УДК 004.588

РАЗРАБОТКА АЛГОРИТМОВ ПОДДЕРЖКИ МЕТОДОВ АКТИВНОГО ОБУЧЕНИЯ НА БАЗЕ МНОГОПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРАКТИВНОГО ВЗАИМОДЕЙСТВИЯ

Е.А. Павлова, старший преподаