и аудиовизуальной информацией. Визуальную грамотность можно рассматривать в разных аспектах: как процесс коммуникации между объектом и субъектом восприятия, как взаимодействие элементов зрительного образа с субъектом восприятия и как умение субъекта адекватно воспринимать и продуцировать зрительные образы [134]. В процессе визуальной коммуникации взаимодействуют три компонента: субъект, объект, средства коммуникации (или окружающая среда). Понятие «визуального обучения» можно трактовать как обучение на основе развития и использования визуального мышления обучающихся с использованием средств и методов визуализации.

Собственно термин «визуализация» используется в разных аспектах, например, различают внешнюю и внутреннюю визуализацию. Внешняя

визуализация – это процесс представления информации, данных, знания в виде изображения, имеющий целью максимальное удобство их восприятия, понимания и анализа. Результатом визуализации является визуальная модель – зрительно воспринимаемое представление (образ), имитирующее сущность объекта познания. В когнитивной психологии и философии под визуализацией понимается «вынесение в процессе познавательной деятельности из внутреннего плана во внешний план мыслеобразов, форма которых стихийно определяется механизмом ассоциативной проекции» (М. Минский). Показательным является то, что такое определение появилось в ходе научных исследований в области искусственного интеллекта. Весьма близким по смыслу является интерпретация термина, данная А.А. Вербицким: «процесс визуализации представляет собой свертывание мыслительных содержаний, включая разные виды информации, в наглядный образ ... ; будучи воспринятым, образ может быть развернут и служить опорой адекватных мыслительных и практических действий» [35, с.110]. Таким образом, мы можем говорить о внешней визуализации (представлении готового образа, заданного извне) и внутренней визуализации (создании визуального образа внутри и продуцировании его во внешний мир). Кроме того, необходимо различать визуализацию знания, визуализацию информации и визуализацию данных.

Для повышения эффективности процесса обучения используются средства визуализации знания, которые можно определить как «наборы графических элементов и связей между ними, используемые для передачи знаний от эксперта к человеку или группе людей, раскрывающие причины и цели этих связей в контексте передаваемого знания» [95, с. 422].

В настоящее время существует огромное количество методов, принципов и научных подходов к визуализации. Например, Р. Ленглер и М. Эпплер создали весьма интересную и информативную классификацию методов визуализации, которую называли «Периодической таблицей методов визуализации» [194]. В ней систематизировано более 100 методов. В образовательной практике используются как традиционные методы визуального структурирования – различные типы

диаграмм, графики, блок-схемы, графы, сети Петри, деревья (относящиеся в основном к визуализации данных и информации), так и более современные методы, позволяющие визуализировать знания, процессы и стратегии – «дорожные» карты (roadmaps), гиперболические деревья, визуальные метафоры, концептуальные карты, ментальные карты и т.д.

В процессе обучения программированию целесообразно использовать такие средства визуализации знания, как концептуальные и ментальные карты, адаптированные к предметной области программирования. Дадим краткую характеристику этим средствам.

Идея карты понятий или концептуальной карты (concept map) была предложена Д. Новаком, разработавшим метод обучения на их основе и развившем более раннюю концепцию семантической сети [196]. Семантическая сеть состоит из узлов и дуг. Узлы – это концепты (понятийные схемы), дуги – связи между концептами. А.М. Коллинс и М.Р. Квиллиан предложили модель организации знания в памяти человека в виде семантической сети [190]. В модели Коллинса и Квиллиана дуги только одного типа, описывающие связи «быть видом (подклассом) другого понятия». Впоследствии в семантические сети стали включать связи (атрибутные отношения) разного характера. Использовать семантические сети в учебном процессе предложил американский психолог Д. Озьюбел [186]. Он высказал идею о том, что организовать учебные материалы необходимо так, чтобы помочь обучаемому объединить новый материал с ранее изученным материалом, если сравнивать, сопоставлять и находить связи между новыми и уже известными понятиями.

Пример семантической сети некоторых базовых понятий объектно-ориентированного программирования приведен на рисунке 10 (пояснение: все неподписанные связи, обозначенные пунктирными линиями, имеют атрибутное отношение «имеет свойство»).

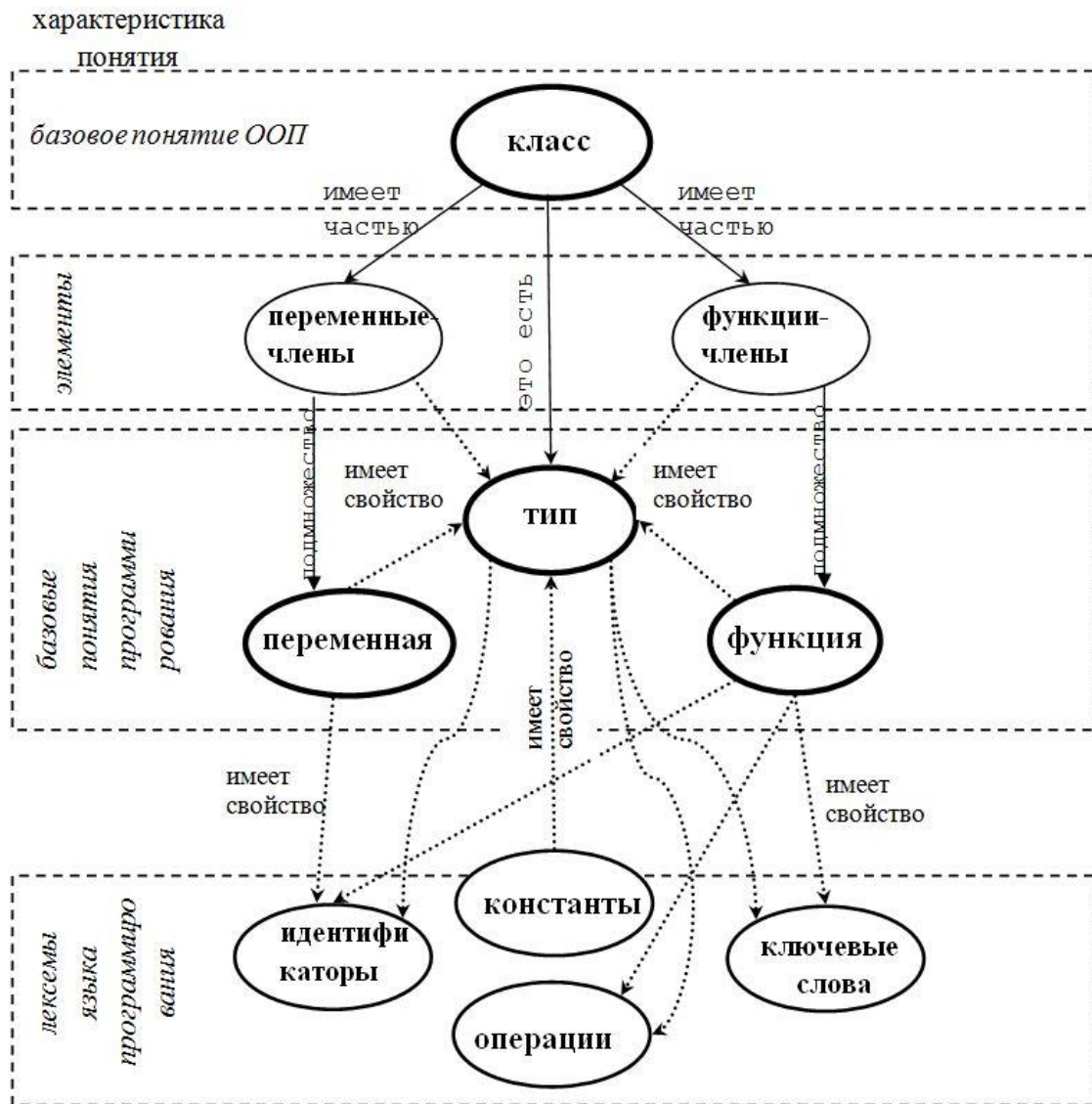


Рис. 10. Семантическая сеть базовых понятий объектно-ориентированного программирования (фрагмент)

Концептуальная карта также предназначена для систематизации множества понятий, которые образуют иерархию и соединены между собой подписанными стрелками, показывающими характер связи. Подписи стрелок могут носить достаточно произвольный характер. Наиболее общие понятия обычно располагают на самом верху карты. Для большей эффективности на концептуальную карту помещают примеры (часто в виде гиперссылок на файлы или веб-сайты), способствующие пониманию отображаемых понятий. Отличие концептуальных карт от семантических сетей заключается в уровне формализации: семантические сети относятся к формальным (машинно-

читаемым) моделям представления знаний, так же как и, например, фреймы, концептуальные графы. Концептуальные карты – пример неформальных моделей, используемых, в первую очередь, для обучения. Пример концептуальной карты «Состав языка программирования C++», являющейся образовательным ресурсом в электронном курсе программирования, приведен в приложении 2.

Ещё одним методом картирования, который может способствовать повышению эффективности обучения программированию, является метод ментальных карт (интеллект-карт). Здесь требуется уточнение данного понятия. Собственно термин «mind mapping» связывается с именем американского психолога и популяризатора науки Т. Бьюзена, предложившим использовать ментальные карты – специальные радиальные схемы, на которых изображены в виде логических связей все вопросы, касающиеся решаемой задачи или изучаемого понятия [32]. Для построения такой карты в большей мере используются ассоциативные связи между включаемыми в карту понятиями. В предметной области программирования целесообразно использовать конкретизированные и частично формализованные ментальные карты в двух направлениях:

- как *обучающая понятийная карта*, на которой представлена в графической форме учебная информация по одному изучаемому понятию;
- как *алгоритмическая ментальная карта*, на которой представлен алгоритм решения задачи, отображающий индивидуальный процесс решения, с возможными тупиковыми ветвями и последующим выбором корректного решения. Такая карта может быть доведена до уровня программной реализации, т.е. стать формальным описанием алгоритма. Примеры понятийных и алгоритмических ментальных карт рассматриваются в §2.1.

Подчеркнем, что такие виды ментальных карт будут соответствовать ментальным (когнитивным) схемам, формируемым в процессе обучения программированию.

Алгоритмическую ментальную карту можно рассматривать как альтернативу формальным способам описания алгоритма. Инструментов для

такого описания достаточно много: блок-схемы, псевдокод, Р-схемы, диаграммы Насси – Шнейдермана, ДРАКОН-схемы, диаграммы UML и т.д. Наиболее известным, классическим способом представления алгоритма являются блок-схемы. Между тем, анализ контрольно-измерительных материалов для ЕГЭ по информатике показывает, что наблюдается отказ от использования блок-схем в курсе школьной информатики, поэтому многие абитуриенты не владеют этим способом формализации алгоритма. Кроме того, практика программирования показывает, что при определённом уровне алгоритмического мышления и овладения каким-либо языком программирования, использование блок-схем становится ненужным. Можно предположить, что для обучаемых с низким уровнем алгоритмического мышления – начинающих программистов следует использовать методику с применением алгоритмических ментальных карт, позволяющих более наглядно, с применением образов, знаков, дифференцированного цветового решения и т.п. привести к решению задачи. Алгоритмическая ментальная карта, составленная конкретным обучаемым, будет служить иллюстрацией процесса мышления данного индивидуума, может подсказать ему недостающую информацию для решения, увидеть решение задачи с другой точки зрения, т.е. стать не только алгоритмическим, но и эвристическим способом решения.

Таким образом, использование концептуальных и ментальных карт во время обучения программированию будет способствовать развитию алгоритмического и системного мышления. Достижение определённого уровня указанных стилей мышления целесообразно принять в качестве результатов обучения программированию студентов-математиков.

1.4. Моделирование результативно-оценочного компонента методической системы обучения программированию студентов-математиков

Традиционные методики программирования оценивают, главным образом, деятельностный аспект программирования – выполнение лабораторной работы, программную реализацию задачи на экзамене и т.д. Несомненно, что владение

навыками практического программирования должно быть критерием оценивания студентов в ходе и по завершению базового курса программирования. Тем не менее, переход высшего образования на компетентностную модель, требует нового подхода к оцениванию результатов образовательного процесса. В этой связи необходимо уточнить понятие «результаты обучения программированию».

В психолого-педагогической литературе существуют разные точки зрения на то, что является результатом обучения. Американский психолог Р. Ганье определяет результаты обучения как «приобретенные способности», т.е. такие состояния нервной системы, которые сохраняются в течение длительных периодов времени и приводят к формированию у индивидуумов устойчивой способности к выполнению специфических видов действий [134, с.1230]. Он рассматривает 5 видов результатов обучения:

- вербальные (декларативные) знания (знания, которые могут представлены (декларированы) в устной, письменной или иной форме);
- интеллектуальные умения (процедурные знания): способности в применении правил и понятий к специфическим ситуациям;
- когнитивные стратегии (управление и регулирование когнитивными процессами: вниманием, восприятием, кодированием и извлечением информации из памяти, мышлением);
- аттитюды, т.е. социальные установки, которыми руководствуется личность в отношении людей, объектов или событий;
- двигательные навыки.

В.И. Байденко видит в образовательных результатах «ожидаемые и измеряемые конкретные достижения студентов и выпускников, выраженные на языке знаний, умений, навыков, способностей, компетенций» [20]. Г.Н. Мотова считает, что результаты обучения «выступают средством выражения уровня компетенции» и являются формулировкой того, что может рассказать, показать, продемонстрировать студент после завершения программы (дисциплины, модуля) [101]. В официальных документах Европейской системы переноса и накопления кредитов (ECTS) под результатами обучения понимаются «наборы компетенций,

выражающих, что именно студент будет знать, понимать и(или) будет способен делать после завершения процесса обучения» [27].

Наиболее сложным представляется вопрос о диагностике результатов обучения. Широко известной является таксономия образовательных целей, предложенная Б. Блумом с коллегами в 1956 г. [188]. В соответствии с ней, образовательные результаты должны оцениваться в трех сферах: когнитивной, аффективной, психомоторной. Для когнитивной сферы психолог предложил оценку результатов обучения по шести категориям, соответствующим когнитивным процессам: знание, понимание, применение, анализ, синтез, оценка. По каждой категории составляется набор дескрипторов, обычно выражаемый глаголами, например: воспроизвести, перечислить, объяснить, применить, вычленишь, составить, оценить и т.д. В соответствии с ними, оцениваются показатели обучающихся. Впоследствии, предложенная таксономия много раз подвергалась уточнению, модернизации, ревизии. Так, в 1999 г. Л. Андерсоном с соавторами была предложена обновленная таксономия, в которой учитывалось различие между декларативным знанием (фактическим и концептуальным), процедурным знанием и метакогнитивным знанием (знанием о процессах мышления и обработки информации с целью эффективного управления этими процессами). Когнитивные способности, подлежащие интерпретации и оценке, в обновленной таксономии были: память, понимание, применение, анализ, оценивание, творчество [159].

Нами был проанализирован ряд зарубежных исследований, в которых делались попытки применить указанные таксономии в предметной области информатики и программирования [193], [201], [203]. Наиболее интересным представляется исследование, проведенное международной группой ученых под руководством U.Fuller. На основе таксономии Л. Андерсона, результаты обучения программированию представлены в виде матрицы, двум размерностям которой соответствуют два диапазона компетенций: способности понимать и объяснять (интерпретировать) готовый программный код и способности разработать самостоятельную программу (рисунок 11).

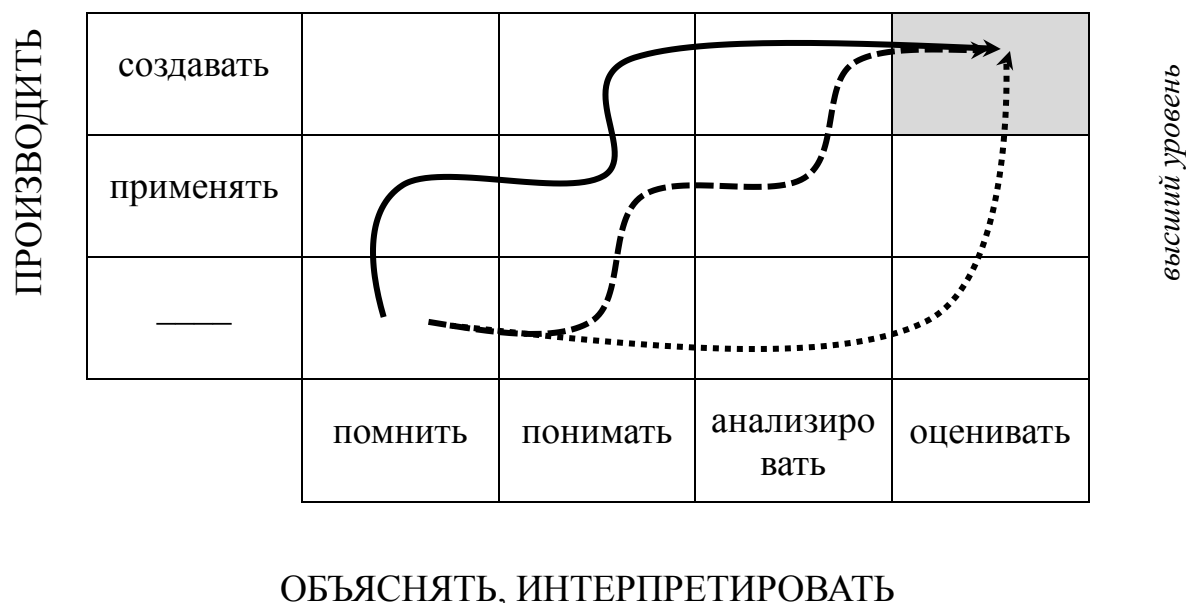


Рис. 11. Двумерная таксономия результатов обучения программированию U. Fuller и др.

Было отмечено, что разные студенты («теоретики» и «практики») могут обучаться программированию, направляясь по разной траектории, в соответствии со своим учебным стилем. Так, студенты-«теоретики» достигают высшего уровня, находящегося в ячейке «оценивать – создавать», двигаясь в направлении: помнить → понимать → анализировать → оценивать → применять → создавать. Студенты, предпочитающие обучение через деятельность, сразу пытаются работать с программным кодом. Понимание всех тонкостей программирования у них может прийти позднее. Также в данной работе было показано, как в ячейках матрицы результатов обучения расположить понятия, связанные с различными видами деятельности программиста: моделирование, разработка, отладка, тестирование и т.д. Например, в верхней правой ячейке «оценивать – создавать» можно расположить такой вид деятельности программиста, требующей высокой квалификации, как рефакторинг (модернизация и оптимизация) программного кода. «Отладка программы» располагается в ячейке «анализировать – применять», «адаптировать программу» – в ячейке «понимать – применять» и т.д. Данная работа, безусловно, имеет теоретическое значение в области диагностики

результатов обучения программированию, но конкретных критериев уровня оценивания этих результатов предложено не было.

Хотя нашей целью является определение понятия «результаты обучения программированию» для бакалавров-математиков, тем не менее, также следует проанализировать профессиональные стандарты, формулирующие квалификационные требования к профессиям ИТ-сферы труда, прежде всего, профессионального стандарта «Программист». Данный стандарт был разработан Ассоциацией предприятий компьютерных и информационных технологий (АПКИТ) и утвержден Приказом Минтруда России в 2013 году [116]. В стандарте представлена функциональная карта профессиональной деятельности программиста, включающая 4 обобщённые трудовые функции, соответствующие уровням квалификации 3–4 (среднее профессиональное образование), 5 – 6 (высшее профессиональное образование). К обобщенным трудовым функциям относятся:

А – Разработка и отладка программного кода.

В – Проверка работоспособности и рефакторинг кода программного обеспечения.

С – Интеграция программных модулей и компонентов, верификация выпусков программного продукта.

Д – Разработка требований и проектирование программного обеспечения.

Для реализации обобщённых трудовых функций необходимы соответствующие знания, в том числе: методов и приёмов формализации и алгоритмизации, программных продуктов для графического отображения алгоритма, алгоритмов типовых задач, синтаксиса языка программирования высокого уровня, среды программирования, методологии разработки программного обеспечения. Необходимы умения и навыки в написании программного кода, определении структур данных и манипулировании с ними, оформлении кода в соответствии с нормативными требованиями, проверке и отладке кода.

Таким образом, анализ компонентного состава обобщённых трудовых функций профессии программиста показал, что приобретение необходимых знаний и умений для осуществления указанных функций, потребует высокого уровня развития алгоритмического и системного мышления.

Обратимся к уточнению понятий «алгоритмическое мышление», «системное мышление» и обоснованию необходимости считать уровень развития данных стилей мышления критерием результативности обучения программированию бакалавров-математиков.

Несмотря на то, что понятие «алгоритмическое мышление» часто используется в научной и научно-популярной литературе, нет устоявшегося, классического определения этого термина. Главная причина этого заключается в том, что данный концепт можно отнести к онтологиям многих наук – информатики, психологии, философии, педагогики. Необходимо отметить, что под «алгоритмическим мышлением» мы будем понимать «алгоритмический способ (метод) мышления». Достаточно распространёнными являются термины «алгоритмический стиль мышления», «алгоритмическая грамотность» (А.Г. Кушниренко, О.Г. Сорока).

М.П. Лапчиком в педагогическую теорию и практику было введено понятие «алгоритмическая культура» [85] . Развитию алгоритмической культуры у школьников посвящены исследования В.М. Монахова, Л.Г. Лучко, А.А. Шрайнера, О.Н. Родионовой, Н.Г. Каратаевой, С.И. Остапенко и др.

Информатика как наука берёт своё начало в математике, поэтому размышления об отличиях алгоритмического мышления от математического и необходимости выделить алгоритмическое мышление как специфический стиль мышления программиста можно найти в трудах многих видных теоретиков программирования – Д. Кнута, Э. Дейкстры, Н. Вирта и др.

А.П. Ершов, опираясь на теорию Ж. Пиаже, ввёл понятие «операционный стиль мышления», которое в его трактовке включало не только собственно алгоритмический способ мышления («умение планировать структуру действий, необходимых для достижения цели при помощи фиксированного набора

средств»), но и способности и умения, которые сейчас называются ИКТ-компетентностью.

Известный американский психолог С. Пейперт использовал понятие «процедурное мышление», видя в нём «инструментальный способ рассуждения» [128]. Там же С. Пейперт указывал, что такое мышление ни в коей мере не следует понимать как линейное, механическое мышление, отвергающее интуицию и творческое мышление. Отметим, что подобный взгляд на алгоритмическое (процедурное) мышление встречается у ряда психологов и в настоящее время. В дальнейшем с именем С. Пейперта связывают появление понятия «вычислительное мышление» (computational thinking). В настоящее время термин получил широкое распространение, в частности, благодаря работам J. Wing, которая называет вычислительное мышление грамотностью 21 века, а алгоритмическое мышление рассматривает как составную часть вычислительного мышления [204].

Определения термина «алгоритмическое мышление» приведены в исследованиях И.Н. Слинкиной, А.И. Газейкиной, Т.Н. Лебедевой, G. Futschek, W. Brown, S. Cooper, W. Dann, R. Pausch.

И.Н. Слинкина рассматривает алгоритмическое мышление как «специфический стиль мышления, предполагающий наличие мыслительных схем, которые способствуют видению проблемы в целом, решению задач крупными блоками с последующей детализацией и осознанному закреплению результатов решения в языковых формах» [154]. Т.Н. Лебедева, уточняя и расширяя данную формулировку, определяет алгоритмическое мышление в виде «... целесообразной (или рациональной) последовательности совершаемых мыслительных процессов с присущей детализацией и оптимизацией укрупненных блоков, осознанным закреплением процесса получения конечного результата, представленного в формализованном виде на языке исполнителя с принятыми семантическими и синтаксическими правилами» [86].

Проведя анализ достаточно большого количества информационных источников, мы приходим к необходимости дать определение понятия

«алгоритмическое мышление» на двух уровнях: предметно-профессиональном и метауровне.

Метауровень определения предполагает, что алгоритмическое мышление можно рассматривать как структурную часть абстрактного мышления человека, позволяющую планировать свои или чужие действия. *Алгоритмическое мышление на метауровне* – это способность выразить процесс решения задачи в виде конечной последовательности элементарных (базовых) действий на естественном, формальном или графическом языке с анализом разных наборов входных данных и вариантов развития событий. На наш взгляд, последняя часть определения подчеркивает нелинейный характер алгоритмического мышления.

Алгоритмическое мышление на предметно-профессиональном уровне – способность понимать, применять, оценивать, оптимизировать и создавать алгоритмы в виде модульно-структурированной формализованной корректной последовательности базовых алгоритмических структур.

Очевидно, что владение алгоритмическим стилем мышления на предметно-профессиональном уровне необходимо и программисту, и математику.

Что касается трактовки понятия «системное мышление», необходимо отметить существенные разногласия исследователей по этому вопросу. Часть исследователей полагает, что нет необходимости говорить о таком феномене как системное мышление, что это – одна из характеристик словесно-логического (абстрактного) мышления, присущего любому взрослому человеку. Другие полагают, что системное мышление – проявление системного подхода к изучению и описанию реального мира. Третьи исследователи рассматривают системное мышление как специфический стиль мышления человека, который необходимо развивать и активизировать. Автор придерживается третьей точки зрения. Приведем ряд определений системного мышления:

«способ мышления, при котором в центре внимания находятся взаимоотношения между частями, взаимодействие которых образует целенаправленное целое» (О`Коннор) [119];

«мышление, учитывающее все положения системного подхода – всесторонность, целостность, многоаспектность, взаимосвязанность, влияние всех значимых для данного рассмотрения систем и связей, как новое видение с направленностью на интегративный синтез знаний, нацеленное на всестороннее познание предмета, отражающее разные стороны, аспекты объектов, на целостность, многомерность бытия» (Ю.В. Федосеева) [170];

«мышление, уровень развития которого при познании человеком мира предметов и явлений объективной действительности позволяет устанавливать взаимосвязи между ними, выявлять закономерности протекания процессов их взаимодействия и развития, прогнозировать это развитие и эффективно решать возникающие при этом проблемы» (Л.И. Шрагина) [182].

Вопросы, связанные с развитием системного стиля мышления, были рассмотрены в исследованиях И.Б. Новика, Э.Г. Юдина, З.А. Решетовой, Л.Я. Зориной, М.А. Науменко, Н.В. Городецкой, А.С. Лобанова, И.А. Сычева, Н.В. Михайловой, Э.М. Сороко, Н.Н. Усковой, Ю.В. Федосеевой, Е.Н. Ляшко, Е.В. Иваньшиной, И. Ю. Асмановой, Г.С. Молоткова, Л.С. Сагатовой, Н.А. Гамовой. Существуют учебные пособия по развитию системного мышления, в том числе, у детей младшего и среднего школьного возраста (Л.Бут Свини, Д. Медоуз, А.В. Кудрявцев, А.И. Никашин, Ю.Г. Тамберг и др.). Что касается диагностики системного мышления, публикаций на эту тему явно недостаточно. В качестве примера приведём методику диагностики системного мышления у школьников, предложенную Л.И. Шрагиной: суть методики в том, что учащийся должен задать как можно больше вопросов по данному понятию или изучаемой теме. Далее вопросы анализируются, разбиваются по категориям, связанными с характеристиками системного мышления, и, по некоторым количественным критериям, делается вывод о широте и глубине системного мышления.

Очевидно, что определение системного мышления следует конкретизировать и дополнить в применении к предметной области программирования. Характеризуя способность к системному мышлению как

универсальную компетенцию личности, выделим следующие показатели *системного мышления* (определение на метауровне):

- задействование в мыслительной деятельности всех базовых операций: анализа, синтеза, сравнения, обобщения, абстракции;
- умение соотнести целое и части;
- способность определить факторы, влияющие на систему, установить причинно-следственные связи и обратную связь;
- способность изменить систему координат (точку зрения), в которой рассматривается данное множество (система);
- умение структурировать систему: выделить и классифицировать подсистемы;
- способность к контекстному рассмотрению системы.

Совершенно очевидно, что перечисленные характеристики также можно отнести к специфическому мышлению математика. К базовым математическим дисциплинам, имеющим непосредственное отношение к изучению систем, относятся системный анализ, теория множеств, теория алгебраических систем. Обращаясь к предметной области программирования, отметим тот факт, что парадигма объектно-ориентированного программирования сформировалась на фундаменте системного подхода к моделированию реального мира. В процедурном программировании системный подход реализован в понятии «подпрограмма (функция)» с помощью механизма передачи параметров (управляющее воздействие на систему) и возвращаемого значения (обратная связь).

Таким образом, принимая во внимание тот факт, что в процессе обучения программированию студентов-математиков происходит формирование алгоритмического и системного мышления, в проектируемой методической системе уместно перенести акцент на развитие этих стилей мышления и соотнести их с критериями результатов обучения программированию. Сформулируем утверждение:

Результаты обучения программированию должны быть выражены в достижении следующих интегративных показателей:

- 1) уровня развития алгоритмического мышления;
- 2) уровня развития системного мышления;
- 3) способности создавать программные продукты на изучаемом языке программирования на основе приобретённых знаний, понимания и умений.

Данные показатели являются взаимопроникающими, носят интегративный характер и могут быть представлены комплексом индикаторов.

Индикаторы уровня развития алгоритмического мышления:

- 1) способность анализировать и специфицировать множество входных и выходных данных задачи;
- 2) способность представить решение задачи через элементарные действия, включающие базовые алгоритмические структуры;
- 3) способность выполнить функциональную декомпозицию задачи;
- 4) способность анализировать и учитывать различные варианты (сценарии) выполнения алгоритма;
- 5) способность оценить эффективность алгоритма;
- 6) способность оптимизировать, улучшить алгоритм.

Предлагаются следующие **критерии оценивания** уровня развития алгоритмического мышления:

- а) начальный уровень: наличие индикаторов 1), 2);
- б) средний уровень: наличие индикаторов 1), 2), 3), 4);
- в) высокий уровень: наличие индикаторов 1), 2), 3), 4), 5), 6).

Индикаторы уровня развития системного мышления:

- 1) способность анализировать и моделировать предметную область задачи;
- 2) способность эффективно организовать данные;
- 3) способность структурировать алгоритм и программное решение задачи;

- 4) способность выделить факторы, влияющие на состояние отдельного объекта и всей системы;
- 5) способность адаптировать решение к другой проблемной области;
- 6) способность интегрировать разработанный алгоритм и программное решение с другими разработками;
- 7) способность объяснить алгоритм и программное решение другим людям на соответствующем уровне;
- 8) способность к интеграции математического знания и теории и практики программирования.

Для оценивания уровня развития системного мышления возможно применить таксономию результатов обучения на основе когнитивного подхода, предложенную психологами J.B. Biggs и K.F. Collis, которая называется SOLO (Структура наблюдаемых результатов обучения) [134, с. 1233]. Согласно этой таксономии, существуют 5 качественно различных уровней мышления: доструктурный, одноструктурный, многоструктурный, реляционный (сетевой), расширительно-абстрактный. Первые три уровня могут характеризовать репродуктивное знание с различной степенью усвоения, четвертый уровень показывает понимание изучаемого предмета, пятый уровень – уровень продуктивного знания и творчества.

Нами предлагается применить указанную таксономию к построению студентами концептуальных карт и сделать это следующим образом. Студент получает задание построить концептуальную карту по данному списку понятий (или по пройденным темам курса программирования). По нашему мнению, достаточно классифицировать 4 уровня:

- *одноструктурный уровень* характеризуется тем, что студент может построить ментальные понятийные карты по отдельным понятиям из предложенного списка;
- *многоструктурный уровень*: по всем понятиям представлена достаточно полная информация, приведены примеры использования;

– *реляционный уровень*: на концептуальной карте присутствуют связи между понятиями, характер которых можно описать вопросом: как и зачем одно понятие используется при определении другого;

– *расширительно-абстрактный уровень* построения концептуальной карты: в данном списке понятий студент видит систему – когнитивную схему более высокого порядка. Например: способен создать алгоритмическую ментальную карту на основе предложенных понятий; обобщает ряд понятий в новое понятие; классифицирует подсистемы.

Наконец, третий показатель результатов обучения программированию, характеризующий деятельностный аспект, – способность создавать программный продукт, может быть представлен следующими индикаторами, которые одновременно являются критериями оценивания этой способности:

- 1) студент способен разработать и отладить (устранить синтаксические ошибки и ошибки времени выполнения) простую программу с базовыми алгоритмическими структурами и типами данных – пороговый уровень;
- 2) студент способен разработать, отладить и тестировать программный код для типовых заданий, на основе стандартных алгоритмов и имеющихся программных решений – базовый уровень;
- 3) студент умеет работать с абстрактными типами данных, самостоятельно разрабатывает алгоритмы и программно реализует их – высокий уровень.

ВЫВОДЫ ПО ГЛАВЕ 1

Подводя итоги теоретического исследования, проведённого в первой главе, можем сделать следующие выводы:

1. Проективно-рекурсивная стратегия обновления целевого, содержательного, технологического и результативно-оценочного компонентов методической системы обучения программированию студентов-математиков даёт возможность превратить базовый курс программирования в средство достижения результатов обучения, которые определены как три взаимосвязанных компонента:

- уровень развития алгоритмического мышления,
- уровень развития системного мышления,
- способность создавать программный продукт на изучаемом языке программирования.

2. Уточнение понятий «алгоритмическое мышление», «системное мышление», выделение определённых индикаторов в каждом компоненте результатов обучения и выбор соответствующих критериев оценивания позволяют предложить модель оценки результатов обучения студентов-математиков программированию с позиций компетентностного и когнитивного подходов.

3. Эффективными средствами достижения требуемых результатов обучения и удобными инструментами их оценивания в показателях уровня развития алгоритмического и системного мышления обучаемых являются когнитивные методы и средства обучения в виде концептуальных и ментальных карт. Проективность и рекурсивность использования и разработки студентами понятийных и алгоритмических ментальных схем по курсу программирования способствуют:

- индивидуализации процесса обучения;
- повышению мотивации студентов к обучению;
- ответственности обучающегося за собственные результаты обучения и продукты учебной деятельности;
- профессионально-ориентированному обучению.

ГЛАВА 2

ПРОЕКТИРОВАНИЕ МЕТОДИКИ ОБУЧЕНИЯ СТУДЕНТОВ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ ПРОЕКТИВНО-РЕКУРСИВНОЙ СТРАТЕГИИ И КОГНИТИВНЫХ ТЕХНОЛОГИЙ

2.1. Структурно-логическая модель обучения программированию будущих бакалавров математических направлений подготовки

С целью целостного представления проективно-рекурсивных компонентов методической системы обучения программированию, определённых в §1.2, необходимо отобразить их в виде структурно-логической модели. Модель представлена на рисунке 12 и показывает состав компонентов и базис обновленной методической системы, логические связи между компонентами системы и внутри них.

В качестве концептуально-методологических основ обновления компонентов методической системы были использованы различные подходы к обучению: системный, деятельностный, когнитивный, компетентностный. Конвергенцию этих подходов обеспечивает проективно-рекурсивная стратегия. На интегрированном концептуально-методологическом базисе происходит проектирование методической системы как единого целого, состоящего из взаимосвязанных компонентов и достижение планируемых результатов обучения программированию.

Вопросам отбора содержания обучения программированию студентов-математиков автор уделит внимание в [9], [11], [13]. В соответствии с заявленными целями обучения, поставленными на дисциплинарном, профессиональном и метадисциплинарном уровнях, дидактическими принципами отбора содержания обучения, были определены основные разделы программирования для изучения и практического освоения на 1 курсе.

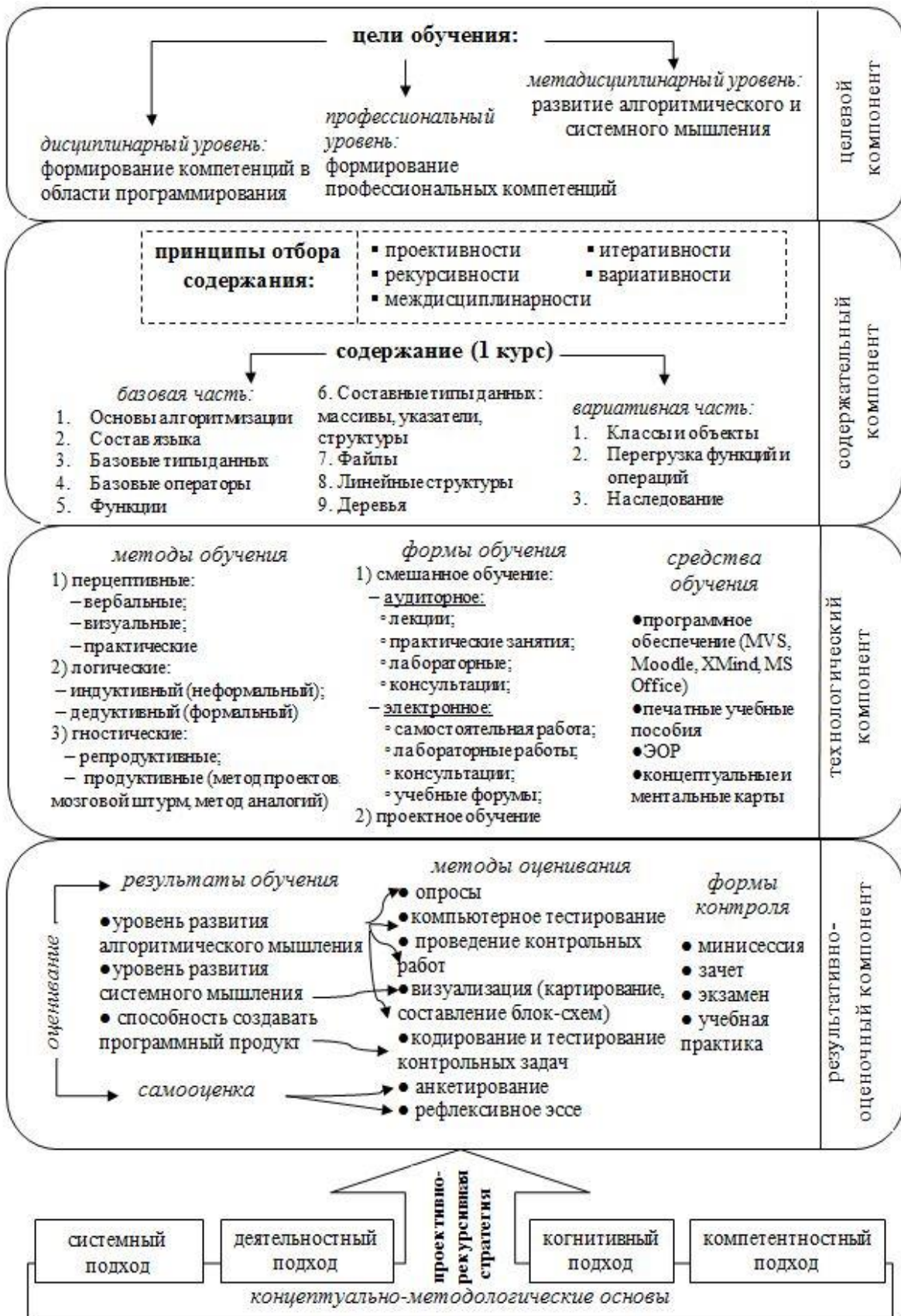


Рис. 12. Структурно-логическая модель обучения программированию

Базовая часть включает следующие темы и соответствующие формы контроля:

- 1) основы алгоритмизации (построение блок-схем и алгоритмических ментальных карт, опросы);
- 2) состав языка (построение понятийных и концептуальных карт);
- 3) базовые типы данных (лабораторная работа, контрольная работа);
- 4) базовые операторы: а) реализация линейной программы, б) программы с ветвлениями, в) программы с циклами (лабораторные работы, понятийные ментальные карты, электронный тест);
- 5) функции: а) параметры и возвращаемые значения (лабораторная работа, тест), б) рекурсивные функции (лабораторная работа);
- 6) составные типы данных: а) массивы (одномерные и двумерные), б) указатели, в) символьные массивы (строки в стиле C), г) структуры (лабораторные работы, тесты, ментальные и концептуальные карты);
- 7) файлы (лабораторная работа);
- 8) линейные структуры данных: стеки, очереди, списки (лабораторные работы, контрольные работы);
- 9) деревья (лабораторная работа, тест).

В *вариативной части* содержательного компонента студентам предлагаются следующие темы, относящиеся к основам объектно-ориентированного программирования:

- 1) классы и объекты;
- 2) перегрузка функций и операций;
- 3) наследование.

По данным темам предусмотрено выполнение лабораторных работ. Отметим, что объектно-ориентированное программирование изучается на 2 курсе, но студенты, успешно освоившие процедурное программирование, имеют возможность двигаться дальше, перейдя к указанным разделам. Кроме того, принцип вариативности реализуется в возможности выбора уровня сложности задания (оцениваемого разным количеством баллов): по всем темам

предусмотрены базовый уровень («А»), повышенной сложности («В»). По некоторым темам уровни варьируются от «А» до «С».

Связь между процедурной (императивной) и объектно-ориентированной содержательными частями происходит с помощью акцентирования внимания на понятии «тип данных». Данная абстракция в единстве с абстракцией «структура данных» (понимаемая как организованная определённым образом совокупность данных) становятся сквозной содержательной линией в рассматриваемой методике обучения программированию. Важность формирования данного концепта (понятийной ментальной схемы) при обучении программированию рассмотрена автором в [6], [7]. Автор языка программирования С++ Б. Страуструп высказался по этому поводу следующим образом: «Программирование на С++ по сути сводится к определению универсальных или зависящих от области приложения типов данных» [160]. Отметим, что в многочисленных учебниках по программированию не всегда чётко определены понятия «тип данных» и «структура данных», что порождает трудности при изучении программирования. В соответствии с пространственно-временной моделью памяти [125], проективно-рекурсивным и когнитивным подходами к обучению, для формирования указанных понятийных схем предлагаются следующие классы заданий:

1) Задачи на анализ входных и выходных данных в постановке конкретной задачи, предлагаемой для решения с помощью программирования (подзадача в задаче программирования). При этом речь идёт не о написании самой программы (это следующий этап), а о том, чтобы обучающийся ответил на ряд вопросов, например:

- а) что является данными в этой задаче;
- б) к какому типу данных (целому, вещественному, символьному) их можно отнести и какие операции над ними производить;
- в) есть ли необходимость структурировать эти данные (например, занести в массив).

Такая постановка задачи («что такое данные», «какие данные и какие операции») помогает сформировать представление (образ) будущих понятийных

схем «структура данных» (отвечает на вопрос «что?») и «тип данных» (отвечает на вопрос «какие?»).

2) Задачи на знание ключевых слов, относящихся к базовым типам данных, их модификацию с помощью спецификаторов и корректное применение ключевых слов в программе. Обучающимся предлагаются фрагменты программ, в которых необходимо:

а) найти и отметить ключевые слова, зарезервированные для базовых типов данных (*int, float, double, char, bool, void*) и спецификаторов типов данных (*short, long, signed, unsigned*);

б) вставить недостающие ключевые слова и обосновать свой выбор.

При решении таких задач у обучающихся одновременно формируются понятийные схемы «переменная» (прообраз будущей понятийной схемы «объект»), «выражение», «операция присваивания». На этом этапе с помощью ключевых слов, являющихся знаковым отражением качественных свойств объектов, изменяется содержание и объём понятийной схемы «тип данных».

3) Задачи на знание правил преобразования типов данных (явное и неявное преобразование типов данных в выражении и при выполнении операции присваивания). Этот класс задач включает:

а) анализ фрагментов программ с целью выяснения значений переменных различных типов;

б) нахождение ошибок, связанных с преобразованием типов данных в предложенном фрагменте программы;

в) самостоятельное написание линейных программ с использованием встроенных типов данных.

При условии успешного решения задач этих трёх классов можно считать, что у обучающегося сформирована понятийная схема «тип данных» и он готов к усвоению понятийной схемы «абстрактный тип данных», т. е. переходу к задачам четвёртого класса.

4) Задачи, реализующие предварительное знакомство с понятийной схемой «класс» (введение в «классы»). Решение задач этого уровня можно представить

как возвращение (рекурсию) к задачам первого уровня, где ответ на поставленные вопросы по анализу данных обучающийся должен дать в терминах «класс» и «объект».

Определим уровни формирования понятийной схемы «тип данных», соответствующие рассмотренным классам заданий:

- 1) пропедевтический уровень (идентификация качественных и количественных свойств данных в определённом образе);
- 2) синтаксически-семантический уровень (формализация понятийной схемы с помощью ключевых слов и спецификаторов базовых типов данных);
- 3) прагматический уровень (корректное использование при написании кода);
- 4) пропедевтический объектно-ориентированный (обобщение понятийной схемы «тип данных», формирование понятийной схемы «класс» как абстрактного типа данных).

Предложенная система заданий, реализованная в учебном процессе, должна помочь образованию, развитию и закреплению у обучающегося программированию фундаментального понятия «тип данных». Отметим, что данная содержательная линия реализована в следующих заданиях:

- лабораторные работы №№ 1, 2, 6, 7, 8, 9, 12, 13, 14, 15;
- контрольные работы №№ 1, 2, 3;
- задание на составление концептуальной карты по материалу 1-го семестра;
- компьютерные тесты по темам «Массивы», «Указатели», «Структуры».

На рисунке 13 представлена логико-структурная схема учебного материала 1-го семестра, отражающая содержание и логические связи изучаемых дидактических единиц.

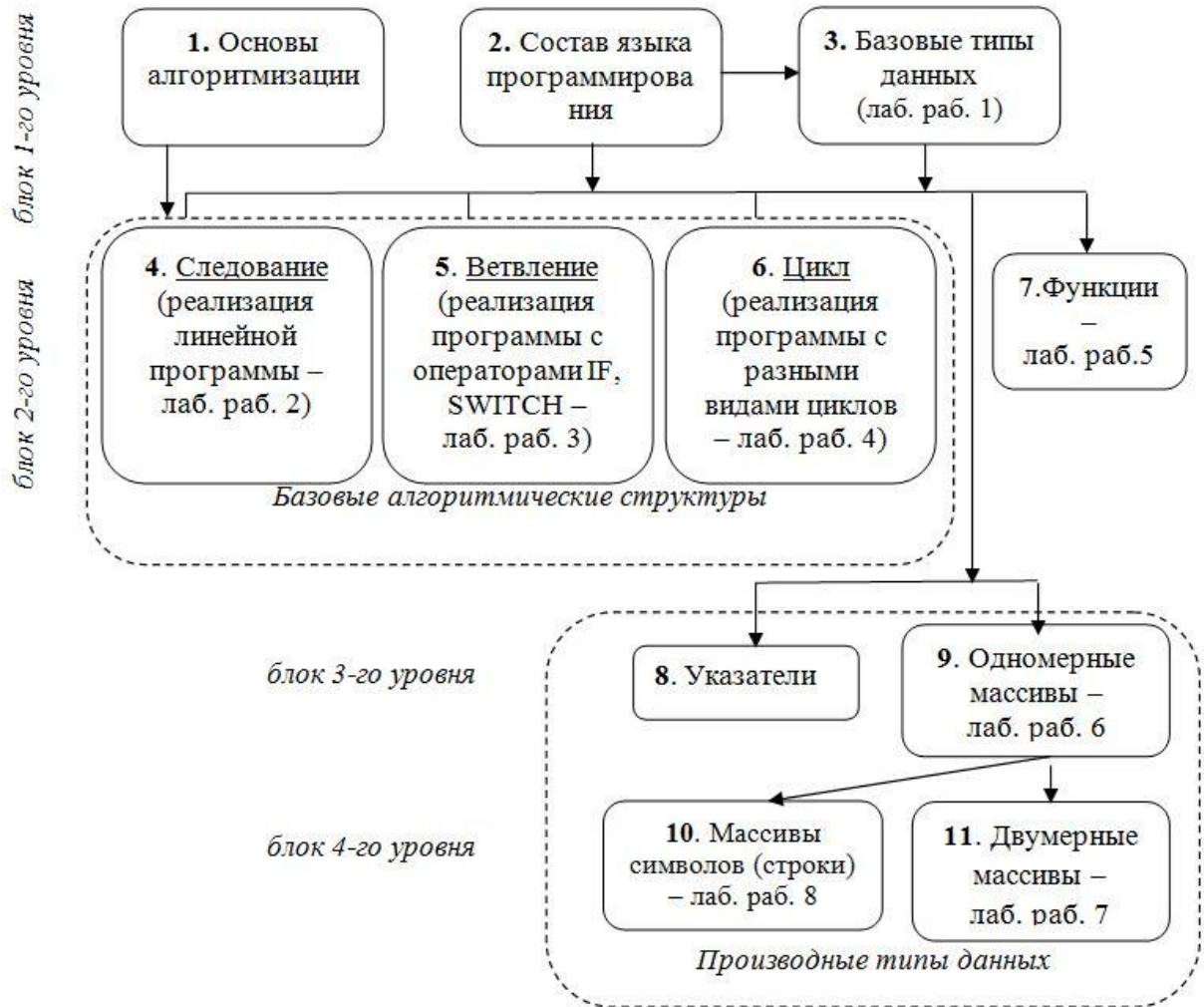


Рис. 13. Структура дидактических единиц содержательного компонента в 1-м семестре

Каждая дидактическая единица организована в виде отдельного модуля – темы в системе Moodle. Отметим, что модули, расположенные в схеме в блоке одного уровня, могут быть изучены и практически реализованы (в виде написания программного кода) в любой последовательности (соблюдение принципа вариативности). В каждой дидактической единице предусмотрено:

1) Изучение теоретического материала в форме презентаций, ментальных карт, концептуальных карт, создание глоссария основных понятий языка программирования C++, возможность ознакомиться с примерами программного кода (файл с методическими указаниями).

2) Выполнение лабораторной работы – самостоятельная разработка программы и защита её на практическом занятии. Задания для лабораторных

работ представляют собой многоуровневую профессионально-ориентированную систему, учитывающую когнитивные особенности мозга и разный уровень подготовки первокурсников как программистов.

3) Мониторинг результатов обучения с помощью выполнения контрольных работ, экзаменационных промежуточных и итоговых тестов, построения блок-схем, ментальных и концептуальных карт.

4) Самоконтроль студентов – наличие тренировочных и обучающих тестов, написанных, в том числе, самими студентами.

Рассмотрим, как дидактические принципы отбора содержания, сформулированные в §1.2, осуществляются на задании, цель которого – проведение тестирования по одной из перечисленных тем.

Постановка задачи:

Разработать, кодировать и отладить программу-тест по теме «...». Программа должна работать в диалоговом режиме, тест содержит 8-10 вопросов. За правильные ответы начисляются баллы. После окончания тестирования выводится сумма набранных баллов и полученная оценка.

Усложняющими элементами задания могут быть:

- а) выбор пользователем количества вопросов;
- б) вывод правильных ответов на неправильный выбор пользователя (обучающий аспект);
- в) случайный вывод вопросов;
- г) организация дружественного интерфейса (приглашение к вводу имени, корректное комментирование результата тестирования и т.д.);
- д) реализация ответа пользователя в виде строки (получение развернутого ответа).

Схема реализации принципов отбора содержания представлена на рисунке 14.

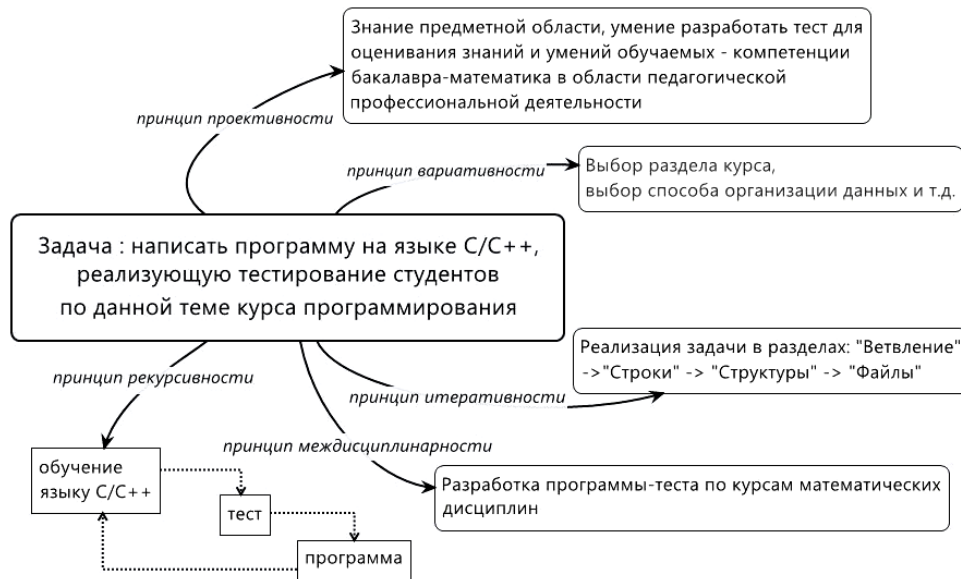


Рис.14. Реализация принципов отбора содержания на задаче тестирования

На рисунке 15 приведен пример работы программы-теста в виде консольного приложения Microsoft Visual Studio (фрагмент). Данный тест по теме «Операторы ветвления» написан студентом в качестве лабораторной работы № 8 («Строки в языке C») и используется как тренировочный тест – обучающий электронный ресурс, представленный в ЭОК «Программирование на языке C/C++».

```

C:\Users\AR\Downloads\uslovie (1).exe
Тест по условным операторам и оператору выбора switch
-----
Введите ФИО и группу.
Иванов Иван ИМ13-03

Вопрос № 1;
Из перечисленных операторов выбрать оператор множественного выбора
1)else
2)while
3)switch
3

Вопрос № 2;
Может ли оператор if применяться без оператора else?
1)Да
2)Нет
1

Вопрос № 3;
Какая из логических операций приведена ?      ||
1)Конъюнкция
2)Отрицание
3)Дизъюнкция
1

Вопрос № 4;
Какая из логических операций приведена ?      &&
1)Конъюнкция
2)Отрицание
3)Дизъюнкция
1

Вопрос № 5;
int a1=3, a2=5;
if(a1=a2) k++; else k--;
Что будет происходить в условном операторе?
1)выполнится k--;
2) выполнится k++;
1

Вопрос № 6;
Что означает запись?
if(a) operator1;
1)Если a!=0, то оператор1;
2)Если a=0, то оператор1;
3)Если a>0, то оператор1;
1

```

Рис.15. Тест по теме «Операторы ветвления», реализованный в виде консольного приложения в Microsoft Visual Studio 2012

Технологический компонент методической системы включает методы, средства и формы обучения.

По определению И.Я. Лернера, метод обучения – это система последовательных и упорядоченных действий преподавателя, организующего познавательную деятельность обучаемых [90]. С позиций компетентностного

подхода к обучению, ставшего определяющим в высшем образовании, мы можем говорить о методе обучения, как системной деятельности преподавателя и обучаемых, направленной на достижение определённых результатов обучения. В структурно-логической модели перечислены *методы обучения*, применяемые в рассматриваемой методике, основанные на классификации Ю.К. Бабанского [65, с. 70]. На основе когнитивного подхода среди методов организации и осуществления учебно-познавательной деятельности он выделяет методы по источнику передачи и восприятия информации, т.е. перцептивные методы. В настоящее время их принято называть вербальными, визуальными и практическими. Весь спектр перцептивных методов активно используется в предлагаемой методике обучения программированию и представлен современными *формами организации учебного процесса*. Кроме традиционных форм аудиторного (контактного) обучения – лекций, практических занятий, лабораторных работ и консультаций, организация электронного обучающего курса в LMS Moodle позволяет эффективно осуществлять самостоятельную работу студентов для выполнения лабораторных работ, онлайн-консультации с преподавателем и учебные форумы для взаимодействия студентов. В соответствии со сложившимися формальным и неформальным подходами в обучении программированию в методике задействованы дедуктивный метод (на лекциях) и индуктивный метод (на лабораторных работах). На практических занятиях, в первую очередь, используются продуктивные методы (проблемно-поисковые по классификации Бабанского), среди которых наиболее востребован метод проектов.

Суть метода проектов (У. Х. Килпатрик, Э. Коллингс, С.Т. Шацкий, П.П. Блонский, Б.В. Игнатьев, Н.Н. Иорданский, В. Н. Шульгин, М. В. Крупенина, П. Ф. Каптерев и др.) заключается в том, что обучающиеся приобретают знания и умения в процессе выполнения практических заданий – проектов. По определению И.Г. Семакина, проектом называется «целенаправленная деятельность, результатом которой является общественно-значимый продукт» [150]. Применение метода проектов в образовательном процессе позволит значительно улучшить результаты этого процесса во всех аспектах: обучающем,

воспитательном, развивающем личностный потенциал обучающегося. Многообразна и роль преподавателя при такой организации учебной деятельности. По мнению Э.Г. Скибицкого, «педагог в процессе выполнения проекта выступает как энтузиаст, специалист, консультант, руководитель, координатор, эксперт. Позиция педагога должна быть открытой, дающей простор самостоятельности обучающихся» [152].

Рассматриваемая методика обучения программированию предполагает реализацию метода проектов в следующих направлениях:

1) Индивидуальные учебные проекты после окончания 1-го курса (учебная практика). Студенты имеют возможность самостоятельно поставить задачу, ориентируясь на свои интересы и способности, или выбрать из предложенного списка задач.

2) Участие студентов в реализации проектов межвузовской кооперации совместно с преподавателями и студентами Красноярского государственного педагогического университета (КГПУ им. В.П. Астафьева) и Сибирского государственного аэрокосмического университета (СибГАУ им. М.Ф. Решетнева) Такая проектная деятельность дает студентам ценный опыт разработки реальных проектов, командной работы, взаимодействия с будущими специалистами в других предметных областях.

3) Привлечение наиболее успешных студентов для выполнения проектов IT-бизнеса в качестве тестировщиков и программистов, что позволит студентам на практике познакомиться с данной профессиональной деятельностью и, возможно, найти будущее место работы.

Обучение программированию невозможно без привлечения современного программного обеспечения в качестве *средств обучения*. В первую очередь, это программное обеспечение, позволяющее осуществлять основные этапы программирования: создание, редактирование, отладку, выполнение и тестирование программного кода в интегрированной среде разработки – программный продукт Microsoft Visual Studio. Студенты имеют возможность работать с лицензионной профессиональной версией пакета, установленной в

компьютерных классах, т.е. во время практических занятий. Для организации самостоятельной работы вне института студенты могут воспользоваться бесплатной версией Visual Studio Express. Кроме того, студентов не ограничивают в выборе компилятора C++ только продуктами компании Microsoft, они могут использовать для выполнения лабораторных работ любые бесплатные компиляторы, доступные для скачивания в Интернете, например, GCC, Dev-C++.

Для эффективной учебной деятельности студентов используется программное обеспечение для создания концептуальных и ментальных карт, блок-схем – XMind, Microsoft Visio. Выбор программного пакета XMind, после проведенного анализа программных продуктов аналогичного класса SmartTools, FreeMind, TheBrain, MindjetMndManager, Aibase и др., был обусловлен следующими факторами:

- данное программное обеспечение является свободно распространяемым (существует профессиональная платная версия);
- имеет простой понятный интерфейс;
- последние версии русифицированы;
- пакет совместим с рядом других программных продуктов;
- созданные карты можно сохранять как в формате xmind, так и в виде изображения;
- имеется возможность воспользоваться облачным сервисом и обратиться к коллекции созданных пользователями XMind ментальных карт в самых разных предметных областях.

Практика показала, что студенты не испытывают никаких затруднений в освоении данного программного обеспечения.

Электронный обучающий курс «Программирование на языке C/C++» разработан и функционирует на базе свободного программного обеспечения LMS Moodle. Moodle – виртуальная образовательная среда, предоставляющая возможность разработки онлайн-курсов и образовательных веб-сайтов. С помощью данной системы обеспечивается:

- разнообразие способов представления учебной информации;

- системный подход к разработке учебного курса;
- конфиденциальность результатов обучения;
- постоянный доступ к образовательному ресурсу;
- интерактивность обучения;
- возможность выбора индивидуальной образовательной траектории;
- мониторинг результатов обучения.

Идейным вдохновителем и первоначальным разработчиком LMS Moodle являлся М. Дугимас [192]. В настоящее время над проектом работает большое число профессиональных программистов. Признание мирового образовательного сообщества способствовало широкому распространению системы в образовательных учреждениях многих стран мира.

Наиболее востребованными компонентами электронного курса в среде Moodle являются элементы, называемые «ресурсы», «задания», «тесты», «опросы», «базы данных», «глоссарии», «форумы».

Теоретические материалы для изучения в контенте курса можно представить с помощью электронных *ресурсов* в разных форматах: текстовой страницы, веб-страницы, ссылки на файл или веб-страницу. Например, рисунок 16 показывает фрагмент презентации по теме «Типы данных в C/C++», которая является ресурсом типа ссылки на файл.

Вещественные типы данных

ключевое слово	размер занимаемой памяти (байт)	диапазон значений
<code>float</code>	4	3.4e-38 / 3.4e38
<code>double</code>	8	1.7e-308 / 1.7e308
<code>long double</code>	10	3.4e-4932 / 3.4e4932

Для представления действительных чисел используются типы данных

- ☐ `float` (одинарной точности),
- ☐ `double` (двойной точности),
- ☐ `long double` (расширенной точности).

[перейти к рассмотрению примера](#)



Рис. 16. Фрагмент презентации по теме «Типы данных в C/C++»

С помощью элемента курса *задание* у преподавателя есть возможность поставить перед обучающимся задачу, на которую студент дает ответ в электронном виде, загружая его на сайт. При этом возможен ответ в виде одного или нескольких файлов (в разных форматах), в виде текста или студент может получить задание, которое предполагается сдавать в аудитории (тип задания – «ответ вне сайта»). Ознакомиться с текстом самостоятельной лабораторной работы и отправить решение – разработанный код программы студент может с помощью данного элемента курса. При этом преподаватель проверяет работоспособность программного кода и может написать отзыв о программе (сделать замечания, подсказать пути исправления ошибок и т.д.). Окончательная защита лабораторной работы происходит в очной форме – в компьютерном классе. Студент обязан не только продемонстрировать работоспособность программы на некотором количестве тестов, но и дать грамотное объяснение работы программного кода, ответить на вопросы. После этого засчитываются баллы за сданную лабораторную работу (выставляются в журнале преподавателя и на электронном курсе). Студент имеет возможность в любое время увидеть свои результаты (набранные баллы) в контенте электронного курса.

Тест – ещё один полезный элемент электронного курса, организованного в среде Moodle. С помощью тестов можно осуществить компьютерное онлайн-тестирование обучающихся, т.е. провести достаточно объективную диагностику знаний. При создании теста возможны следующие типы вопросов.

1) Множественный выбор (один или несколько правильных ответов).
Пример а) – несколько правильных ответов (тест по теме «Циклы»):

Выйти из цикла можно с помощью оператора (выбрать, по крайней мере, один вариант ответа):

- ☐ *goto*
- ☐ *continue*
- ☐ *return*
- ☐ *break*

Пример б) – один правильный ответ (тест по теме «Функции»):

Дан следующий код:

```
#include <iostream>
using namespace std;
void f(int x)
    { int y=5;
      cout << x+y;
    }
void main()
    {int x=5;
      cout << f(x+5);
    }
```

Результатом выполнения данной программы будет:

- 15
- ошибка компиляции
- 10
- 15 15

2) Короткий ответ (вопрос в открытой форме, ответом на который служит строка, набираемая на клавиатуре, это может быть короткий фрагмент программного кода). Пример (тест по теме «Структуры»):

*Объявить структуру с именем **complex**, состоящую из полей **real** и **image**, имеющими тип «вещественный двойной точности». Далее объявить экземпляр структуры с именем **z**. (Все записывать одной строкой)*

3) Альтернативный вопрос, на который дается один из двух вариантов ответов: «верно» или «неверно». Пример (тест по теме «Указатели»):

В программе имеется следующее объявление:

```
int a, *ptr1, **ptr2, ***ptr3;
```

Будет ли оно корректным?

Ответ:

- верно
- неверно

4) Вопрос на соответствие – найти правильный ответ из предложенного списка ответов на каждый вопрос из списка вопросов. Пример (тест по теме «Операторы ветвления»):

Логические операции обозначаются с помощью символов:

логическое «И» – выбрать вариант: && || != ^ !

логическое «ИЛИ» – выбрать вариант: && || != ^ !

отрицание – выбрать вариант: && || != ^ !

5) Числовой вопрос, ответом на который должно быть число, введенное тестируемым с клавиатуры. Пример (тест по теме «Управляющие операторы в циклах»):

Сколько раз выполнится цикл:

```
for ( i=5; i>-5; i -=2)
    if ( i<0 ) continue;
```

Ответ: _____

6) Вопрос-эссе, ответом на который будет развернутый письменный ответ. Ответ должен оцениваться преподавателем вручную. Пример (тест по теме «Файлы»):

С какими типами файлов можно работать средствами языка C++ ?

Чем они отличаются друг от друга?

7) Вложенный вопрос – комбинированный инструмент, позволяющий в текст вопроса вставлять поля для ответов на вопросы типа «Числовой вопрос», «Короткий ответ», «Множественный выбор». Пример (тест по теме «Указатели»):

Имеется следующий фрагмент кода:

```
int a, *ptr_a = &a;
short b, *ptr_b = &b;
double c, *ptr_c = &c;
ptr_a ++;
ptr_b +=5;
```

```
ptrc = ptra + ptrb;
```

После выполнения кода значение ptra увеличится на _____, значение ptrb увеличится на _____, значение ptrc будет (выбрать вариант): увеличится на 6 ; увеличится на 14; ошибка компиляции.

Кроме перечисленных типов вопросов, при разработке теста в среде Moodle можно воспользоваться случайными вопросами, которые формируются случайным образом из имеющихся в данной категории вопросов.

Весьма полезными средствами LMS Moodle являются инструменты, позволяющие анализировать и визуализировать результаты тестирования. Так, с помощью инструмента «Анализ вопросов» можно анализировать качество каждого тестового вопроса и процент тестируемых, которые справились с ним. Например, на рисунке 17 приведен фрагмент таблицы анализа вопросов теста по теме «Функции».

В.#	Текст вопроса	Текст ответа	Частичная оценка	Число ответов	% от-ветов	Индекс легкости	Ср.кв.др. откл.	Индекс диффер.	Кэфф. диффер.
(25277)	прототип : Выбрать синтаксически верный прототип (объявление) функции	int f(int);	(0,33)	5/10	(50%)	70%	0,256	0,53	0,31
		int f(int)	(-0,20)	0/10	(0%)				
		int f(int x);	(0,33)	9/10	(90%)				
		int f(int x)	(-0,20)	1/10	(10%)				
		f(int x);	(-0,20)	3/10	(30%)				
		int f(int x, y);	(-0,20)	1/10	(10%)				
		int f(int x, int y);	(0,33)	10/10	(100%)				
		int f(int x) { int y; y=x; return y; }	(-0,20)	0/10	(0%)				

Рис. 17 Средство Moodle – таблица анализа вопросов теста по теме «Функции» (фрагмент)

В данной таблице индикаторы «индекс дифференциации» и «коэффициент дифференциации» показывают способность данного тестового задания отделить более сильных тестируемых от слабых (неподготовленных) студентов.

Индикаторы могут принимать значения в интервале от -1, когда все тестируемые из сильной группы ответили неправильно, а из слабой группы – правильно, до +1, когда все тестируемые из сильной группы ответили правильно, а из слабой группы – неправильно. Следовательно, если значения данных показателей являются отрицательными, качество данного вопроса должно вызывать сомнение (показанные результаты не соответствуют реальности), и вопрос требует корректировки или исключения из базы тестовых вопросов.

Результаты пройденного теста автоматически визуализируются в виде гистограмм, показывающих распределение тестируемых по баллам. При этом можно получить информацию о результатах данной группы студентов в сравнении со средними показателями всех прошедших данный тест или увидеть общую картину полученных результатов тестирования. На рисунке 18 представлена гистограмма результатов тестирования всех участников курса по теме «Указатель».

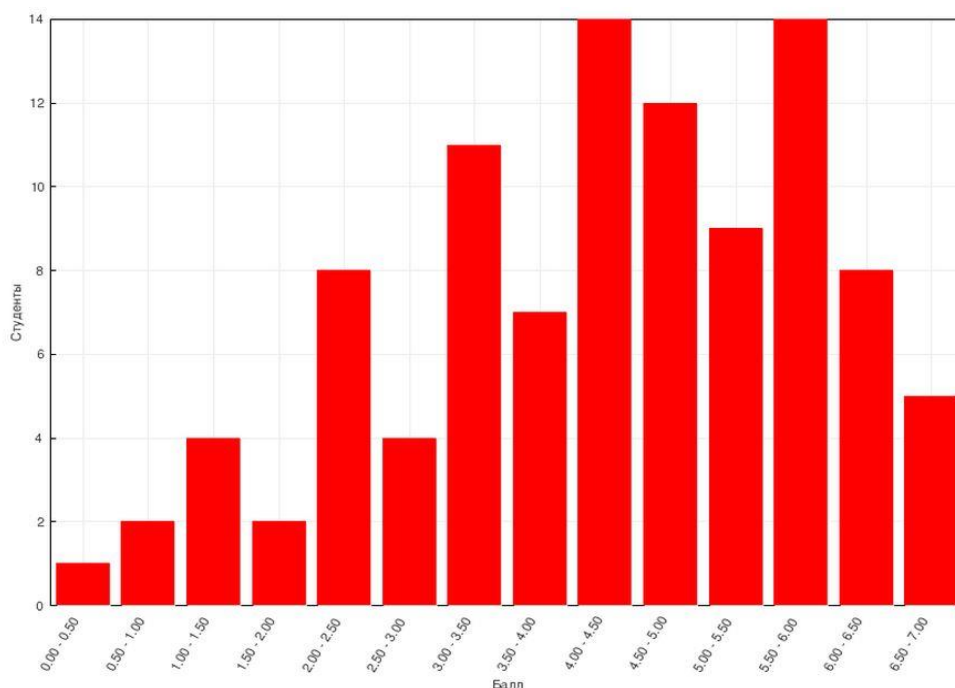


Рис. 18 Результаты тестирования студентов всех групп по теме «Указатель» за все годы: 2012/13; 2013/14; 2014/15.

Рассмотрим когнитивные средства обучения – концептуальные и ментальные карты, представленные в контенте электронного курса. Вопросы,

связанные с теоретическим обоснованием и практической разработкой концептуальных и ментальных карт, рассмотрены автором в [10], [16], [19].

Пример понятийной ментальной карты по теме «Указатели» приведен в приложении 3. На карте, выполненной с помощью программного продукта XMind, в наглядной и доступной форме отражена основная информация по данной теме: определение понятия «указатель»; способы и примеры объявления, инициализации, возможных операций над указателями, использования. Описание даётся вначале на метаязыке, затем непосредственно на языке C++.

В приложении 4 дан пример понятийной ментальной карты по теме «Рекурсивные функции», которая построена рекурсивно: примеры с кодами программ находятся в свёрнутом состоянии (кнопки с символом «+»), присутствует гиперссылка на ментальную карту «Функция». Отметим, что на ментальной карте «Функция» целесообразно поставить гиперссылки на карты, связанные с понятием «Функция»: «Тип данных» (тип функции и типы параметров), «Указатель» и «Ссылка» (связано со способами передачи параметров в функцию).

На рисунке 19 изображена алгоритмическая ментальная карта, составленная для решения задачи:

С клавиатуры вводятся длины 3-х сторон треугольника. Выяснить, является ли данный треугольник прямоугольным.

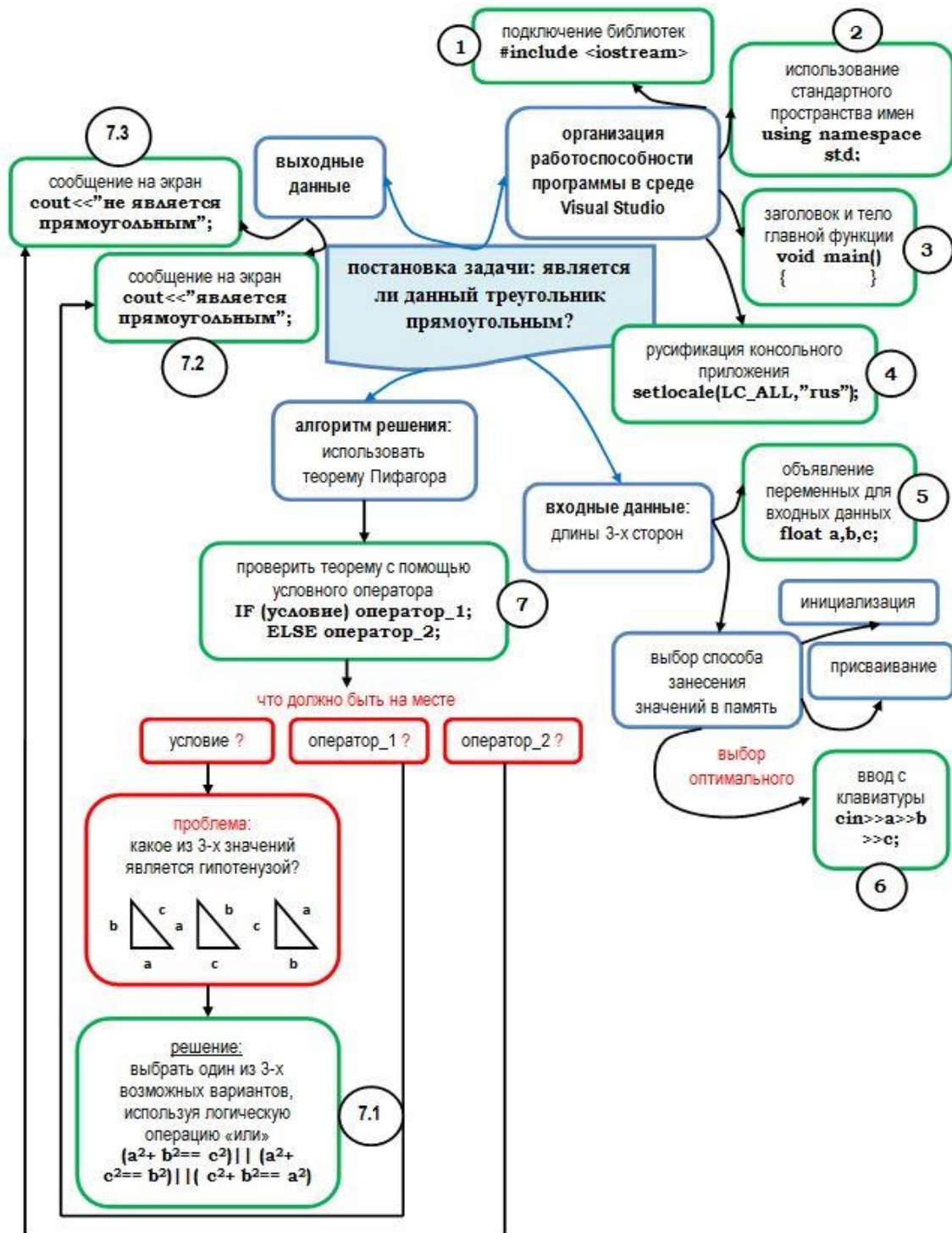


Рис. 19. Пример алгоритмической ментальной карты «Решение задачи о прямоугольном треугольнике»

Дадим пояснения к рисунку. Ментальная карта в наглядной форме демонстрирует возможный алгоритм решения поставленной задачи с момента анализа условий задачи до написания кода (решение доведено до полностью формализованного представления). Для усиления чувственного восприятия и

ассоциативного мышления используются графические изображения треугольников, различное цветовое оформление: синим цветом выделены исходные данные и основные шаги решения задачи, красным цветом – возможные затруднения в процессе решения, зеленым цветом – принимаемые к исполнению решения. Так, после принятия решения о возможности использовать теорему Пифагора, возникает вопрос о том, какая сторона из трёх является гипотенузой, какие стороны являются катетами (если треугольник нарисовать, то такой вопрос возникнет обязательно). Ответом на поставленный вопрос будет решение о проверке всех возможных сочетаний гипотенузы и 2-х катетов (порядок катетов будет неважен): {a, b, c}, {a, c, b}, {c, b, a}. Так как представленная ментальная карта является алгоритмической, последовательность решения задается с помощью нумерации (цифры в кружках). Если рассматривать ментальную карту как дерево решения задачи, то собственно решение можно получить, сделав обход нумерованных листьев дерева по часовой стрелке. Использовать данную алгоритмическую карту возможно при изучении темы «Операторы ветвления».

Кроме ментальных средств обучения, таких как концептуальные и ментальные карты, в учебном процессе находят применение электронные обучающие средства ИКТ, ставшие уже традиционными, – презентации, flash-анимации. Эти средства обучения, наряду с концептуальными и ментальными картами, дают возможность визуализировать знание в предметной области программирования. Исходя из разделения знания на декларативное (знание – сообщение о чём-то) и процедурное (знание о том, как действовать) (Г. Райл) [134], в курсе программирования целесообразно использовать следующую схему:

декларативное знание → визуализация понятий программирования и структур данных;

процедурное знание → визуализация алгоритмов.

На рисунке 20 отображен кадр flash-анимации, которая визуализирует сортировку массива методом выбора. Анимация является электронным образовательным ресурсом, представленным в ЭОК «Программирование на языке C/C++».



Рис. 20. Визуализация сортировки массива с помощью flash-анимации

Возможности современного программного обеспечения, практически повсеместный доступ в Интернет, позволяют самостоятельно создавать динамическую визуализацию алгоритмов и пользоваться готовыми сетевыми ресурсами. Задача преподавателя – информировать и сориентировать студентов для поиска наиболее удачных и эффективных анимаций, реализующих изучаемые алгоритмы. В качестве примеров рекомендованных к использованию интернет-ресурсов можно привести портал AlgoViz (<http://algoviz.org>) и бесплатный программный продукт ALViE, дающий возможность ознакомиться с большим количеством (почти 40) анимаций алгоритмов и структур данных (ссылки на них даны в контенте электронного курса) [197] , [191] . Если же привлекать к созданию визуализации алгоритмов наиболее подготовленных студентов, можно предположить, что существенно усилится мотивация студентов к изучению теории и практики программирования.

Таким образом, для визуализации декларативного знания мы предлагаем применять концептуальные карты, понятийные ментальные карты, презентации.

Для визуализации процедурного знания – алгоритмические ментальные карты, классические блок-схемы, flash-анимации. В этом случае, вариативность и возможность выбора форм предъявления учебной информации позволяет учитывать индивидуальные психические особенности и стили учебной деятельности студентов, что в полной мере соответствует когнитивному подходу к обучению.

Для реализации результативно-оценочного компонента методической системы обучения программированию был создан комплекс контрольно-измерительных материалов, включающий:

- задания для лабораторных работ (используются для оценивания способности создавать программный продукт);
- электронные тесты и задания для контрольных работ (используются для определения уровня алгоритмического мышления).

М.И. Холодная определяет стили учения обучающихся как «учебные стратегии, которые характеризуют ответные действия индивидуума на требования конкретной учебной ситуации» [172]. При этом, говоря о необходимости соотношения индивидуального стиля учения обучающегося и адекватной этому стилю формы контроля, она ссылается на работы Б.Лу Ливер, систематизировавшей разные типы учебных предпочтений учащихся. Рассмотрим, как, согласно Ливер, соотносятся разные стили учения обучающихся и применяемые в нашей методике формы контроля (таблица 2).

Таблица 2

Соответствие стиля учения формам контроля

Стиль учения обучающегося	Форма контроля
аналитический	вопросы с множественным выбором в компьютерных тестах; лабораторные работы «Функции», «Массивы»
синтетический	открытые вопросы и эссе в компьютерных тестах;

	лабораторные работы; написание программ (фрагментов программ) в контрольных работах; проекты
импульсивный	компьютерные тесты (ограничение по времени)
рефлексивный	лабораторные работы (продолжительное время выполнения)
визуальный	контрольные работы; компьютерные тесты; создание концептуальных и ментальных карт
аудиальный	защита лабораторных и экзаменационных работ, проектов
кинестетический	лабораторные работы; проекты; создание концептуальных и ментальных карт

Таким образом, широкий спектр форм контроля, представленный в методике обучения программированию, дает возможность учитывать учебные стили студентов, обусловленные их когнитивными особенностями.

В электронных тестах и контрольных работах применялись задания типа:

- 1) определить результат выполнения фрагмента программы;
- 2) исправить ошибки в приведенном фрагменте программы;
- 3) вставить недостающие ключевые слова в операторах или фрагментах кода;
- 4) расставить операторы в правильной последовательности (логика выполнения программы);
- 5) убрать из фрагмента кода лишние описания или операторы (избыточность кода);
- 6) написать фрагмент кода, реализующий некоторую задачу.

С помощью подобного рода заданий в тестах, разработанных по изучаемым темам курса программирования, фиксируются индикаторы уровня развития алгоритмического мышления, перечисленные в §1.3. Пример вариантов заданий в

контрольной работе по темам «Состав языка программирования», «Типы данных» приведен в приложении 5.

2.2. Процессуальная модель обучения программированию студентов-математиков

Методическая система обучения представляет собой единство структурной и функциональной организации. Э.М. Сороко понимает под функциональной организацией системы «известный порядок и последовательность в выполнении системой необходимых ей действий, направленных на достижение ближайших и долгосрочных целей» [159]. Функциональную организацию системы можно представить в виде процессуальной модели.

Процессуальная модель методической системы, являясь образом этой системы, и отражая отношения с субъектами образовательного процесса и образовательной средой, должна удовлетворять, согласно [106], трём необходимым требованиям:

- ингерентности (согласованности со средой, наличия определённого интерфейса со средой);
- простоты (игнорирования несущественных в данном контексте характеристик системы);
- адекватности (возможности достичь поставленной цели в соответствии со сформулированными критериями).

Наличие перечисленных критериев при разработке процессуальной модели обучения программированию представим с помощью схемы (рисунок 21).

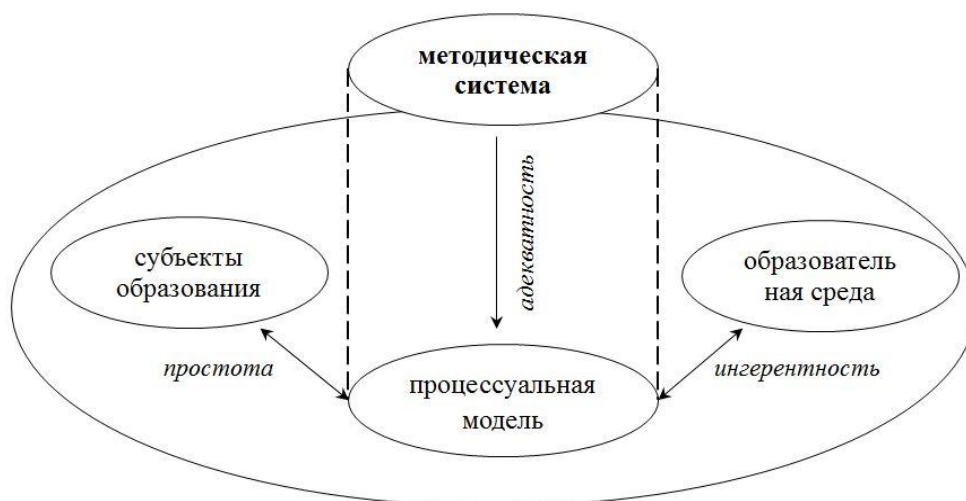


Рис. 21. Критериальная основа разработки процессуальной модели обучения программированию

С позиций системного подхода предлагаемая методика обучения – это процесс взаимодействия субъектов и объектов образования. Наглядным образом это отражено в виде процессуальной модели обучения программированию, представляющую собой пространственно-временную схему (рисунок 22).

На субъектно-объектно-временной плоскости показаны временные срезы, иллюстрирующие состояние динамической системы, включающей субъекты и объекты педагогического процесса. Компонентами этой системы являются:

- группа студентов, обучающихся программированию (это в свою очередь, подсистема рассматриваемой системы);
- преподаватель или группа преподавателей, реализующих методику проективно-рекурсивного обучения программированию;
- подсистема методической системы, включающая содержательный, технологический и результативно-оценочный компоненты.

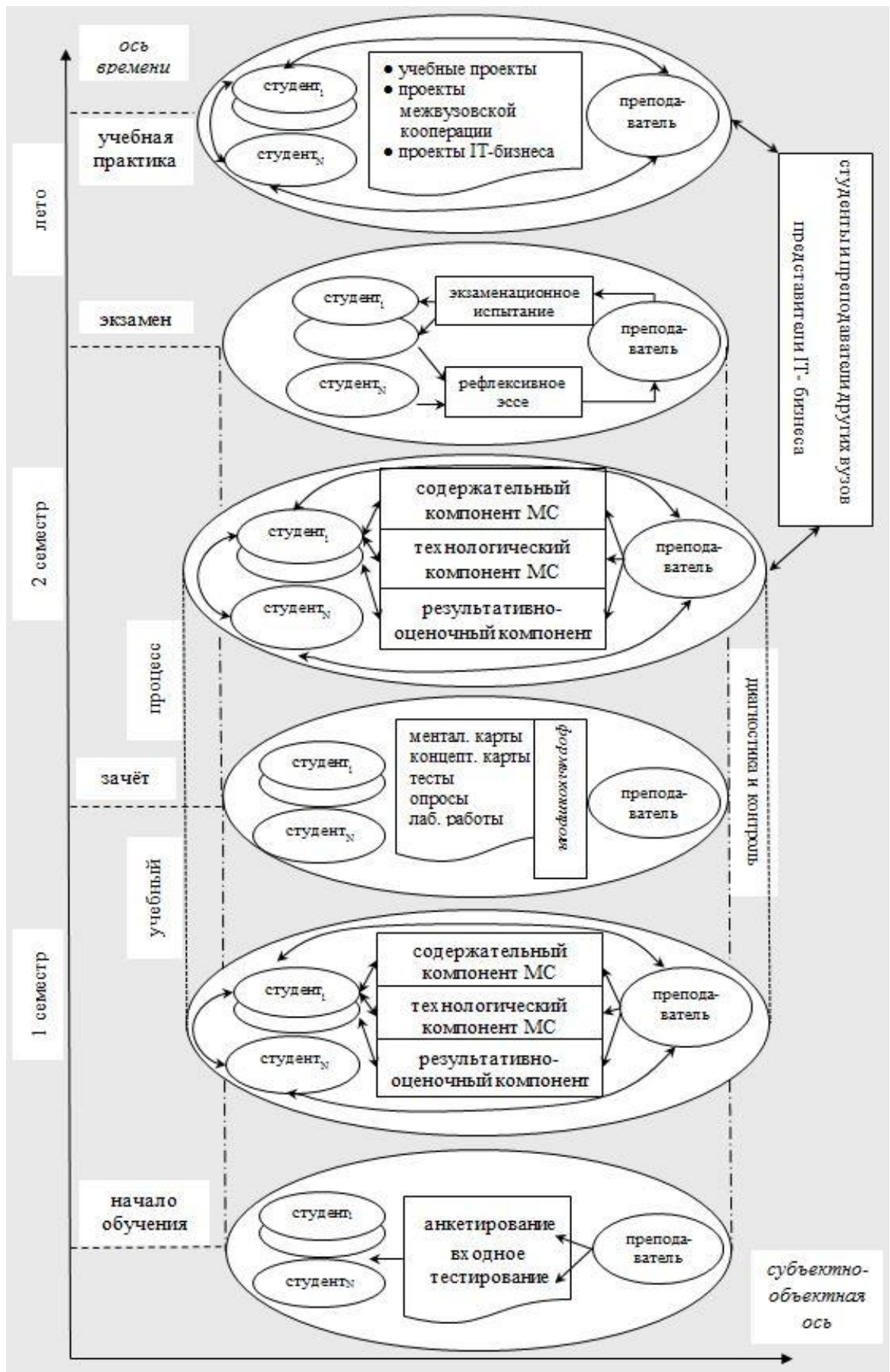


Рис. 22. Процессуальная модель обучения программированию

Целевой компонент не представлен в процессуальной модели в силу ограниченного размера изображения. Кроме того, целевой компонент можно считать статическим элементом данной модели. Ещё одно упрощение в предложенной модели – в качестве внешних взаимодействий представлены только отношения, возникающие в процессе работы над проектами межвузовской кооперации и проектами IT-бизнеса, имеющими значение для реализации методики обучения программированию. Во внимание не приняты внешние связи с организационной, материально-технической, юридической, финансово-экономической и другими структурами информационно-образовательной среды вуза.

Системообразующими факторами рассматриваемой системы являются студент и преподаватель. Отношения между элементами системы носят субъект-объект-субъектный характер. На оси времени, показывающей протекание учебного процесса во времени, выделены контрольные точки: начало обучения, зачет, экзамен, учебная практика. В течение учебного процесса характер взаимоотношений меняется. В начале обучения между студентами внутри группы только начинают выстраиваться взаимосвязи, преподаватель формирует представление о каждом студенте с помощью анкетирования, наблюдения, входного тестирования. Затем появляются новые связи: между каждым студентом и преподавателем, между студентами внутри группы. Осуществление проектной деятельности обуславливает появление внешних связей со студентами и преподавателями других вузов и возможными работодателями из сферы IT-бизнеса. Взаимодействие между студентами и преподавателем происходит как непосредственно на занятиях (аудиторное обучение), так и с помощью электронного обучающего курса (ЭОК), разработанного в среде LMS Moodle.

Отметим двунаправленный характер связи между студентами и ЭОК: как уже говорилось ранее, студенты причастны к наполнению контента курса через реализацию проективно-рекурсивной стратегии обновления компонентов методической системы.

Важный момент – влияние студентов на результативно-оценочный компонент. Например, с помощью такого инструмента Moodle как опрос, студент (также как и любой участник курса) имеет возможность оценить разработанные другими студентами концептуальные и ментальные карты, блок-схемы, тем самым внося вклад в комплексную оценку каждого студента за курс.

Итоговым испытанием по программированию за 1 курс является экзамен. При этом оценка, выставляемая по дисциплине, является комплексной: учитываются результаты экзамена, промежуточного контроля (минисессии), выполнения лабораторных работ, тестов, контрольных работ, построения концептуальных и ментальных карт. В конце 1 курса студентам предлагается написать рефлексивное эссе с целью оценить собственные успехи в изучении программирования, оценить электронный курс, сделать какие-либо замечания или внести предложения по улучшению курса. Подобная практика может использоваться для оценивания аксиологического компонента результатов обучения программированию. Но, по нашему мнению, это целесообразнее осуществлять на старших курсах подготовки бакалавров-математиков, когда студенты имеют уже сформированное, чёткое представление о будущей профессиональной деятельности, видят перспективы занять своё место на рынке труда (многие уже трудоустроены), и, следовательно, могут оценить значение курса программирования для своего образования.

Согласно учебной программе, соответствующей ФГОС ВПО 3-го поколения, у студентов проходит летняя учебная практика по программированию, в ходе которой они должны выполнить самостоятельный проект. В Федеральном законе РФ «Об образовании в Российской Федерации» дано следующее определение: «практика – вид учебной деятельности, направленной на формирование, закрепление, развитие практических навыков и компетенции в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью» [111]. В учебном плане, соответствующем стандарту 2-го поколения, была предусмотрена дисциплина «Практика на ЭВМ», по которой в течение четвертого семестра обучения студенты выполняли самостоятельные

учебные проекты в компьютерном классе, консультируясь с преподавателем. Сейчас это полностью самостоятельная работа студентов по окончании 1 курса обучения, результаты которой они обязаны предъявить в третьем семестре (формой контроля выполнения является зачет). Таким образом, важно подготовить студентов к такой деятельности в процессе обучения программированию, сформировать у них не только необходимые умения для её осуществления, но и обеспечить мотивационно-ценностные установки. Методика обучения программированию, актуализирующая проективно-рекурсивную стратегию, способна удовлетворить указанным требованиям.

Как уже было отмечено ранее, студенты имеют возможность разработать и реализовать проекты учебной практики в трёх направлениях.

Первое направление предполагает выполнение учебных проектов. Задания для них формулируются преподавателем. Например, ниже приведены несколько вариантов заданий, которые предлагались для выполнения студентам в 2012/13 учебном году:

ВАРИАНТ № 1.

а) Создать класс «КВАДРАТ», поле класса – длина стороны. Предусмотреть в классе методы вычисления и вывода информации о фигуре – диагональ, периметр, площадь. Перегрузить 3–4 операции.

б) Создать производный класс – «ПРАВИЛЬНАЯ КВАДРАТНАЯ ПРИЗМА». Добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода информации о фигуре.

в) Написать программу, демонстрирующую работу с этими классами: дано N квадратов и M призм, найти квадрат с максимальной площадью и призму с минимальной диагональю.

г) В программе обязательно наличие меню и ввод–вывод информации в текстовый файл.

ВАРИАНТ № 2.

а) Создать класс «РАВНОСТОРОННИЙ ТРЕУГОЛЬНИК», поле класса – длина стороны. Предусмотреть в классе методы вычисления и вывода

информации о фигуре – высота, биссектриса, периметр, площадь. Перегрузить 3-4 операции.

б) Создать производный класс – «ТЕТРАЭДР». Добавить в класс вычисление объёма, двугранного и телесного углов, перегрузить метод вычисления площади.

в) Написать программу, демонстрирующую работу с этими классами: дано N равносторонних треугольников и M тетраэдров, найти равносторонний треугольник с минимальной площадью и тетраэдр с максимальной высотой.

г) В программе обязательно наличие меню и ввод-вывод информации в текстовый файл.

Предложенные задания для учебной практики (базовый уровень) должны были закрепить знания и умения по основам объектно-ориентированного программирования, которые студенты получили в конце 1-го курса. Кроме того, для реализации принципа междисциплинарности, поставленные задачи относились к предметной области аналитической геометрии.

В 2013/14 учебном году был проведен эксперимент: студентам было предложено самостоятельно сделать постановку задачи, исходя из своих интересов (прикладная задача могла быть из любой области) и возможностей как программистов. Те студенты, которые затруднялись с самостоятельной постановкой задачи, могли выбрать задание из набора, предложенного преподавателем. Постановки задач относились к следующим областям:

1) Классические математические дисциплины.

Например:

Составить программу для решения системы линейных уравнений с n неизвестными методом Гаусса. Система записана в виде:

$$Ax = B, \text{ где } X = \|x_i\|, A = \|a_{ij}\|, B = \|b_i\|$$

Пользователь задает с клавиатуры матрицу коэффициентов A и правую часть B , на экран выводится вектор решения.

2) Прикладные математические задачи.

Например:

Составить программу для зашифровки и расшифровки данного файла с помощью шифра Цезаря. Шифрование выполняется следующим образом: сдвинуть каждую букву каждого слова файла на заданное число позиций. Например: задано число 2, тогда слово «бит» зашифруется так: «б» + 2 = «г», «и» + 2 = «к», «т» + 2 = «ф». Получится слово «гкф». Необходимо зашифровать файл, вывести его на просмотр, затем сделать обратное преобразование и вывести на экран расшифрованный файл.

3) Задачи программирования (структуры данных и алгоритмы).

Например:

Построить дерево поиска по серии 100 чисел, полученных с помощью датчика случайных чисел. Необходимо определить разницу между максимальным уровнем и минимальным уровнем узлов построенного дерева. Программа должна печатать таблицу разниц высот для N серий, число N вводится с клавиатуры.

4) Игровые задачи. Например:

Написать программу «Игра в города». Названия городов заданы в текстовом файле. Правила игры: первым элементом цепочки может быть любой город, каждый следующий начинается с последней буквы предыдущего города. Программа должна выводить на экран цепочку максимальной длины из названий городов.

Результаты проведенного эксперимента с самостоятельной постановкой задач и анализа предпочтений студентов приведены в §2.3.

Второе направление в организации учебной практики по программированию заключается в привлечении наиболее успешных студентов к разработке и реализации проектов межвузовской кооперации. Такая учебная деятельность студентов носит научно-исследовательский характер и требует определённых временных затрат. Поэтому студенты принимали участие в ней в течение всего учебного года. Кроме студентов-математиков в этой проектной деятельности участвовали студенты и преподаватели Красноярского

государственного педагогического университета и Сибирского государственного аэрокосмического университета. Результаты этого сотрудничества представлены в публикациях [14], [15], [17], [18]. Для обоснования целесообразности совместной проектной деятельности студентов разнопрофильных вузов мы применили компетентностный подход, анализируя ФГОС ВПО 3-го поколения по направлениям подготовки бакалавров «010100. Математика», «050100. Педагогическое образование», «161101. Системы управления летательными аппаратами» [112], [113], [114]. Наглядно сопоставить компетенции будущих бакалавров по разным направлениям можно с помощью диаграмм Эйлера-Венна (рисунок 23).

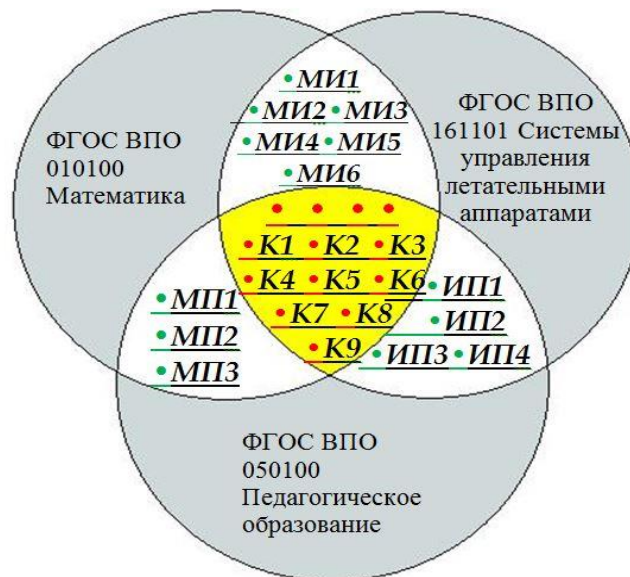


Рис. 23. Диаграмма Эйлера-Венна – пересечение множеств компетенций бакалавров разнопрофильных вузов

На диаграмме с помощью трех кругов изображены множества компетенций будущих математиков, инженеров и педагогов. В пересечения этих кругов входят множества компетенций, общих для математиков, педагогов и инженеров (центральный сегмент). Ещё три сегмента – это пересечения множеств компетенций математиков и инженеров, математиков и педагогов, инженеров и педагогов. Следует отметить, что, т.к. в стандартах по разным направлениям

подготовки компетенции обозначены по-разному, они были перенумерованы следующим образом: МИ – общие компетенции для математиков и инженеров, МП – общие компетенции для математиков и педагогов, ИП – общие компетенции для инженеров и педагогов. В сегмент общих компетенций для всех трёх направлений подготовки (обозначенных буквой К) попали 9 следующих компетенций:

К1 – навыки межличностных отношений, готовность работать в команде, способность к кооперации с коллегами, в том числе и над междисциплинарными, инновационными проектами;

К2 – принятие различий и мультикультурности, толерантность;

К3 – способность к анализу, синтезу, восприятию информации;

К4 – применение методов теоретического и экспериментального исследования, исследовательские навыки;

К5 – базовые знания в областях информатики и современных информационных технологий, владение методами и способами получения, хранения, переработки информации;

К6 – способность работать с информацией в компьютерных сетях;

К7 – соблюдение требований информационной безопасности, защита государственных интересов;

К8 – способность к письменной и устной коммуникации на русском языке;

К9 – навыки публичной речи, ведения дискуссии и полемики, представления собственных и известных научных результатов.

Компетенции МИ, МП, ИП расшифрованы в таблице 3.

Таблица 3

Общие формируемые компетенции будущих математиков, инженеров, педагогов

компетенции математиков и инженеров	компетенции математиков и педагогов	компетенции инженеров и педагогов
-------------------------------------	-------------------------------------	-----------------------------------

МИ1–самостоятельное построение алгоритма и его анализ, выбор методик и средств решения задачи;	МП1 – владение основами речевой профессиональной культуры;	ИП1 – постановка исследовательских задач и выбор пути их достижения;
МИ2–умение находить, анализировать и контекстно обрабатывать научно-техническую информацию;	МП2–способность реализовать учебные программы базовых и элективных курсов в образовательных учреждениях;	ИП2 – осознание социальной значимости профессии, обладание мотивацией к осуществлению профессиональной деятельности;
МИ3– способность приобретать новые знания, в том числе в новых областях, непосредственно не связанных со сферой деятельности;	МП3 – владение основами педагогического мастерства.	ИП3 – способность использовать базовые положения математики, естественных наук при решении социальных и профессиональных задач;
МИ4–способность составлять практические рекомендации по использованию результатов научных исследований;		ИП4 – использование знаний гуманитарных, социальных и экономических наук при решении социальных и профессиональных задач.
МИ5–владение методами математического моделирования;		
МИ6–способность разрабатывать варианты решения проблемы, проводить		

системный анализ этих вариантов		
---------------------------------	--	--

Таким образом, почти половина компетенций математиков, более половины компетенций педагогов, треть компетенций инженеров (в силу большого количества узкопрофессиональных компетенций) попадает в область пересечения множеств компетенций. Это позволяет сделать вывод о целесообразности привлечения студентов к разработке и реализации проектов межвузовской кооперации с целью более эффективного формирования общекультурных и профессиональных компетенций, необходимых выпускнику высшей школы для успешной реализации его личностных и профессиональных качеств. Перечислим, что может послужить мотивацией студентов к участию в межвузовских исследовательских проектах:

- получение опыта работы в команде;
- реальное понимание организации процесса научного исследования в разных предметных областях;
- конкретные результаты исследований для выполнения курсовых и дипломных проектов;
- получение опыта самостоятельной работы при выполнении части коллективного задания;
- формирование навыков презентации результатов выполненной работы.

Формы взаимовыгодного сотрудничества студентов и преподавателей разнопрофильных вузов могут быть достаточно разнообразны. В рамках рассматриваемой методики обучения программированию студентов-математиков реализовать сотрудничество возможно при совместной разработке электронных образовательных ресурсов. Например, при изучении дисциплин «Информационно-измерительные системы и устройства летательных аппаратов» и «Технология изготовления приборов и систем» у студентов Института космической техники Сибирского государственного аэрокосмического

университета (ИКТ СибГАУ) возникает необходимость в создании электронных обучающих ресурсов. Такие ресурсы состоят из информационной (содержательной) и программно-реализуемой частей. Студенты СибГАУ берут на себя задачу реализации содержательной части ЭОР и выступают в роли заказчиков и авторов технического задания для студентов-математиков, которые становятся разработчиками мультимедийных демонстрационных программ. Программы представляют собой учебные симуляторы некоторых технологических процессов. Участие в коллективной разработке ЭОР требует от студентов самостоятельного освоения современных программных продуктов, анализа их функциональных возможностей, надежности, эффективности и других критериев качества, принятия решения о целесообразности использования конкретного программного продукта для достижения поставленной цели. Студенты приобретают действительный опыт разработки информационного продукта, востребованного в реальном учебном процессе. Таким образом, уже на первом курсе студенты вовлекаются в научно-исследовательскую проектную деятельность и имеют возможность увидеть в перспективе будущие профессиональные обязанности, и, следовательно, проектировать развитие личностного потенциала в профессиональной сфере. Более подробно о результатах данной работы сообщается в §2.3.

Наконец, третье возможное направление проектной работы студентов – организация некоммерческого сотрудничества с IT-фирмами. Отметим, что такая деятельность соответствует тем задачам, которые государство ставит в качестве мер поддержки отрасли информационных технологий: в Дорожной карте «Развитие отрасли информационных технологий» разработаны механизмы для «повышения активности сотрудничества компаний отрасли информационных технологий и образовательных организаций высшего образования» [115]. Несмотря на то, что студенты, принимая участие в разработке и реализации программных проектов, не будут зарабатывать деньги (по крайней мере, на начальных этапах работы), они получают от такой работы несомненную выгоду: знакомство с различными аспектами будущей профессиональной деятельности, с

трендами в развитии аппаратных и программных продуктов IT-рынка, опыт прикладного практического программирования. В настоящее время эта проектная работа находится на стартовом этапе и развивается в направлении участия студентов в тестировании и создании облачных приложений.

2.3. Планирование, проведение и результаты педагогического эксперимента

Проведение педагогического эксперимента имело следующие цели:

- 1) подтверждение выдвинутой гипотезы о возможности достигнуть планируемые результаты обучения программированию студентов-математиков, обновив компоненты методической системы на основе проективно-рекурсивной стратегии и применения когнитивных технологий;
- 2) внедрение предлагаемой методики обучения программированию в реальный педагогический процесс подготовки бакалавров-математиков;
- 3) оценка эффективности методики путем сравнения результатов обучения программированию в экспериментальной и контрольной группах.

Экспериментальной базой исследования был Институт математики и фундаментальной информатики Сибирского федерального университета. В экспериментальную группу входили первокурсники направлений подготовки 01.03.01 «Математика», профили подготовки «Вещественный, комплексный и функциональный анализ», «Алгебра, теория чисел, математическая логика» и 01.03.02 «Прикладная математика и информатика», профили подготовки «Математическое моделирование и вычислительная математика», «Системный анализ, исследование операций и управление». Контрольную группу составили первокурсники направления подготовки 01.03.02 «Прикладная математика и информатика» по вышеуказанным профилям. Общее количество участников экспериментальной группы за три учебных года составило 124 человека. На первом курсе по двум указанным направлениям подготовки в учебной программе, соответствующей ФГОС ВПО 3 поколения, предусмотрено освоение следующих дисциплин: «Основы информатики» (форма контроля – зачет), «Языки и методы

программирования» (форма контроля – экзамен), «Информатика и программирование» (форма контроля – зачет и экзамен).

Концептуально-констатирующая стадия эксперимента проходила с 2011 г. по 2012 г. Выбор темы исследования был обусловлен многолетней практикой автора в качестве преподавателя программирования, желанием осмыслить и обобщить накопленный опыт. На данном этапе потребовалось изучение и анализ психолого-педагогической и научно-методической литературы по теме исследования с целью выбора методологических основ исследования, изучение, сравнение и обобщение существующих методик обучения программированию в отечественной и зарубежной практике. Анализ проблемной области и проведение констатирующего эксперимента позволили выявить противоречия в разных аспектах педагогического процесса, связанного с подготовкой бакалавров-математиков и обучением программированию, сформулировать основную проблему, цель и гипотезу исследования. Были поставлены задачи и определена их логическая последовательность.

Констатирующий эксперимент включал анкетирование студентов набора 2010 года, проведенное в конце 1 курса (приложение 6). Анкеты заполнили 61 человек, причем среди них были студенты направления подготовки «Математика и компьютерные науки», изучающие язык программирования Object Pascal. Анкетирование имело целью выявить некоторые характеристики контингента поступающих в Институт математики и фундаментальной информатики, их мотивацию к обучению и самооценку. Ряд вопросов был задан с целью дальнейшей разработки эффективной методики обучения программированию и планирования содержательного компонента методической системы. Анализ анкет представлен в таблице 4.

Таблица 4

Результаты анкетирования студентов 2010-2011 гг. (показатели до начала обучения)

Вопрос анкеты		Количество студентов	Процент от количества опрошенных
1. Оценка в аттестате по математике (алгебре):			
отлично		28	46%
хорошо		26	43%
удовлетворительно		7	11%
2. Оценка в аттестате по информатике:			
отлично		42	69%
хорошо		18	30%
удовлетворительно		1	2%
3. Изучали ли Вы какой-либо язык программирования до поступления в ВУЗ?			
На уроках информатики в школе		35	57%
Из них изучали дополнительно:			
На дополнительных занятиях по информатике в школе		6	
В другом месте помимо школы		7	
Самостоятельно		6	
Изучали в школе и дополнительно		19	31%
Только на дополнительных занятиях по информатике в школе		1	2%
Только в другом месте помимо школы		4	7%
Только самостоятельно		2	3%
Не изучал		19	31%
4. Умели ли Вы использовать блок-схемы или другие способы записи алгоритмов до поступления в ВУЗ?	Да	44	72%
	Нет	17	28%

Результаты анкетирования показали, что, несмотря на достаточно высокий процент отличных и хороших оценок в аттестате по информатике, почти треть опрошенных студентов (31%) не изучали языки программирования в школе и не умели использовать формальное описание алгоритмов. Ещё 12% студентов указали, что изучали язык программирования не на уроках информатики. Следует заметить, что опрос проводился в конце второго семестра обучения среди студентов, оставшихся после отчисления по итогам зимней сессии.

Вторая часть вопросов анкеты была посвящена выяснению самооценки студентов по усвоению курса программирования. Обобщенные результаты приведены в таблице 5.

Таблица 5

Результаты анкетирования студентов 2010-2011 гг. (показатели в конце первого года обучения)

Вопрос анкеты	Количество студентов	Процент от количества опрошенных
1. Насколько успешно Вы осваиваете курс программирования:		
Могу самостоятельно написать программу по всем пройденным темам программирования	26	43%
Могу написать программу по аналогии с разобранной программой	21	34%
Могу разобраться в готовой программе, но не пишу программы самостоятельно	11	18%
Ничего не понимаю в программировании	3	5%
2. Если Вы испытываете трудности в изучении языка программирования, то причина этого, на Ваш взгляд:		
Собственное нежелание научиться программировать	16	26%

Недостаточность аудиторных часов курса программирования	23	38%
Недостаток бумажных и электронных источников информации	16	26%
Качество преподавания	6	10%
Другие причины	12	28%

Ответы, полученные в ходе анкетирования, были сравнены с результатами минисессии, проводимой в институте в середине второго семестра в форме экзамена по программированию в компьютерном классе. Данные, полученные в ходе сравнения, визуализированы на рисунке 24.

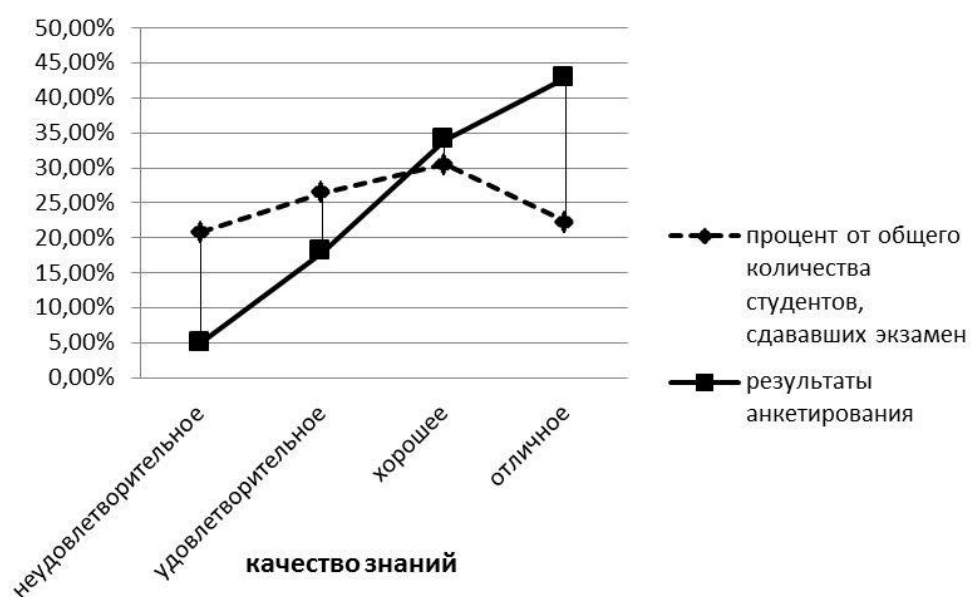


Рис 24. Показатели результатов весенней минисессии (2011 г.) и самооценки студентов, полученной при анкетировании

Представленные графики показывают значительное расхождение в оценке качества знаний самими студентами (самооценка) и преподавателями, кроме уровня, обозначенного как «хорошее качество знаний» (в анкете соответствует пункту «могу написать программу по аналогии с разобранный»). Наблюдалась завышенная самооценка у большой группы студентов: 43% опрошенных считают, что могут успешно самостоятельно писать программы, но результаты минисессии

этого не подтвердили, т.к. только 22% из сдававших экзамен продемонстрировали высокий уровень знаний и умения программировать.

Анализ пункта анкеты о наличии трудностей в обучении программированию (студенты могли выбрать несколько пунктов) показал, что всего 3% респондентов не видят сложности в теории и практике программирования. Интересно, что этот показатель оказался близким к цифре, приведенной в известной статье Д. Кнута [67], в которой он размышлял о сходстве и различии математического и алгоритмического мышления – примерно 2% студентов обладают природным алгоритмическим мышлением и могут стать квалифицированными программистами. Наибольший процент (38%) собрал пункт «недостаточность аудиторных часов», 26% опрошенных указали, что испытывают недостаток в качественных источниках информации. Исходя из анализа представленных данных, был сделан вывод о необходимости электронного обучающего курса по программированию на языке C/C++.

Начиная с 2012 г. анкетирование стало проводиться в самом начале обучения на 1 курсе. Предлагаемая студентам анкета представлена в приложении 7. Кроме этого, с 2012 по 2014 гг. проводится входное тестирование. Основные результаты тестирования приводились в §1.1.

Поисково-формирующая стадия эксперимента проходила в 2012-2014 гг. Этот этап включал продолжение теоретико-методологических исследований, уточнение и корректировку основных теоретических положений методики, проектирование электронного обучающего курса, проведение формирующего эксперимента.

На *заключительной стадии эксперимента* (2014-2015 гг.) был проведен анализ и обобщение результатов педагогического эксперимента, подведены итоги формирующего эксперимента для оценки эффективности методики, проверена обоснованность основных положений исследования, результаты эксперимента соотнесены с выдвинутой гипотезой. Далее происходило оформление диссертации.

Основная часть экспериментальной работы (формирующий эксперимент) была проведена в 2012-2014 годах и продолжается в настоящее время. Апробация в учебном процессе теоретических положений, сформулированных в первой главе, происходила поэтапно.

Обучение программированию экспериментальной группы набора 2012 года было реализовано с позиций проективно-рекурсивной стратегии. В этой группе, наряду со значительным количеством слабых студентов (36,4% – по результатам входного тестирования), был высок процент студентов с хорошим уровнем начальной подготовки (42,4%). Сильные студенты успешно включились в научно-исследовательскую работу по проектам межвузовской кооперации. Реализация проектов потребовала от студентов самостоятельного изучения объектно-ориентированного программирования на языке C++, сценарных (скриптовых) языков JavaScript, SVG, освоения графической технологии OpenGL (которая изучается в дисциплине 2 курса «Компьютерная графика» для студентов направления подготовки 01.03.02 «Прикладная математика и информатика»). Часть студентов предпочла самостоятельную работу над проектом, другая часть объединилась в группу. Результатом работы были программные продукты, реализующие различные визуализированные схемы ракетных двигателей, с возможностью пошагового и интерактивного взаимодействия пользователя со схемами. Пример программного продукта – интерактивной анимации (один из кадров) представлен на рисунке 25. Созданная интерактивная схема моделирует пневмогидравлическую систему заправки топливом космического летательного аппарата на орбите.

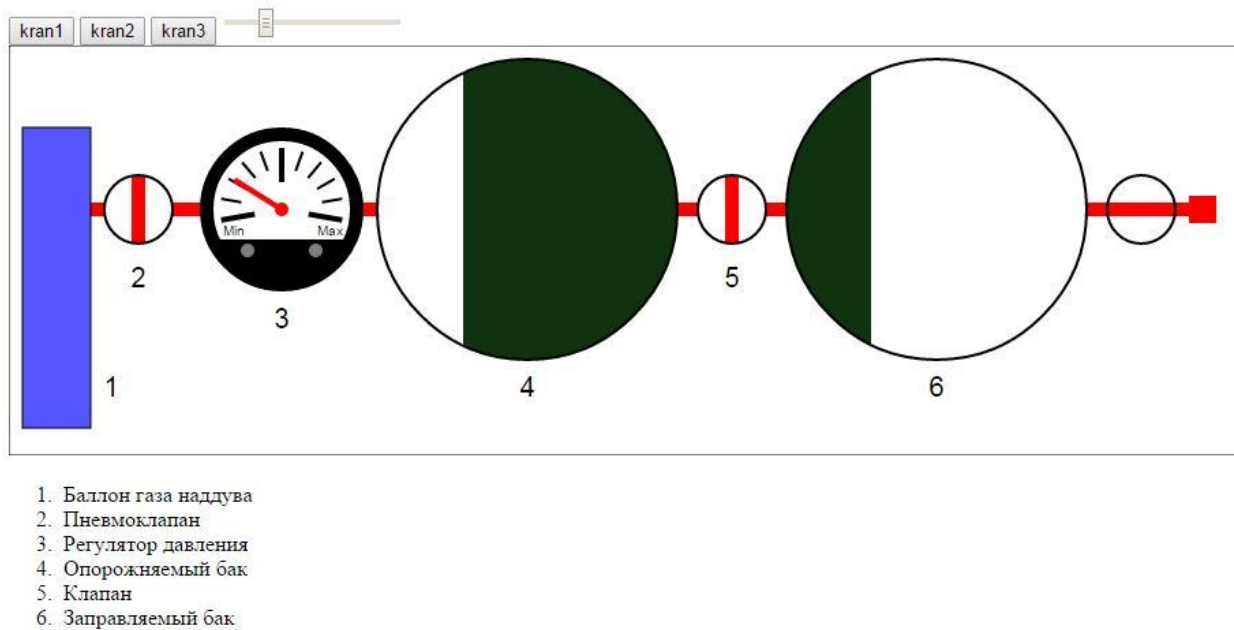


Рис. 25. Интерактивная схема системы заправки топливом космического летательного аппарата

Разработанные программные продукты были переданы студенческо-преподавательской группе Сибирского государственного аэрокосмического университета для использования в качестве электронных образовательных ресурсов. Кроме этого, студенты ИМиФИ совместно со студентами СибГАУ принимали участие во Всероссийском конкурсе компьютерной графики и анимации, Всероссийских научно-практических конференциях творческой молодежи (2013г. и 2014 г.), опубликовали результаты научных исследований [39], [46], [75].

Главные методические задачи, решаемые в процессе обучения программированию слабых студентов, были следующие:

а) студенты должны достигнуть базового уровня по 3-му показателю результатов обучения программированию (способность создавать программный продукт);

б) привлечь студентов к наполнению контента электронного курса путем создания учебных ресурсов и контрольно-измерительных материалов (тесты);

в) существенно повысить уровень алгоритмического и системного мышления студентов;

г) студенты не должны потерять заинтересованности и желания освоить практическое программирование при первых неудачах.

Результаты экспериментальной и контрольной группы представлены на рисунке 26.

Как видно из гистограммы, в экспериментальной группе достигнуты значительно лучшие результаты по сравнению с контрольной группой: низкие результаты обучения показали 3% испытуемых экспериментальной группы против 25% студентов контрольной группы.

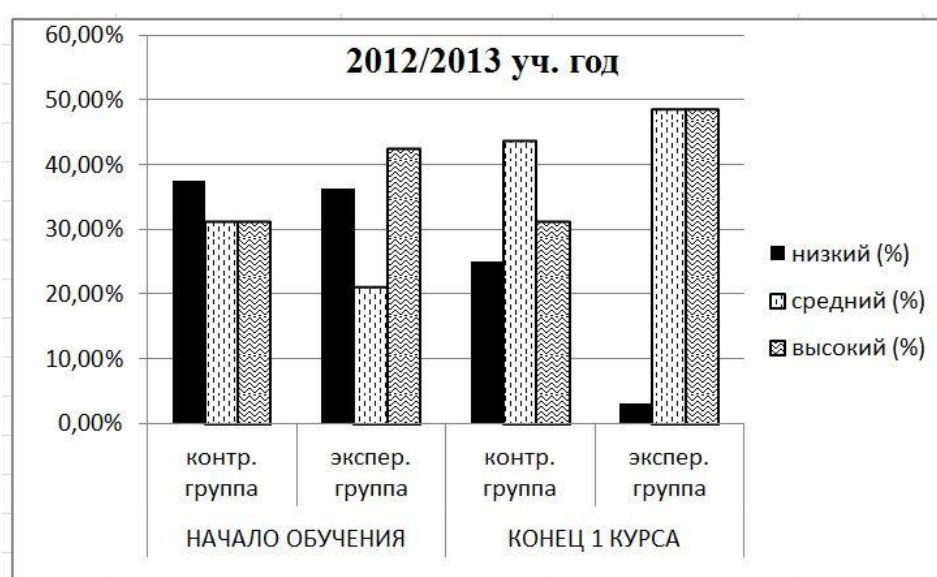


Рис.26. Результаты обучения программированию экспериментальной и контрольной групп в 2012-2013 гг.

Набор первокурсников 2013 года оказался существенно слабее предыдущего года: 75% студентов экспериментальной группы показали неудовлетворительные результаты во входном тестировании. Был проведен анализ рейтинга (суммарное количество баллов ЕГЭ) абитуриентов, поступивших на 1 курс ИМиФИ. Результаты отображены в таблице 6.

Усредненный рейтинг первокурсников 2013 г. по группам и направлениям
подготовки

группа	направление подготовки, профиль	Средний суммарный балл
ИМ13–01 (контрольная группа)	010400.62 Прикладная математика и информатика, профиль «Математическое моделирование и вычислительная математика»	228,62
ИМ13–02 (контрольная группа)	010400.62 Прикладная математика и информатика, профиль «Системный анализ, исследование операций и управление»	174, 23
ИМ13–03 (экспериментальная группа)	010400.62 Прикладная математика и информатика, профиль «Системный анализ, исследование операций и управление»	157,1
ИМ13–04 (экспериментальная группа)	010100.62 Математика, профиль «Алгебра, теория чисел, математическая логика	200,77

Таким образом, средний рейтинг экспериментальной группы составил 178, 94 балла, контрольной группы – 201, 43.

Возникла необходимость в корректировке методики обучения программированию: добавлено использование когнитивных средств обучения, к которым также была применена проективно-рекурсивная стратегия. Когнитивные средства обучения должны были повысить доступность и понимание учебного материала, для большинства обучающихся нового и сложного. Эксперимент показал, что, несмотря на преимущество в результатах высокого уровня в контрольной группе, (что объясняется лучшими стартовыми показателями и способностями студентов в контрольной группе в соответствии с результатами из

таблицы 4), в экспериментальной группе наблюдается лучшая динамика в сокращении количества студентов, не достигших базового уровня (рисунок 27).

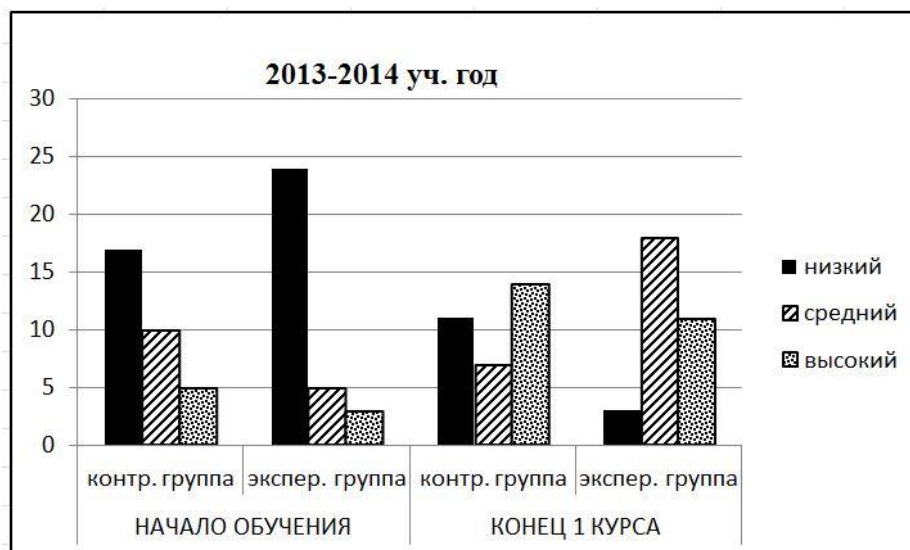


Рис.27. Результаты обучения программированию экспериментальной и контрольной групп в 2013-2014 гг.

Эксперимент продолжается в 2014-2015 гг. Можно подвести предварительные итоги после окончания 1-го семестра. Поскольку формой аттестации студентов по программированию в 1-м семестре является зачет (т.е. возможные результаты обучения – «зачтено» и «не зачтено»), более информативным представляется сравнение баллов, полученных студентами экспериментальной и контрольной групп в ходе промежуточной аттестации – минисессии. Результаты такого сравнения отображены на рисунке 28.

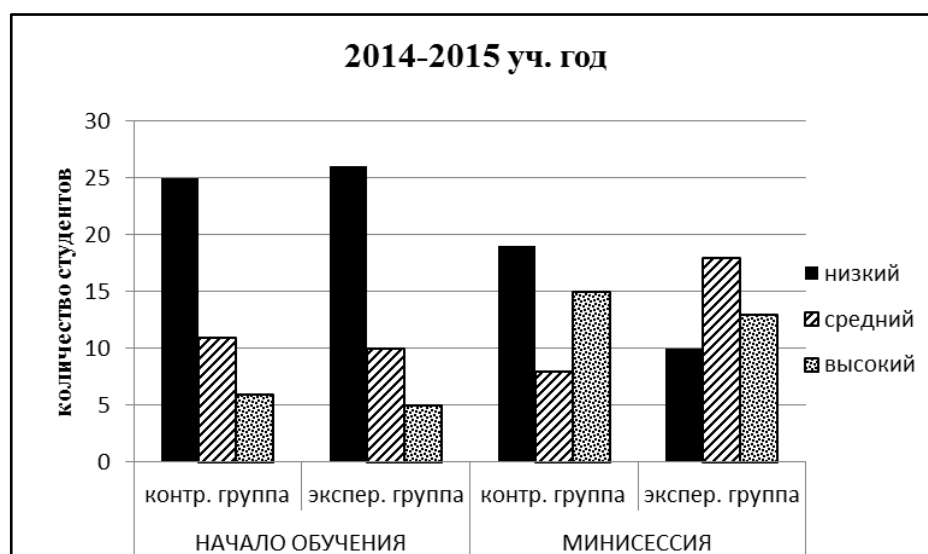


Рис.28. Результаты обучения экспериментальной и контрольной групп набора 2014-2015 гг. после 1 семестра

На гистограмме сохраняется ранее отмеченная тенденция: количество студентов с низкими результатами обучения существенно меньше в экспериментальной группе. Все вышеприведенные результаты были исследованы на достоверность с помощью статистического критерия χ^2 . После проведенных расчетов были получены значения:

2012 – 2013 учебный год: $\chi^2_{\text{эмп}} = 6,94$;

2013 – 2014 учебный год: $\chi^2_{\text{эмп}} = 9,78$;

2014 – 2015 учебный год: $\chi^2_{\text{эмп}} = 6,9$.

Все полученные эмпирические значения превышают $\chi^2_{\text{крит}} = 5,99$ при уровне статистической значимости 0,05, что свидетельствует о достоверности различий в результатах контрольной и экспериментальной групп в начале и конце эксперимента с вероятностью 95%.

Анализ проектной деятельности студентов во время учебной практики дал следующие результаты (таблица 7).

Таблица 7

Сравнительный анализ проектных задач учебной практики

тип задачи		Количество выполнивших студентов (в %) по отношению к общему числу	Самостоятельная постановка задачи
Разработка образовательных ресурсов для ЭОК		26,7%	да
Реализация игровых задач, из них	поставлена преподавателем	20%	нет
	выбрана студентом	13,3%	да
Реализация математических задач, из них	поставлена преподавателем	23,3%	нет
	выбрана студентом	3,3%	да

Реализация задач программирования, из них	поставлена преподавателем	10%	нет
	выбрана студентом	3,3%	да

Результаты, приведенные в таблице, свидетельствуют об актуализации принципа проективности, заинтересованности и самостоятельности студентов: 66,7% студентов предпочли задачи, связанные с будущей профессиональной деятельностью, почти половина студентов (46,6%) выбрали (поставили) задачу самостоятельно, исходя из своих интересов и возможностей.

В экспериментальной группе набора 2014-2015 г. были проведены исследования эффективности активного использования когнитивных технологий в обучении программированию. Под активным использованием будем понимать создание ментальных и концептуальных карт самими обучаемыми, пассивным использованием назовем изучение готовых карт в процессе обучения. При подготовке к компьютерному тестированию часть обучаемых выполнила задание на составление ментальной карты по изучаемой теме (активное использование), часть – нет. Набранные баллы в обеих группах отражены в виде лепестковой диаграммы на рисунке 29.

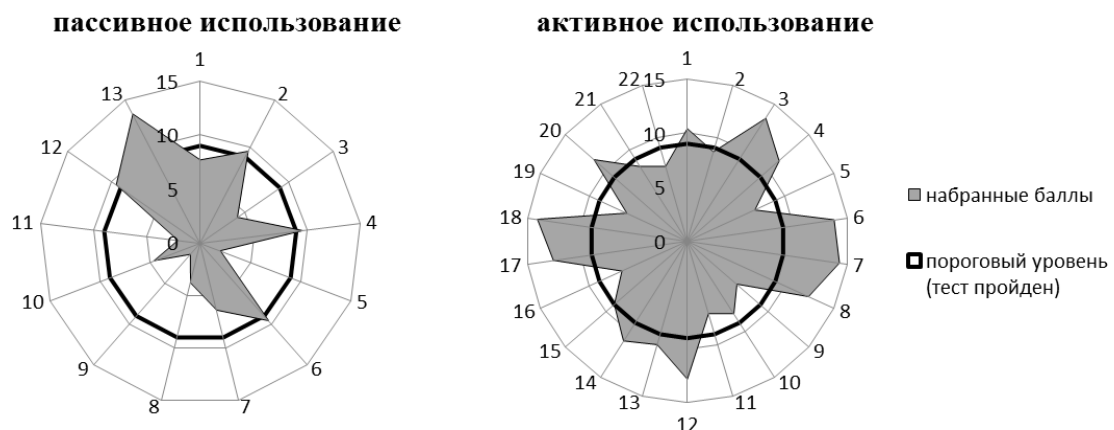


Рис.29. Результаты теста (в баллах) при активном и пассивном использовании ментальных карт

Подчеркнем, что на диаграмме не учитывались результаты студентов с высоким входным уровнем, т.к. прохождение теста для них не представляло сложности. Количество испытуемых, прошедших тест (набравших более половины баллов) больше в группе активного использования ментальных карт (59% против 38%), к тому же в этой группе меньшее количество очень низких результатов. Приведенный пример демонстрирует преимущество активного использования когнитивных средств обучения перед пассивным.

Что касается оценивания качества концептуальных карт, которые создают студенты, и на основании этого, оценивания уровня развития системного мышления, можно резюмировать, что, недостаточное время обучения (один семестр) пока не позволили студентам выйти на высокий уровень. В качестве зачетного задания студентам предлагалось составить концептуальные карты по учебному материалу первого семестра (см. рисунок 13). Расширительно-абстрактный уровень системного мышления (высший) для этого задания заключался в том, что студенты должны были обобщить понятие «тип данных» для понятия «массив», провести категоризацию изученного материала в виде подсистем «данные, структуры данных», «команды, управление», «алгоритмы». Только 6% представленных концептуальных карт удовлетворяли данным критериям. Подавляющее большинство студентов продемонстрировали многоструктурный уровень системного мышления, после доработки концептуальных карт вышли на реляционный уровень. Очевидно, что материал 2-го семестра (изучение типа данных «структуры», основы объектно-ориентированного программирования) должен в значительной мере способствовать повышению уровня системного мышления.

Уровень развития алгоритмического мышления оценивался посредством компьютерного тестирования. Сравнить полученные результаты тестирования за три учебных года можно с помощью рисунка 30. Баллы, полученные испытуемыми, были распределены на 5 групп и выражены в процентном соотношении. Заметно преобладание хороших результатов (выполнено от 60% до 80% предложенных заданий тестов). Сочетание проективно-рекурсивной

стратегии и когнитивных технологий дало более высокие результаты в 2013-2014 учебном году по сравнению с 2012-2013 г., хотя экспериментальная группа 2013-2014 г. была объективно слабее предыдущей. В целом за все годы экспериментального исследования наблюдался существенный рост уровня развития алгоритмического мышления по сравнению с началом обучения.

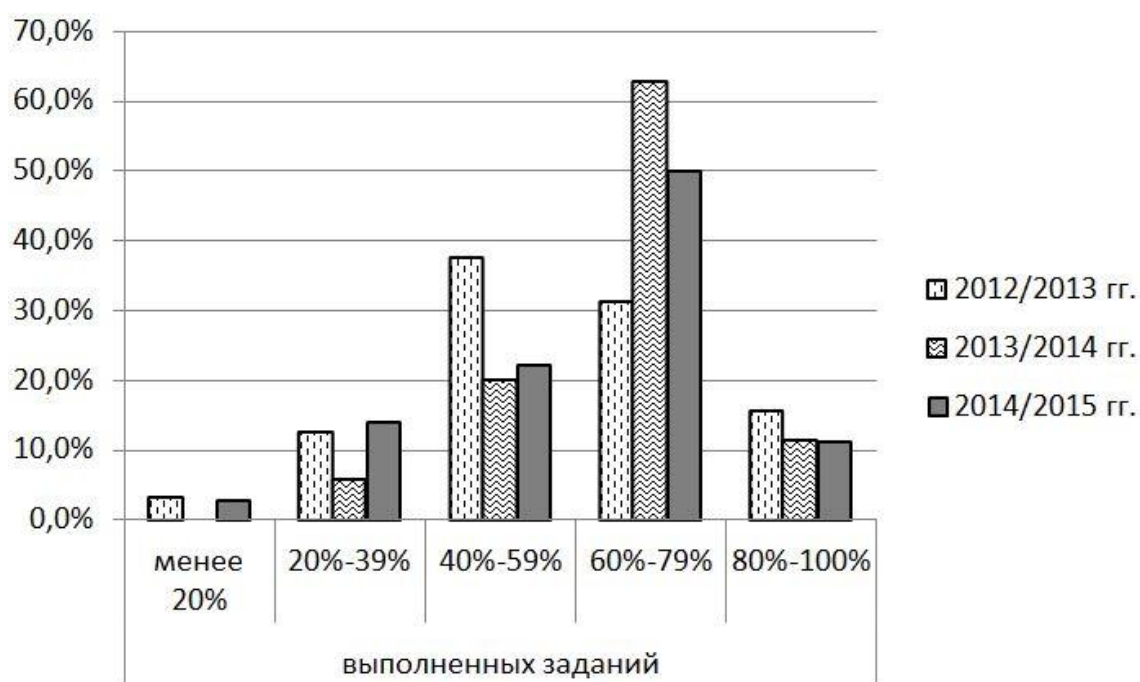


Рис. 30. Динамика изменения результатов компьютерного тестирования

ВЫВОДЫ ПО ГЛАВЕ 2

Подводя итоги разработки структурно-логической и процессуальной моделей проективно-рекурсивного обучения программированию студентов математических направлений подготовки, выясним соответствие указанных моделей принципам обновления компонентов методической системы на основе проективно-рекурсивной стратегии, которые были сформулированы в §1.2:

– *принцип открытости* реализуется в возможности свободного доступа к контенту электронного курса, вариативности и гибкости содержательного и

технологического компонентов методической системы обучения программированию;

– *принцип динамичности* реализуется в постоянном обновлении и модернизации технологического компонента методической системы, разработке новых проектов и привлечении новых участников к их реализации;

– *принцип рекурсивности* выражен в том, что обновленные содержательный и технологический компоненты методической системы имеют рекурсивную природу: это комплексы самостоятельных проектов, в том числе и индивидуальных проектов участников;

– *принцип модульности* актуализирован в модульной организации контента электронного курса, что даёт возможность студентам самостоятельно выбирать индивидуальную образовательную траекторию в рамках изучаемого курса программирования;

– *принцип эволюционности* реализован в организации содержательного компонента по правилу дидактической спирали;

– *принцип адаптивности* выражен в настроенности всех компонентов обновленной методической системы на потребности и возможности студентов как полноправных субъектов образовательного процесса.

Разработанная на основе структурно-логической и процессуальной моделей методика обучения программированию студентов-математиков удовлетворяет требованиям, предъявляемым к методике обучения программированию (§ 1.1):

1) на всех этапах разработки используется системный подход;

2) методика разработана с целью повышения качества обучения программированию будущих бакалавров математических направлений подготовки, что обусловлено социальным заказом общества и профессиональными требованиями работодателей;

3) учитывается компетентностный подход к оцениванию результатов обучения: результаты обучения программированию непосредственно влияют на формирование профессиональных и общекультурных компетенций будущих бакалавров-математиков;

4) при организации обучения программированию используется современное программное обеспечение, широко распространённый в профессиональной среде программирования универсальный язык программирования C/C++;

5) эффективная самостоятельная работа студентов происходит с помощью электронного обучающего онлайн-курса «Программирование на C/C++»;

6) разработка ЭОК в среде Moodle обеспечивает постоянную интерактивную связь между студентами и преподавателем, студентов друг с другом;

7) представлены разнообразные образовательные ресурсы в виде визуальной и текстовой информации, формы контроля также учитывают различные учебные стили обучающихся;

8) использование когнитивных средств обучения способствует пониманию обучающимися сложного учебного материала;

9) выполнение студентами самостоятельных лабораторных работ и проектов учебной практики обеспечивает практико-ориентированный характер обучения;

10) активное использование метода проектов включает обучающихся в научно-исследовательскую и профессиональную деятельность;

11) проективно-рекурсивная стратегия обучения программированию способствует повышению мотивации и ответственности студентов в процессе учебной и научно-исследовательской деятельности.

Таким образом, с учетом имеющегося соответствия перечисленных принципов и разработанных моделей, удовлетворения требованиям, предъявляемым к современной методике и результатов проведения педагогического эксперимента, можно сформулировать следующие утверждения:

1. Созданные на основе проективно-рекурсивной стратегии и когнитивного подхода структурно-логическая и процессуальная модели обучения программированию студентов математических направлений подготовки позволяют эффективно реализовать теоретические и практические аспекты разработки методики обучения программированию в соответствии с поставленными целями.

2. Гипотеза исследования о возможности достижения планируемых результатов обучения программированию студентов математических направлений подготовки получила эмпирическое подтверждение в процессе проведения педагогического эксперимента.

ЗАКЛЮЧЕНИЕ

Таким образом, обобщая результаты проведенного диссертационного исследования, можно сделать следующие выводы:

1. Анализ требований, предъявляемых к современной эффективной методике обучения программированию бакалавров-математиков в условиях реализации ФГОС ВПО, и возможности повышения качества математического образования за счет профессиональной подготовки, в том числе, по программированию, позволил сделать вывод о несоответствии существующих методик обучения программированию сформулированным требованиям и необходимости обновления компонентов методической системы обучения студентов программированию.

2. Компоненты методической системы целесообразно модернизировать с позиций проективно-рекурсивной стратегии следующим образом:

- в целевом и результативно-оценочном компонентах акцент переносится на достижение планируемых результатов обучения программированию на трех уровнях: дисциплинарном, профессиональном и метадисциплинарном;
- содержательный компонент состоит из базовой и вариативной частей и представлен системой многоуровневых профессиональных задач; на протяжении всего курса поддерживается сквозная содержательная линия «тип данных»;
- технологический компонент содержит традиционные и когнитивные средства и методы.

3. Проективно-рекурсивная стратегия обновления целевого, содержательного, технологического и результативно-оценочного компонентов методической системы обучения программированию студентов-математиков обеспечивает возможность использовать базовый курс программирования как средство достижения результатов обучения, которые определены как три взаимосвязанных компонента:

- уровень развития алгоритмического мышления;

- уровень развития системного мышления;
- способность создавать программный продукт на изучаемом языке программирования.

4. Уточнение понятий «алгоритмическое мышление», «системное мышление», выделение определённых индикаторов в каждом компоненте результатов обучения и выбор соответствующих критериев оценивания позволяют предложить модель оценки результатов обучения студентов-математиков программированию с позиций компетентностного и когнитивного подходов.

5. Эффективными средствами достижения требуемых результатов обучения и удобными инструментами их оценивания в показателях уровня развития алгоритмического и системного мышления обучающихся являются когнитивные методы и средства обучения в виде концептуальных и ментальных карт. Принципы проективности и рекурсивности, положенные в основу разработки и использования студентами понятийных и алгоритмических ментальных схем по курсу программирования, способствуют:

- индивидуализации процесса обучения;
- повышению мотивации студентов к обучению;
- ответственности обучающихся за собственные результаты обучения и продукты учебной деятельности;
- профессионально-ориентированному обучению.

6. В соответствии с планируемыми результатами обучения обоснованы и разработаны: структурно-логическая модель (система, состоящая из целевого, содержательного, технологического и результативно-оценочного компонентов); процессуальная модель (субъект-объектная динамическая схема) обучения программированию бакалавров-математиков, опирающиеся на проективно-рекурсивный и когнитивный подходы в обучении. Разработанные модели обеспечивают организационно-методические условия достижения запланированных результатов обучения: высоких уровней развития алгоритмического и системного мышления, способности самостоятельно создавать программный продукт на изучаемом языке программирования.

7. Спроектированный и апробированный в реальном учебном процессе электронный обучающий курс «Программирование на C/C++» для первокурсников математических направлений подготовки позволяет обеспечить условия для реализации дидактических возможностей смешанного обучения с использованием среды Moodle.

8. Разработанная на основе структурно-логической и процессуальной моделей методика проективно-рекурсивного обучения студентов-математиков программированию с использованием традиционных и когнитивных методов и средств обучения, диагностики и контроля результатов обучения с опорой на метод проектов (представленный реализацией проектов учебной практики, межвузовской кооперации и IT-бизнеса) и метод ментальных карт, обеспечивает достижение проектируемых результатов обучения, удовлетворяющих требованиям ФГОС ВПО.

9. Возможность достижения планируемых результатов обучения на основе разработанной методики проективно-рекурсивного обучения студентов-математиков программированию экспериментально подтверждена в ходе педагогического эксперимента на базе Института математики и фундаментальной информатики Сибирского федерального университета. В этой связи, методику проективно-рекурсивного обучения программированию можно рекомендовать к использованию в учебной практике.

Таким образом, цель диссертационного исследования достигнута, полученные результаты доказывают положения, выносимые на защиту, и имеют теоретическую и практическую значимость.

Продолжение исследования и практической работы планируется по следующим направлениям:

1) Дальнейшее наполнение и оптимизация контента электронного курса. Целесообразно разделение ресурсов для обучения процедурному и объектно-ориентированному программированию, чтобы избежать излишней перегруженности курса. Планируется совершенствование фонда контрольно-

измерительных материалов на основе когнитивного и компетентностного подходов.

2) Расширение участия студентов в проектах межвузовской кооперации и IT-бизнеса с целью их профессиональной ориентации, повышения мотивации к обучению программированию.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Альтшуллер, Г. С. Найти идею. Введение в теорию решения изобретательских задач / Г.С. Альтшуллер. – Новосибирск: Наука. Сиб. отд., 1991. – 175 с.
2. Арнхейм, Р. Визуальное мышление / Р. Арнхейм // Зрительные образы: феноменология и эксперимент. вып. 1,3. – Душанбе, 1973.
3. Асмолов, А.Г. Принципы организации памяти человека: системно-деятельностный подход к изучению познавательных процессов / А.Г. Асмолов. – М.: изд-во Моск. ун-та, 1985. – 104 с.
4. Ахметова, Л.В. Методы когнитивного обучения: психолого-дидактический подход / Л.В. Ахметова // Вестник ТГПУ. – 2009. – №7(85). – с. 48–52.
5. Бабанский, Ю.К. О дидактических основах повышения эффективности обучения / Ю.К. Бабанский // Народное образование. – 1986. – № 11. – с.105–111.
6. Баженова, И.В. Формирование базовых понятий программирования у студентов в контексте информационного подхода / И.В. Баженова // Вестник КГПУ им. В.П. Астафьева. – 2012. – № 2. – с. 21–25.
7. Баженова, И.В. Система заданий по теме «Типы данных» при обучении объектному программированию с позиций информационного подхода / И.В. Баженова // Информационные ресурсы в образовании: материалы Международной научно-практической конференции. – Нижневартовск: НГГУ, 2012. – с. 58–60.
8. Баженова, И.В. Использование проективной стратегии в обучении программированию студентов-математиков / И.В. Баженова // Интегрированная система профессионального образования: проблемы и пути развития: материалы Всероссийской научно-методической конференции. – Красноярск: СибГАУ, 2012. – с. 247–249.
9. Баженова, И.В. Проектирование содержания и результатов обучения программированию студентов-математиков / И.В. Баженова // Вестник Красноярского государственного педагогического университета им. В.П. Астафьева. – 2013. – № 2(24) – с.79–82.

10. Баженова, И.В. Визуализация знания как метод когнитивного подхода к обучению программированию / И.В. Баженова // Практико-ориентированное обучение в профессиональном образовании: проблемы и пути развития: материалы научно-практической конференции в рамках XVIII Междунар. научной конференции «Решетневские чтения». – Красноярск: СибГАУ, 2014. – с. 281–285.
11. Баженова, И.В. К вопросу отбора содержания курса «Информатика и программирование» для студентов математических направлений подготовки на основе проективно-рекурсивной стратегии / И.В. Баженова // Фундаментальная информатика, информационные технологии и системы управления: реалии и перспективы. ФПТМ-2014: материалы международной научно-практической конференции. – Красноярск: СФУ, 2014. – с. 18–23.
12. Баженова, И.В. Проблемы перехода от информатики в школе к изучению курса программирования в вузе / И.В. Баженова, С.Г. Толкач // Открытое образование: опыт, проблемы, перспективы: материалы VI Всероссийской научно-практической конференции с международным участием. – Красноярск: КГПУ им. Астафьева, 2011. – с.20–24.
13. Баженова, И.В. Информатика и программирование (учебно-методическое пособие) / И.В. Баженова, И.В. Баранова, С.Н. Баранов, С.Г. Толкач // Красноярск: СФУ, 2012. – 112 с.
14. Баженова, И.В. Социальное партнерство как один из инструментов управления инновациями в образовании / И.В. Баженова, Г.М. Гринберг, Л.М. Ивкина // Современные материалы, техника и технология: материалы 2-й Международной научно-практической конференции. – Курск: Юго-Зап. гос. университет, 2012. – с. 59–65.
15. Баженова И.В. Межвузовская кооперация субъектов образования как один из факторов повышения качества их профессиональной подготовки / И.В. Баженова, Г.М. Гринберг, Л.М. Ивкина // Информатика и информационные технологии: материалы Всероссийской научной конференции с международным участием – Челябинск: ЧГПУ, 2013. – с. 157–162.

16. Баженова, И.В. Использование методики ментальных карт при обучении программированию в высшей школе / И.В. Баженова, Т.А. Степанова // Человек, семья и общество: история и перспективы развития: материалы II Международного научно-образовательного форума. – Красноярск: КГПУ им. В.П. Астафьева, 2013. – с. 173–176.
17. Баженова, И.В. Формирование ключевых компетенций у студентов разнопрофильных вузов на основе межвузовской кооперации / И.В. Баженова, Г.М. Гринберг, Л.М. Ивкина // Решетневские чтения: материалы XVII Международной научной конференции. – Красноярск: СибГАУ, 2013. – с. 462–463.
18. Баженова, И.В. Развитие компетенций будущих педагогов и инженеров в условиях межвузовской кооперации / И.В. Баженова, Г.М. Гринберг, Л.М. Ивкина // Вестник Челябинского государственного педагогического университета. – 2014. – № 2 – с. 62-69.
19. Баженова, И.В. Особенности методики обучения программированию студентов-математиков на основе проективно-рекурсивной стратегии и когнитивных технологий / И.В. Баженова // Педагогическое образование в России. – 2015. – № 3.
20. Байденко, В. И. Выявление состава компетенций выпускников вузов как необходимый этап проектирования ГОС ВПО нового поколения: методическое пособие / В.И. Байденко – М.: Исследовательский центр проблем качества подготовки специалистов, 2006. – 72 с.
21. Баранов, Ю.С. Методика профессионально ориентированной предметной подготовки будущих учителей информатики на основе проективно-рекурсивного обучения: автореф. дис. ... канд пед. наук : 13.00.02 / Баранов Сергей Николаевич. – Красноярск, – 2008. – 26 с.
22. Бартлет, Ф.Ч. Повседневное мышление / Ф.Ч. Бартлет // Хрестоматия по психологии. Психология мышления. Под ред. Ю.Б.Гиппенрейтер, В.А. Спиридонова, М.В. Фаликман, В.В.Петухова – М., АСТ: Астрель, 2008. – с.594–618.

23. Безрукова, В.С. Педагогика. Проективная педагогика: учеб. пособие для инженер.-пед. ин-тов и индустр.-пед. техникумов / В.С. Безрукова. – Екатеринбург: Деловая книга, 1999. – 344 с.
24. Бершадский, М.Е. Теоретико-практические аспекты работы с картами интеллект-понятий / М.Е. Бершадский // Народное образование. – 2012. – №6. – с. 203–212.
25. Бершадский, М.Е. Классификации технологий учебного процесса / М.Е. Бершадский // Профильная школа. – 2014. – №1. – с. 22–33.
26. Бобков, В. В. Реализация дифференцированного подхода в обучении студентов программированию с помощью дидактического технологического комплекса: автореф. дис. канд. пед. наук: 13.00.02 / В. В. Бобков – Красноярск, 2006. – 24 с.
27. Болонский процесс: Результаты обучения и компетентностный подход (книга-приложение 1) / Под науч. ред. В.И. Байденко. – М.: Исследовательский центр проблем качества подготовки специалистов, 2009. – 536 с.
28. Богомаз, И.В. Научно-методические основы базовой подготовки студентов инженерно-строительных специальностей в условиях проективно-информационного подхода: автореф. дис. ... д-ра пед. наук : 13.00.02 / Богомаз Ирина Владимировна. – Москва, – 2012. – 37 с.
29. Брунер, Дж. Психология познания: за пределами непосредственной информации / Дж. Брунер. – М.: Прогресс, 1977. – 412 с.
30. Буторин, Д.Н. Практическое внедрение проективной обучающей системы в педагогическую практику // Информатика и образование. – 2008.– № 6. – с.100–103.
31. Буторин, Д.Н. Открытая среда openSEE как инструмент решения задач непрерывного образования // Инновации в открытом образовании. – 2012.–№ 4.
32. Бьюзен, Т. Научите себя думать : пер. с англ. 2-е изд. / Т. Бьюзен. – Минск: Попурри, 2004. – 192 с.
33. Веккер, Л. М. Восприятие и основы его моделирования / Л.М. Веккер. – Л.: Изд-во Ленинградского университета, 1964. – 194 с.

34. Величковский, Б.М. Когнитивная наука: Основы психологии познания: в 2 т. – т.1 / Б. М. Величковский. – М.: Смысл: Издательский центр «Академия», 2006. – 448 с.
35. Вербицкий, А. А. Активное обучение в высшей школе: контекстный подход : метод. пособие / А. А. Вербицкий. – М.: Высш. шк., 1991. – 207 с.
36. Вербицкий, А.А. Контекстно-компетентностный подход к модернизации образования / А.А. Вербицкий // Высшее образование в России. – 2010. – № 5. – с. 32–37.
37. Виденин С.А. Изучение вузовского курса «история информатики» в контексте обучающей технологии «обучение через делание» / С.А. Виденин // Вестник КГПУ им. В.П. Астафьева. – 2008. – № 1. – с. 45–48.
38. Вирт, Н. Алгоритмы и структуры данных /Н. Вирт – М.: Мир, 1989.–360 с.
- Внукова, Л. А. Дифференциация обучения информатике на основе использования элементов дистанционной формы обучения в классно-урочной системе : дис. ... канд. пед. : 13.00.02 / Внукова Людмила Анатольевна. – Омск, 2003. – 198 с.
39. Войтов И.Ю. Визуализация процессов, протекающих в пневмогидравлической схеме ракетного двигателя / И.Ю. Войтов, А.А. Севрунов // Актуальные проблемы авиации и космонавтики: материалы IX Всероссийской научно-практической конференции творческой молодежи. – Красноярск: СибГАУ, 2013. – с. 266-267.
40. Выготский Л.С. Проблема обучения и умственного развития в школьном возрасте / Л.С. Выготский // Теории учения. Хрестоматия. Часть 1. Отечественные теории учения. Под ред. Н.Ф. Талызиной, И. А. Володарской. – М.: Редакционно-издательский центр «Помощь», 1996.
41. Газейкина, А. И. Обучение школьников 5-7-х классов объектно-ориентированному подходу к созданию и использованию средств информационных технологий: дис. ... канд.пед.наук: 13.00.02: / Газейкина Анна Ивановна. – Екатеринбург, 2004. – 164 с.
42. Гальперин, П. Я. Методы обучения и умственное развитие ребенка / П. Я. Гальперин – М.: Педагогика, 1985. – 46 с.

43. Гальперин, П.Я. Введение в психологию: учебное пособие для вузов / П.Я. Гальперин – М.: Книжный дом «Университет», 1999. – 332 с.
44. Гилев, А.А. Когнитивное обучение физике в системе высшего инженерного образования / А.А. Гилев // Вестник Самарского государственного технического университета. – 2006. – № 46. – с.20–29.
45. Голуб, Б.А. Основы общей дидактики : учеб. пособие для студ. педвузов / Б.А. Голуб. – М.: Туманит, изд. центр ВЛАДОС, 1999. – 96 с.
46. Гончарик, К.В. Использование инновационных образовательных технологий для повышения качества обучения / К.В. Гончарик, Р.В. Коваль// Актуальные проблемы авиации и космонавтики: материалы X Всероссийской научно-практической конференции творческой молодежи. – Красноярск: СибГАУ, 2014. – с. 368–369.
47. Гофман, И. Анализ фреймов: эссе об организации повседневного опыта / И. Гофман. Под ред. Г.С. Батыгина и Л.А. Козловой; вступ. статья Г.С. Батыгина. М.: Институт социологии РАН, – 2003.
48. Гребнева, Д.М. Обучение школьников программированию на основе семиотического подхода: дис. ... канд пед. наук : 13.00.02/ Гребнева Дарья Михайловна. – Нижний Тагил, – 2014. – 183 с.
49. Григорьев, С.А. Программирование на языке Паскаль для математиков: учебное пособие / С.А. Григорьев. –Калининград: Калининградский ун-т, 1997. – 92 с.
50. Гурье, Л.И. Проектирование педагогических систем: учеб. пособие / Л.И. Гурье. – Казань: Казан. гос. технол. ун-т, 2004. – 212с.
51. Дединский, И.А. Как хотеть учиться / И.А. Дединский. – Компьютерра online: [электронный ресурс] – Режим доступа: <http://www.ded32.ru/index/0-7>
52. Дейкстра, Э. Дисциплина программирования : пер. с англ. / Э. Дейкстра. – М.: Мир, 1978.– 276 с.
53. Диков, А.В. Обучение математике через программирование / А.В. Диков // Вестник Московского городского педагогического университета. Серия

- «Информатика и информатизация образования». – М.: МГПУ, 2004, – №1(2). – с.49–56.
54. Дорохова, Т.С. Ментальный подход как основание развития самопознания студентов на примере изучения истории социальной педагогики / Т.С. Дорохова // Педагогическое образование в России. – 2013. – № 2. – с. 49–53.
55. Джонсон-Лэйрд, Ф. Дедуктивное мышление: теория ментальных моделей / Ф. Джонсон-Лэйрд // Хрестоматия по психологии. Психология мышления. Под ред. Ю.Б.Гиппенрейтер, В.А. Спиридонова, М.В. Фаликман, В.В.Петухова – М., АСТ: Астрель, 2008. – с.170–181.
56. Дуванов, А. А. Азбука Роботландии / А. А. Дуванов, Н. Д. Шумилина. // Вестник Ярославского регионального отделения РАЕН.–2012.–№ 1.– с. 35–40.
57. Дьюи, Д. Психология и педагогика мышления. Пер. с англ. Н.М. Никольской / Д. Дьюи. – М.: Совершенство, 1997. – 208 с.
58. Ершов, А. П. О человеческом и эстетическом факторах в программировании / А.П. Ершов. // Кибернетика. – 1972. – № 5. – с. 95–99.
59. Ершов А. П. Избранные труды. / А.П. Ершов. ИСИ СО РАН, отв. ред. И. В. Поттосин. – Новосибирск: ВО «Наука», 1994. – 416 с.
60. Есаян А.Р. Творческая лаборатория Mathematica. Часть 2. Программирование, функции алгебры и анализа: учеб. пособие / А.Р. Есаян, В.Н. Чубариков, Н. М. Добровольский – Тула: Изд-во Тул. гос. пед. ун-та, – 2005. – 258 с.
61. Жемчужников, Д. Г. Методика обучения программированию, основанная на создании школьниками динамических компьютерных игр : автореф. дис. ... канд. пед. наук : 13.00.02 / Жемчужников Дмитрий Григорьевич. – Москва, 2013. –24 с.
62. Жужжалов, В.Е. Интеграционные методы изучения программирования в вузовском курсе информатики / В.Е. Жужжалов // Вестник МГПУ. Серия «Информатика и информатизация образования». – Москва, 2003. – № 1 (1). – с.53–54.
63. Жужжалов, В.Е. Совершенствование содержания обучения программированию на основе интеграции парадигм программирования: автореф. дис. ... д-ра пед. наук : 13.00.02 / Жужжалов Валерий Евгеньевич. – Москва, – 2004. – 45 с.

64. Забродина, О. М. Методика выравнивающе-развивающего обучения информационным технологиям студентов вуза в курсе информатики : автореф. дис. ... канд. пед. наук : 13.00.02 / Забродина Ольга Михайловна. – Волгоград, 2009. – 24 с.
65. Загвязинский, В.И. Теория обучения: современная интерпретация / В.И. Загвязинский. – М.: Академия, 2001. – 192 с.
66. Зинченко, В.П. Психологические основы педагогики (Психолого-педагогические основы построения системы развивающего обучения Д. Б. Эльконина – В. В. Давыдова): учеб. пособие / В.П. Зинченко. – М.: Гардарики, 2002. – 431 с.
67. Зинченко, В. П. Развитие зрения в контексте перспектив общего духовного развития человека / В.П. Зинченко // Вопросы психологии. – 1988. – № 6. – с.15–28.
68. Идеальная разработка ПО. Рецепты лучших программистов / Под ред. Э. Орама, Г.У. Уилсона. – СПб.: Питер, 2012. – 592 с.
69. Ильин, Г. Л. Проективное образование и реформация науки / Г. Л. Ильин. – М.: Флинта, 1993. – 216 с.
70. Ильин, Г.Л. Концепция проективного образования личности в контексте эволюции понимания предмета психологии / Г.Л. Ильин // Вестник практической психологии образования, 2010. № 3. – с. 27 – 35.
71. Кауфман, В. Ш. Языки программирования: концепции и принципы / В. Ш. Кауфман. – М.: ДМК-пресс, 2011. – 464 с.
72. Кедров, Б. М. О синтезе наук / Б. М. Кедров // Вопросы философии. – 1973. – № 3.
73. Кнут, Д.Э. Алгоритмическое мышление и математическое мышление / Д.Э.Кнут: [Электронный ресурс] – Режим доступа: <http://lib.rin.ru/doc/i/194604p.html>
74. Кнут, Д.Э. Искусство программирования для ЭВМ. т.1. Основные алгоритмы / Д.Э. Кнут. – М.: Мир, – 1976. – 736 с.

75. Коваль, Р.В. Использование инновационных образовательных технологий для практико-ориентированного обучения / Р. В. Коваль, Ф. А. Ефимов, А. А. Парфенов, А. А. Севрунов, А. Н. Черепанский // Актуальные проблемы авиации и космонавтики: материалы X Всероссийской научно-практической конференции творческой молодежи. – Красноярск: СибГАУ, 2014. – с. 368.
76. Колесникова, И.А. Педагогическое проектирование: учеб. пособие для высш. учебн. заведений / И.А. Колесникова, М.П. Горчакова-Сибирская; под ред. В.А. Сластёнина, И.А. Колесниковой. – 3-е изд., стер. – М.: Издательский центр «Академия». – 2008. – 288с.
77. Ксенчук, Е.В. Системное мышление. Границы ментальных моделей и системное видение мира / Е.В. Ксенчук. – М. : Изд. дом «Дело» РАНХиГС, 2011. – 368 с.
78. Кудрявцева, И. А. Методика обучения бакалавров физико-математического образования математическим основаниям парадигм программирования: дис. ... канд. пед. наук : 13.00.02/ Кудрявцева Ирина Андреевна. – СПб., 2005. – 243 с.
79. Кузнецов А.А. Основы информатики / А.А. Кузнецов. – М.: Дрофа. – 1998. – 175 с.
80. Кушниренко А. Г. Программирование для математиков / А.Г. Кушниренко, А.Г. Лебедев. – М.: Наука, 1988. – 384 с.
81. Кушниренко, А. Г. Методика преподавания основ алгоритмизации на базе системы КуМир / А. Г. Кушниренко, А. Г. Леонов. – М.: Первое сентября, 2007. – 64 с.
82. Кушниренко, А. Г. КуМир вернулся! / А. Г. Кушниренко, А. Г. Леонов. // Информатика. – 2009. – № 6 (583) – с. 8–10.
83. Лавров, С.С. Программирование. Математические основы, средства, теория: учебное пособие. / С.С. Лавров. – СПб.: БХВ-Петербург, 2001. – 314 с.
84. Лапчик, М.П. Вычисления. Алгоритмизация. Программирование: пособие для учителя / М.П. Лапчик. – М.: Просвещение, 1988. – 208 с.

85. Лапчик, М. П. Информатика и информационные технологии в системе общего и педагогического образования: монография / М. П. Лапчик. – Омск: Изд-во ОмГПУ, 1999. – 294 с.
86. Лебедева, Т. Н. Формирование алгоритмического мышления школьников в процессе обучения рекурсивным алгоритмам в профильных классах средней общеобразовательной школы: автореф. дис. ... канд. пед. наук: 13.00.02 / Лебедева Татьяна Николаевна – Екатеринбург, 2005. – 20 с.
87. Леонов, А.Г. Переход от непосредственного управления исполнителями к составлению программ в пропедевтическом курсе информатики / А.Г. Леонов, Ю.А. Первин // Ярославский педагогический вестник. – 2013. – № 4. – с. 17–30.
88. Леонтьев, А. Н. Деятельность. Сознание. Личность / А.Н. Леонтьев. – М.: Политиздат, 1975. – 304 с.
89. Леонтьев, А. Н. Психология образа / А.Н. Леонтьев // Вестник Моск. ун-та, серия 14. Психология. – 1979. – № 2. – с. 3—13.
90. Лернер, И. Я. Дидактические основы методов обучения / И.Я. Лернер. – М.: Педагогика, 1981. – 186 с.
91. Лоарер Э. Когнитивное обучение: история и методы / Э. Лоарер, М. Юто // Когнитивное обучение: современное состояние и перспективы. Под ред.Т. Галкиной, Э. Лоарер. М.: изд-во ИПРАН, 1997. – с. 17–33.
92. Ломов Б.Ф. Методологические и теоретические проблемы психологии / Б.Ф. Ломов. – М.: Наука, 1984. – 444с.
93. Лурия, А.Р. Лекции по общей психологии / А.Р. Лурия. Под ред. Е. Строгановой. – СПб.: Питер, 2006. – 320 с.
94. Люлькин, А.Е. Логическое программирование и его использование при обучении студентов-математиков / А.Е. Люлькин // Материалы междунар. науч. конф. «Информатизация обучения математике и информатике: педагогические аспекты». – Минск: БГУ, 2006. – с. 244-247.
95. Магалашвили В.В., Бодров В.Н. Ориентированная на цели визуализация знаний // Educational Technology & Society. – 2008. – № 11(1). – с.420–433.

96. Макарова, Н. В. Системно-деятельностный подход при обучении информатике в средней школе / Н. В. Макарова, Ю. Ф. Титова // Педагогическое образование в России. – 2002. – №5. – с. 88–95.
97. Малинецкий, Г.Г. Когнитивный вызов и информационные технологии / Г.Г.Малинецкий [и др.] // Препринты ИПМ им. М.В.Келдыша. – 2010. – № 46. – 28 с.
98. Матюшкин, А.М. Проблемные ситуации в мышлении и обучении / А. М. Матюшкин. – М.: Директмедиа Паблишинг, 2008. – 392 с.
99. Минский М. Фреймы для представления знаний. Пер. с англ. О. Н. Гринбаума; Под ред. Ф. М. Кулакова. / М. Минский. – М.: Энергия, 1979. – 151 с.
100. Монахов В.М. Педагогическая технология В.М. Монахова от А до Я: самоучитель проектирования учебного процесса / В.М. Монахов. – Липецк: ИРО, – 2007.
101. Мотова, Г.Н. Результаты обучения и оценка качества результата – по-европейски / Г.Н. Мотова: [Электронный ресурс] – Режим доступа: http://www.akvobr.ru/ocenka_kachestva_rezultata_po_evropeiski.html
102. Найсер, У. Познание и реальность / У. Найсер – М.: Прогресс, 1981. – 32 с.
- Науменко, М. А. Формирование системного стиля мышления студентов вуза в процессе компьютерного моделирования математических задач : дис. ... канд. пед. наук : 13.00.08 / Науменко Михаил Анатольевич. – Невинномысск, 2010.– 198 с.
103. Нефедова, В. Ю. Методические подходы к обучению школьников объектно-ориентированному программированию в системе дополнительного образования: на примере курса информатики для 8-10 классов: автореф. дис. ... канд. пед. наук: 13.00.02 / В. Ю. Нефедова. – М., 2011. – 19 с.
104. Новик, И. Б. О философских вопросах кибернетического моделирования / И. Б. Новик. – М.: Знание, – 1964.
105. Новиков А.М. Образовательный проект (методология образовательной деятельности) /А.М. Новиков, Д.А. Новиков. – М.: Эгвес, 2004. – 120 с.
106. Новиков, А. М. Педагогика: словарь системы основных понятий / А. М. Новиков. – М.: Издательский центр ИЭТ, 2013. – 268 с.

107. Новиков Д.А. Статистические методы в педагогических исследованиях (типовые случаи) / Д.А. Новиков.– М.: МЗ-Пресс, 2004. – 67 с.
108. Никашин, А.И. Системный подход в ознакомлении с окружающим миром и развитии фантазии/ А.И. Никашин, А.М. Страунинг Ростов-на-Дону: Академия, 1992.
109. Ньюэлл А. Движение в пространстве задачи / А. Ньюэлл, Г.А. Саймон // Хрестоматия по психологии. Психология мышления. Под ред. Ю.Б.Гиппенрейтер, В.А. Спиридонова, М.В. Фаликман, В.В.Петухова – М., АСТ: Астрель, 2008. – с.138–148.
110. О Концепции развития математического образования в Российской Федерации: [распоряжение Прав-ва РФ от 24.12.2013 N 2506-р]. [Электронный ресурс] – Режим доступа: <http://www.garant.ru/products/ipo/prime/doc/70452506/>
111. Об образовании в Российской Федерации: [федер. закон: принят Гос. Думой 21 дек. 2012 г.]. / Российская Газета: фед. выпуск № 5976 от 31 дек. 2012 г.
112. Об утверждении и введении в действие федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 010100 Математика (квалификация(степень) «бакалавр»: [приказ Минобрнауки России от 13.01.2010 N 8 (в ред. Приказа Минобрнауки РФ от 31.05.2011 N 1975)]. [Электронный ресурс] – Режим доступа: <http://www.garant.ru/products/ipo/prime/doc/97578/>
113. Об утверждении и введении в действие федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 050100 Педагогическое образование (квалификация (степень) «бакалавр»: [приказ Минобрнауки РФ от 22 декабря 2009 г. N 788 (с изменениями от 31.05.2011 г.)]. [Электронный ресурс] – Режим доступа: <http://fgosvo.ru/uploadfiles/fgos/5/20111207163943.pdf>
114. Об утверждении и введении в действие федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 161101 Системы управления летательными аппаратами (квалификация (степень) «специалист»: [приказ Минобрнауки РФ от 17 января

- 2011 г. N 70]. [Электронный ресурс] – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_114704/
115. Об утверждении плана мероприятий («дорожной карты») «Развитие отрасли информационных технологий»: [распоряжение Правительства РФ от 30.12.2013 N 2602-р (ред. от 05.12.2014)]. [Электронный ресурс] – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_172042/
116. Об утверждении профессионального стандарта «Программист»: [приказ Министерства труда и социальной защиты РФ от 18 ноября 2013 г. № 679н]. [Электронный ресурс] – Режим доступа: <http://www.garant.ru/products/ipo/prime/doc/70447858/>
117. Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.01 Математика (уровень бакалавриата): [приказ Минобрнауки России от 07.08.2014 N 943]. [Электронный ресурс] – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_168629/
118. Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата): [проект приказа Минобрнауки России]. [Электронный ресурс] – Режим доступа: http://www.osu.ru/docs/fgos/proekt/bak_01.03.02.doc
119. О'Коннор, Дж. Искусство системного мышления: Необходимые знания о системах и творческом подходе к решению проблем / Дж. О'Коннор, И. Макдермотт. – М.: Альпина Бизнес Букс, 2006. – 256 с.
120. Окулов, С.М. Развитие интеллекта школьника как принцип организации синергетической среды обучения информатике: автореф. дис. д-ра пед. наук / С. М. Окулов. – Киров, 2004. – 56 с.
121. Олешков, М.Ю. Педагогическая технология: проблема классификации и реализации / М.Ю. Олешков // Профессионально-педагогические технологии в теории и практике обучения: Сборник научных трудов. – Екатеринбург: РГППУ, 2005. – с. 5-19.

122. Орлова М.С. Система смешанного обучения программированию, ориентированная на формирование профессиональной компетентности: автореф. дис. ... канд пед. наук : 13.00.02 / Орлова Марина Сергеевна. – Москва, – 2009. – 22 с.
123. Официальный информационный портал единого государственного экзамена. [Электронный ресурс] – Режим доступа: <http://www.ege.edu.ru/ru/main/satistics-ege/>
124. Пак, Н.И. Проективный подход в обучении как информационный процесс: монография / Н.И. Пак – Красноярск: РИО КГПУ, 2008. –112 с.
125. Пак, Н.И. Информационный подход и электронные средства обучения: монография / Н.И. Пак. – Красноярск: РИО КГПУ, 2013. –196 с.
126. Пак, Н.И. О модели мышления и ментальных схемах / Н.И. Пак // Практико-ориентированное обучение в профессиональном образовании: проблемы и пути развития: материалы научно-практической конференции в рамках XVIII Междунар. научной конференции «Решетневские чтения». – Красноярск: СибГАУ, 2014. – С. 306–310.
127. Паронджанов, В.Д. Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации / В.Д. Паронджанов. – М.: ДМК Пресс, 2012. – 520 с.
128. Пейперт, С. Переворот в сознании: дети, компьютеры и плодотворные идеи / С. Пейперт. – М.: Педагогика, 1989. – 222 с.
129. Первин, Ю. А. Методика раннего обучения информатике / Ю. А. Первин. – 2-е издание. – М. : БИНОМ, Лаборатория знаний, 2008. – 288 с.
130. Петров, А.Н. Совершенствование методики обучения объектно-ориентированному программированию на основе объектно-ориентированного проектирования: автореф. дис. ... канд пед. наук : 13.00.02 / Петров Алексей Николаевич. – Москва, – 2009. – 19 с.
131. Пестряков П. Н. Научно-педагогические условия совершенствования подготовки студентов к проективной деятельности: дис. ... канд. пед. наук : 13.00.08 / Пестряков Павел Николаевич. – Липецк, – 2001. – 212 с.

132. Пиаже, Ж. Природа интеллекта / Ж. Пиаже // Хрестоматия по психологии. Психология мышления. Под ред. Ю.Б.Гиппенрейтер, В.А. Спиридонова, М.В. Фаликман, В.В.Петухова – М., АСТ: Астрель, 2008. – с.78–89.
133. Поляков, К.Ю. Алгоритмизация и программирование / К.Ю. Поляков, А.П. Шестаков, Е.А. Еремин // Информатика, 2011. – № 17. – с. 4-33.
134. Психологическая энциклопедия / под.ред. Р. Корсини, А. Ауэрбаха. – 2-е изд. – СПб.: Питер, 2006. – 1832 с.
135. Пышкало, А.М. Методическая система обучения геометрии в начальной школе: авторский доклад по монографии «Методика обучения элементам геометрии в начальных классах», представленной на соискание ... д-ра пед. наук / А.М. Пышкало. – М.: Академия пед. наук СССР, 1975. – 60 с.
136. Решетова, З. А. Формирование системного мышления в обучении: учебное пособие для вузов / З. А. Решетова. – Москва: ЮНИТИ-ДАНА, 2002. – 344 с.
137. Ришар Ж.-Фр. Ментальная активность: понимание, рассуждение, нахождение решений. – М.: Издательство «Институт психологии РАН», 1998. – 232 с.
138. Роберт, И.В. Современные информационные технологии в образовании: дидактические проблемы; перспективы использования / И.В. Роберт. – М.: ИИО РАО, 2010. – 140 с.
139. Роберт, И. В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты) / И.В. Роберт. – М.: БИНОМ: Лаборатория знаний, 2013. – 398 с.
140. Роберт, И.В. Дидактика периода информатизации образования / И.В. Роберт // Педагогическое образование в России. – 2014. – №8. – с. 110–118.
141. Российская педагогическая энциклопедия в двух томах: Том I / Гл. ред. В.В. Давыдов. – М.: Большая Российская энциклопедия, 1993. – 608 с.
142. Российская педагогическая энциклопедия в двух томах: Том II / Гл. ред. В.В. Давыдов. – М.: Большая Российская энциклопедия, 1999. – 560 с.
143. Рубинштейн, С. Л. Основы общей психологии / С.Л. Рубинштейн. – СПб: Питер, 2000. – 712 с.

144. Салангина, Н. Я. Реализация линии алгоритмизации в курсе «Языки и методы программирования» физико-математических специальностей педагогических вузов: дис. ... канд пед. наук : 13.00.02 / Салангина, Надежда Яковлевна. – Москва, 1999. – 154 с.
145. Саранцев Г. И. Теория, методика и технология обучения / Г. И. Саранцев // Педагогика. 1999.– № 1.– с. 19–24.
146. Сериков Г. Н. Управление образованием: Системная интерпретация / Г. Н. Сериков. – Челябинск: Изд-во ЧГТУ «Факел», 1998. – 664 с.
147. Светличная, С.В. Методика проективно-рекурсивного обучения учителей начальных классов в области ИКТ в муниципальной системе повышения квалификации: автореф. дис. ... канд пед. наук : 13.00.02 / Светличная Светлана Викторовна. – Красноярск, – 2012. – 25 с.
148. Селевко, Г. К. Энциклопедия образовательных технологий / Г. К. Селевко. – М.: Народное образование, 2005. – 552 с.
149. Семакин, И. Г. Эволюция школьной информатики / И. Г. Семакин // Информатика в школе. – 2011. – №5. – с. 2–7 .
150. Семакин, И.Г. Формирование личностных и метапредметных результатов обучения информатике на профильном уровне / И.Г. Семакин // Информационные ресурсы в образовании: материалы Международной научно-практической конференции. – Нижневартонск: НГГУ, 2012. – С. 14–17.
151. Скаткин, М.Н. Проблемы современной дидактики / М.Н. Скаткин. 2 – е изд. – М.: Педагогика, 1984. – 96 с.
152. Скибицкий, Э.Г. Методика профессионального обучения / Э.Г. Скибицкий, И.Э. Толстова, В.Г. Шефель. – Новосибирск: НГАУ, 2008. – 166 с.
153. Сластенин, В. А. Педагогика / В. А. Сластенин, И. Ф. Исаев, Е. Н. Шиянов. – М., 2007. – 170 с.
154. Слинкина, И.Н. Использование компьютерной техники в процессе развития алгоритмического мышления у младших школьников: дис. ... канд пед. наук : 13.00.02 / Слинкина Ирина Николаевна. – Екатеринбург, – 2000. – 192 с.

155. Смирнова, О. В. Разработка содержания обучения программированию в системе информационной подготовки бакалавра математики: дис. ... канд. пед. наук : 13.00.02 / О. В. Смирнова. – Москва, 2005 – 195 с.
156. Сорокина Н.Д. О компетентностном подходе и проективном обучении (на примере курса «Конфликтология» / Н.Д. Сорокина // Социологические исследования, 2010. – № 5. – с.123–128.
157. Слугина Н. Л. Повышение уровня подготовки кадров в области современных информационных технологий на базе центра компетенций / Н.Л. Слугина, В.М. Гриняк // Современные проблемы науки и образования. – 2012. – № 5. – с. 205–205.
158. Солсо Р.Л. Когнитивная психология / Р.Л. Солсо – пер. с англ. – М.: Триволта. – 1996. – 600 с.
159. Сороко, Э. М. Золотые сечения, процессы самоорганизации и эволюции систем: введение в общую теорию гармонии систем / Э. М. Сороко. – 2-е изд. – М., 2006.
160. Страуструп, Б. Язык программирования C++ / Б. Страуструп. – СПб.: изд-во Невский диалект, 1999. – 991 с.
161. Султанова Т.А. Формирование проективных умений студентов педагогического колледжа: дис. ... канд. пед. наук: 13.00.01 / Т.А. Султанова. – Оренбург, 2007. – 197 с.
162. Сычев, И.А. Педагогические условия формирования элементов системного мышления учащихся старших классов.: дис. ... канд. пед. наук: 13.00.01/ Барнаул, 2009. – 197 с.
163. Табаченко, Т. С. Проблемы когнитивного обучения в педагогическом образовании / Т.С. Табаченко // Среднее профессиональное образование. – 2007. – № 2. – с. 2–4.
164. Татур, Ю.Г. Высшее образование: методология и опыт проектирования / Ю.Г. Татур. – М.: Логос, Университетская книга, 2006.
165. Татур, Ю.Г. Как повысить объективность измерения и оценки результатов образования / Ю.Г. Татур // Высшее образование в России, 2010. – № 5. – с. 22–31.

166. Толстова, Н.С. Бимодульность как условие построения адаптивных методических систем обучения программированию: автореф. дис. ... канд. пед. наук: 13.00.02 /Толстова Наталья Сергеевна. – Екатеринбург.– 2005.–26 с.
167. Угринович, Н. Д. Информатика и ИКТ. 8–11 классы / Н. Д. Угринович. – М.: Лаборатория знаний. – 2010. – 178 с.
168. Уразова,М.Б. Фреймовая технология как способ формирования самостоятельного мышления студентов педагогических вузов / М.Б. Уразова, Ш.Н. Эшпулатов // Вестник Томского государственного педагогического университета. – 2011. – № 4. – с. 163–165.
169. Фалина И.Н. Методика выравнивающего и развивающего обучения информатике в физико-математических классах: дис. канд. пед. наук :13.00.02 / Фалина Ирина Николаевна. –Москва, – 2000. – 130 с.
170. Федосеева, Ю.В. Развитие системного мышления студентов колледжа на основе использования информационных технологий: дис. ... канд. пед. наук: 13.00.01/ Ю.В. Федосеева. – Магнитогорск, 2009. – 197 с.
171. Холодная, М.А. Психология интеллекта: парадоксы исследования / М.А. Холодная. –2-е изд., перераб. и дополн. – СПб.: Питер, 2002. – 272 с.
172. Холодная, М. А. Когнитивные стили. О природе индивидуального ума / М. А. Холодная. – СПб: Питер, 2004. – 384 с.
173. Хуторской, А. В. Ключевые компетенции как компонент личностно-ориентированного образования / А. Хуторской // Народное образование. – 2003. – №2. – с. 58–64.
174. Хуторской, А. В. Педагогическая инноватика / А. В. Хуторской. – М.: Академия, 2008. – 256 с.
175. Чельшкова, М.Б. Аттестация выпускников вузов в рамках компетентностного подхода / М.Б. Чельшкова // Вестник КГУ им. Н. А. Некрасова. – 2012. – № 6.– с. 270–273.
176. Черниговская, Т.В. Если зеркало будет смотреться в зеркало, что оно там увидит (к вопросу об эволюции языка и сознания) / Т.В. Черниговская // В кн.:

Когнитивные исследования / под ред. Ю.И. Александрова. – М.: Изд-во Института психологии РАН, 2010. – с. 13–37.

177. Черникова, И.В. Когнитивные науки и когнитивные технологии в зеркале философской рефлексии / И.В. Черникова // Эпистемология и философия науки. – 2011. – Т. XXVII. – № 1. – с. 101–117.

178. Шамшев, А. Б. Алгоритмическое мышление при решении задач (на примере языка C#) / А. Б. Шамшев, К. В. Святков. – Ульяновск: УлГТУ, 2011. – 113 с.

179. Шапиро С.И. От алгоритмов – к суждениям (Эксперименты по обучению элементам математического мышления) / С.И. Шапиро. – М.: Сов. радио. – 1973. – 288 с.

180. Швецкий М.В. Методическая система фундаментальной подготовки будущих учителей информатики в педагогическом вузе в условиях двухступенчатого образования: дис. ... д-ра. пед. наук: 13.00.02 / М.В. Швецкий – СПб., 1994. – 445 с.

181. Шень, А.Х. Программирование: теоремы и задачи / А.Х. Шень. – М.: МЦНМО, 2004. – 296 с.

182. Шрагина, Л.И. Подходы к диагностике и развитию системного мышления школьников [электронный ресурс] – Режим доступа: <http://www.vashpsixolog.ru/lectures-on-the-psychology/119-conferences-and-reports-on-psychology/1568-podxody-k-diagnostike-i-razvitiyu-sistemnogo-myshleniya-shkolnikov>

183. Энциклопедия эпистемологии и философии науки / Сост. и общ. ред. И.Т. Касавин. М.: «Канон+» РООИ «Реабилитация», 2009.

184. Ядровицкая, М.В. Моделирование в реализации когнитивного обучения / М.В. Ядровицкая // Образовательные технологии и общество. – № 2. – т.15. – 2012. – с. 602–616

185. Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J. and Wittrock, M.C., Eds. 2001. A taxonomy for learning and teaching and assessing: A revision of Bloom's taxonomy of educational objectives / Addison Wesley Longman, Inc., 2001.

186. Ausubel, D.P. Educational Psychology, a Cognitive View / New York: Holt, Rinehart, 1968.
187. Bennedsen, J.B., Caspersen, M.E. Failure Rates in Introductory Programming / SIGCSE Bulletin, – v. 39, n. 2, 2007. – pp. 32–36.
188. Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. Taxonomy of Educational Objectives: Handbook 1 Cognitive Domain. / London: Longmans, Green and Co Ltd. – 1956.
189. Caspersen, M.E. Educating Novices in the Skills of Programming :DAIMI PhD Dissertation PD-07-4. / Aarhus, Denmark. – 2007. – 311 p.
190. Collins, A.M., Quillian, M.R. Retrieval Time from Semantic Memory / J. Verb. Learn, and Verb. Behav., 8. – 1969. – pp. 240–248.
191. Crescenzi, P., Nocentini, C. Fully integrating algorithm visualization into a CS2 course: A two-year experience / Proceedings of the 12th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2007. –pp. 296–300.
192. Dougiamas, M., Taylor, P. Moodle: Using learning communities to create an open source course management system / Proceedings of the World conference on educational multimedia, hypermedia and telecommunications, 2003. – v.1 – pp. 171–178.
193. Fuller, U., Johnson, C., Ahoniemi, T., Cukierman, D., Hernan-Locada, I., et al. Developing a computer science-specific learning taxonomy / ACM SIGCSE Bulletin, 39 (4) – 2007. – pp. 152–170.
194. Lengler, R., Eppler, M. A Periodic Table Of Visualization Methods: [Электронный ресурс] – Режим доступа: http://www.visual-literacy.org/periodic_table/periodic_table.html
195. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. A multi-national, multi-institutional study of assessment of programmingskills of first-year CS students. / ACM SIGCSE Bull., 33(4). – 2001. – p.125–180.
196. Novak, J.D., Gowin, D.B. Learning How to Learn / Cambridge Cambridge University Press, 1984.

197. Shaffer, C. A. Getting algorithm visualizations into the classroom / C.A. Shaffer, et al. // Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, 2011. – pp. 129–134.
198. Sloan, R.H., Troy, P. CS 0.5: A better approach to introductory computer science for majors / Proceedings of the 39th SIGCSE technical symposium on computer science education, 2008. – pp. 271–275.
199. Soloway, E. Cognitive strategies and looping constructs: An empirical study / E. Soloway, J. Bonar, et al. // Communications of the ACM 26(ii), 1983. – pp. 853–860.
200. Tapscott, D. Grown Up Digital: How the Net Generation is Changing Your World / McGraw-Hill, 2008.
201. Terzieva, T. Some criteria and indicators for diagnosis of the forming of algorithmic thinking in computer science / Plovdiv University Scientific Works, vol. 38, book 3. – 2011. – pp. 123–132.
202. Utting, I., Tew, A.E., McCracken, M., Thomas, L., Bouvier, D., Frye, R., Paterson, J., Caspersen, M.E., Kolikant, Y.B-D., Sorva, J., and Wilusz, T. A Fresh Look at Novice Programmers' Performance and Their Teachers' Expectations / Proceedings of the 18th Annual Conference on Innovation and Technology in Computer Science Education Working Group Reports, ITiCSE 2013, Canterbury, Kent, UK, July 2013, pp. 15 – 32.
203. Whalley, J.L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P. K.A., and Prasad, C. An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies / Proceedings of the 8th Australasian Conference on Computing Education – Vol. 52, Hobart, Australia, Australian Computer Society, Inc, 2006. – pp. 243–252.
204. Wing, J.M. Computational Thinking / Communications of the ACM. –Vol. 49, No. 3. – 2006. – pp. 33–35.
205. Winslow, L. E. Programming Pedagogy – A Psychological Overview, SIGCSE Bulletin 28(3), 1996, pp. 17–22.

Приложения

Приложение 1

Задание на входном тестировании

- 1) Назвать базовые алгоритмические структуры (с их помощью можно выразить любой алгоритм).
- 2) Написать на естественном языке (можно на алгоритмическом языке или в виде блок-схемы) **подробный** алгоритм, демонстрирующий, как Вы обычно добираетесь до места учёбы (ИМиФИ). Отметить в этом алгоритме базовые алгоритмические структуры.

- 3) Дан фрагмент программы на алгоритмическом языке:

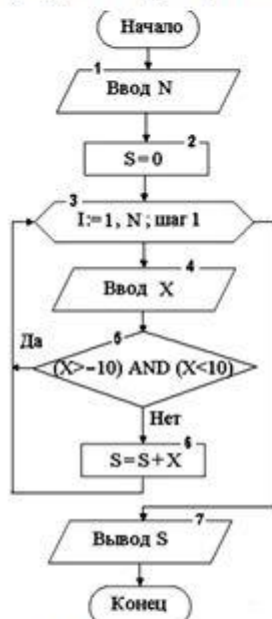
```

A := 3
B := 5
A := B
B := A
Вывод A, B

```

Что будет напечатано в результате выполнения этого фрагмента?

- 4) Дана следующая блок-схема:



а) Что вычисляет данный алгоритм? (Что такое S?)

б) Каков результат выполнения программы, реализующей данный алгоритм с входными данными:

$N = 5$, $X = -10; -4; 2; 8; 14$.

в) Если в блоке 5 изменить логическую операцию AND («И») на логическую операцию OR («ИЛИ»), изменится ли результат?

г) Какие изменения нужно внести в блок-схему, если предполагается использовать массив для входных данных? Что, на Ваш взгляд, даёт использование массива? (Достоинства, недостатки?)

- 5) Алгоритм написан на алгоритмическом языке:

```

нач
...
ввод X, N
...
нц для I от 1 до N
    P := P * X
кц
...
кон

```

а) Что должно стоять вместо многоточий?

б) Что вычисляет данный алгоритм?

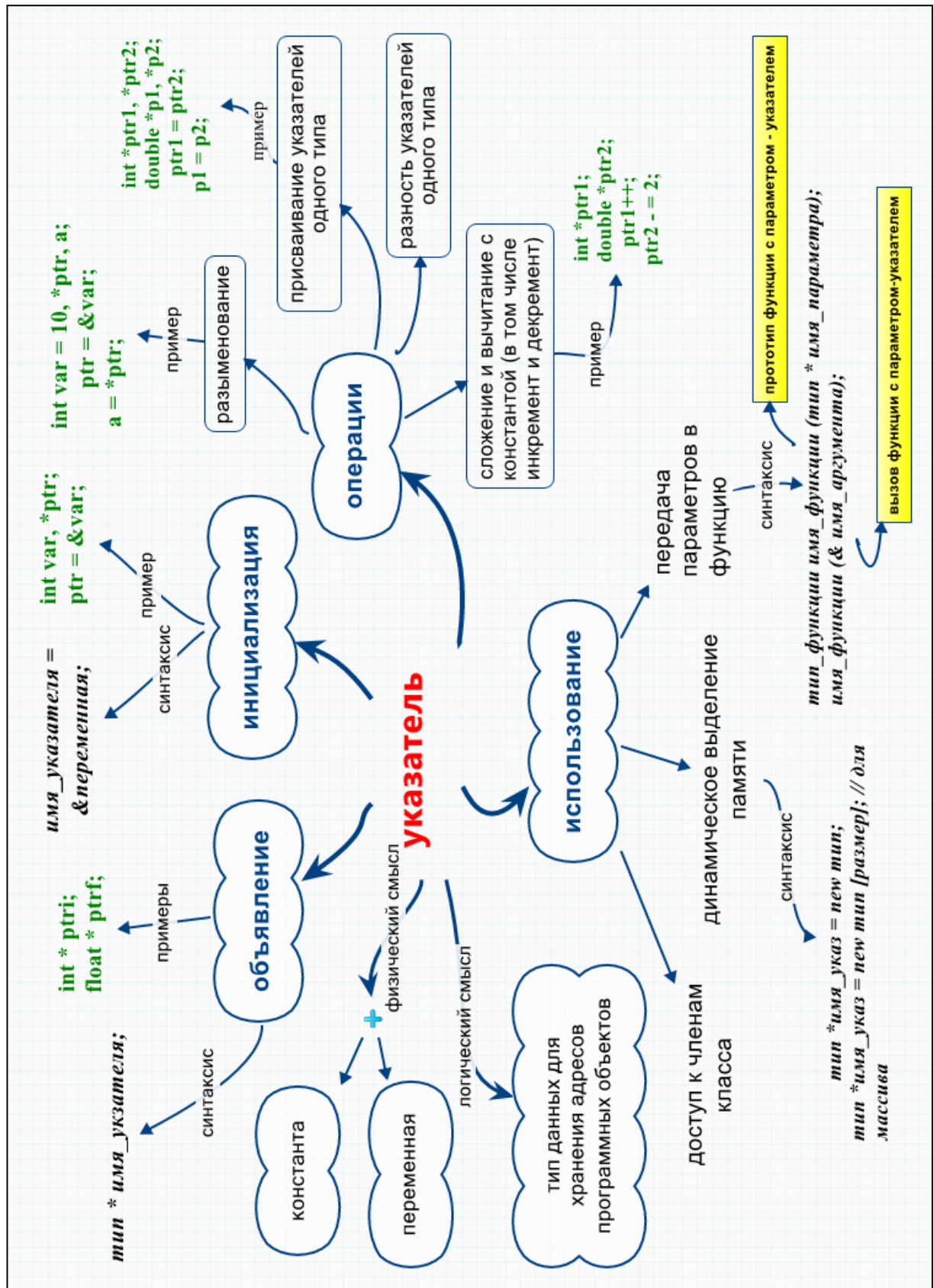
в) * Написать программу, реализующую данный алгоритм, на языке высокого уровня.

- 6) Написать (на алгоритмическом языке или языке высокого уровня) программу, в которой находится наибольшее из трёх заданных целых чисел.

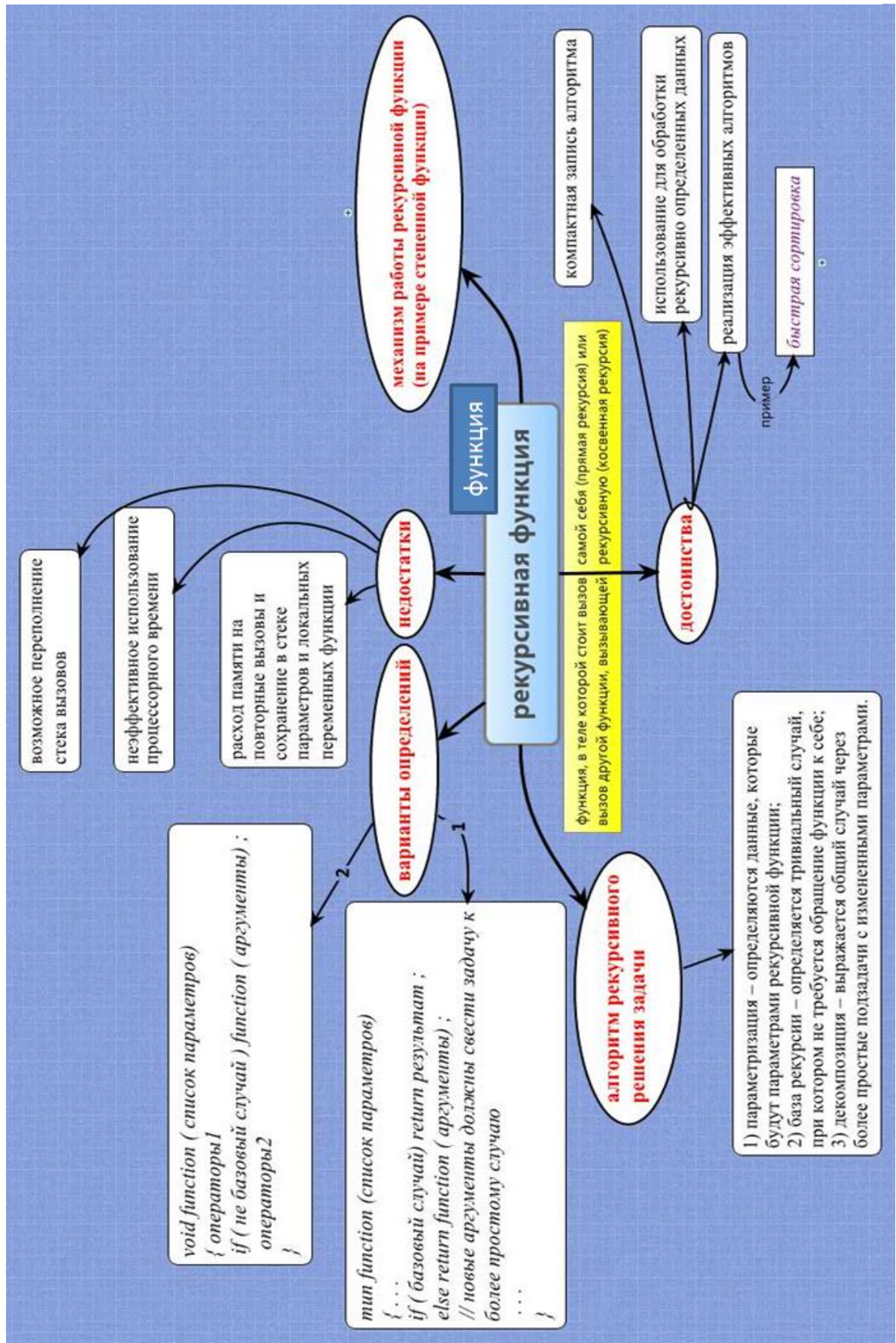
- 7) а) Написать, что Вы понимаете под термином «система».

б) Приведен список дисциплин, которые Вы изучаете в 1 семестре: 1) алгебра; 2) аналитическая геометрия; 3) введение в специальность; 4) информатика; 5) история; 6) мат. анализ; 7) музыка; 8) физ. культура. Изобразите этот список в виде системы (Подсказка: внешние и внутренние связи, подсистемы, надсистемы ...)

Понятийная ментальная карта по теме «указатель»



Понятийная ментальная карта по теме «Рекурсивные функции»



**Контрольная работа по темам «Состав языка программирования» и
«Тип данных»**

ВАРИАНТ 1.

1. Дать определение типа данных. (1балл)
2. Какие ключевые слова используются для объявления целых типов данных?
(1 балл)
3. Среди наборов символов выбрать те, которые можно использовать в качестве идентификаторов: **C++**, **1Lab**, **Лаб1**, **Lab_1**, **sin**, **масса**. (1 балл)
4. Указать порядок выполнения операций в выражении $(5 + -x + +) - \sin(x) / x$
(1 балл)
5. Написать программу, в которой
 - а) объявляются две переменные вещественного типа, беззнаковая символьная константа, логическая переменная, инициализированная значением TRUE;
 - б) одной переменной вещественного типа присвоить значение константы, записанной в десятичном формате, другой присвоить значение константы в экспоненциальном формате;
 - в) изменить значение логической переменной с помощью операции инкремента;
 - г) вывести значения всех переменных на печать.(3 балла)
6. Дан фрагмент программы:

```
char a='a';  
double b=4.5;  
int c;  
c=a++ +b;  
cout<<"a"<<a<<"b"<<b<<"c"<<c;
```

Что будет на экране компьютера в результате выполнения этого фрагмента?
(3 балла)

ВАРИАНТ 2.

1. Что такое литеральная константа? Какие литеральные константы Вы знаете?
(1 балл)

2. Как можно объявить константы в программе? (1 балл)

3. Среди наборов символов выбрать те, которые можно использовать в качестве идентификаторов: **C+K**, **1variable**, **x1_2**, **sin2x**, **volume**, **ТЕМА**.

(1 балл)

4. Указать порядок выполнения операций в выражении $(z + -x * \cos(y)) < \sin(x) + 1$
(1 балл)

5. Написать программу, в которой

а) объявляются две константы разных вещественных типов, беззнаковая целая короткая переменная, инициализированная значением 10, две логические переменные;

б) логическим переменным присвоить значения «ИСТИНА» и «ЛОЖЬ»;

в) сложить значения логических переменных и присвоить целой переменной;

г) вывести значения всех переменных на печать, объяснить полученные результаты с точки зрения преобразования типов.

(3 балла)

6. Дан фрагмент программы:

```
char a, int b;  
const x=4.5;  
float c=.0;  
c+=a+b;  
25+=10;  
cout<<' c='<<c<<\n;
```

Какие ошибки допущены в этом фрагменте?

(3 балла)

Анкета «Показатели: начало обучения – конец 1 курса» на концептуально-констатирующем этапе эксперимента

1. Укажите МЕСТО ПОЛУЧЕНИЯ СРЕДНЕГО ОБРАЗОВАНИЯ:
 - а) край (область), населённый пункт _____
 - б) образовательное учреждение (ср. школа, гимназия, лицей) _____
2. Укажите ОЦЕНКУ В АТТЕСТАТЕ + баллы ЕГЭ
 - а) по математике _____ б) по информатике _____
3. С КАКОЙ ЦЕЛЬЮ Вы поступили в институт математики (выбрать вариант ответа):
 - а) не прошёл по конкурсу на другие специальности;
 - б) хотел бы преподавать математику или информатику в школе;
 - в) хотел бы работать программистом;
 - г) планирую заниматься научной деятельностью по профилю образования (аспирантура, защита диссертации);
 - д) целенаправленно поступал в ИМ (нравится математика и (или) информатика), но пока нет планов на будущее.
4. ИЗУЧАЛИ ли Вы какой-либо ЯЗЫК ПРОГРАММИРОВАНИЯ до поступления в ВУЗ? Если изучали, то укажите, где это было (выбрать вариант ответа и указать язык программирования)
 - а) на уроках информатики в школе
 - б) на дополнительных занятиях по информатике в школе
 - в) в другом месте помимо школы
 - г) самостоятельно
 - д) не изучал
5. Умели ли Вы использовать БЛОК-СХЕМЫ или другие способы записи алгоритмов до поступления в ВУЗ? (если нет, то почему?) _____
6. Насколько УСПЕШНО Вы ОСВАИВАЕТЕ курс программирования (выбрать вариант ответа):
 - а) могу самостоятельно написать программу по всем пройденным темам курса программирования;
 - б) могу написать программу по аналогии с разобранный программой;
 - в) могу разобраться в готовой программе, но не пишу программы самостоятельно;
 - г) ничего не понимаю в программировании.
7. Если Вы испытываете ТРУДНОСТИ в изучении языка программирования, то причина этого, на Ваш взгляд, (можно отметить несколько пунктов)
 - а) собственное нежелание научиться программировать;
 - б) недостаточность аудиторных часов курса программирования;
 - в) недостаток бумажных источников информации (учебники, методические пособия);
 - г) недостаток электронных источников информации;
 - д) качество преподавания;
 - е) другие причины.
8. Считаете ли Вы, что практические задания по курсу программирования должны быть связаны с МАТЕМАТИЧЕСКИМИ ДИСЦИПЛИНАМИ? (укажите, с какими) _____

Входная анкета первокурсника

1. Ф.И.О. группа _____
2. Укажите место получения среднего образования:
 - а) населённый пункт _____ б) обр. учреждение _____
3. Укажите оценку в аттестате и количество баллов ЕГЭ
 - а) по математике _____ б) по информатике _____
4. С какой ЦЕЛЬЮ Вы поступили в ИМиФИ (выбрать вариант ответа):
 - а) не прошёл по конкурсу на другие специальности;
 - б) хотел бы преподавать математику или информатику в школе;
 - в) хотел бы работать программистом;
 - г) планирую заниматься научной деятельностью по профилю образования (аспирантура и т.д.);
 - д) целенаправленно поступал в ИМиФИ (нравится математика и(или) информатика), но пока нет планов на будущее;
 - е) по другим причинам _____
5. Есть ли у Вас ВОЗМОЖНОСТЬ ЗАНИМАТЬСЯ ПРОГРАММИРОВАНИЕМ дома (в общежитии):
 - а) есть компьютер (да /нет) б) выход в Интернет (если да, указать адрес электронной почты _____)
6. Умеете ли Вы использовать БЛОК-СХЕМЫ или другие способы записи алгоритмов (если нет, то по какой причине?) _____
7. Изучали ли Вы какой-либо ЯЗЫК ПРОГРАММИРОВАНИЯ до поступления в ВУЗ? Если изучали, то укажите, где это было (выбрать один или более вариантов ответа и указать ЯП):
 - а) на уроках информатики в школе _____
 - б) на дополнительных занятиях по информатике в школе _____
 - в) в другом месте помимо школы _____
 - г) самостоятельно _____
 - д) не изучал
8. Есть ли у Вас ЖЕЛАНИЕ НАУЧИТЬСЯ ПРОГРАММИРОВАТЬ (выбрать вариант ответа):
 - а) Я уже классный программист;
 - б) есть твёрдое желание;
 - в) желания нет, но буду стараться, т. к. это обязательная дисциплина;
 - г) программирование – это не моё призвание: предпочитаю математику / Я скорее гуманитарий;
 - д) затрудняюсь с ответом.
9. Какой СТИЛЬ ОБУЧЕНИЯ, на Ваш взгляд, будет Вам ближе:
 - а) предпочитаю писать программы и работать самостоятельно;
 - б) хотел бы получать помощь преподавателя индивидуально;
 - в) хотел бы работать в команде;
 - г) затрудняюсь с ответом.
10. Ваша СФЕРА ИНТЕРЕСОВ? _____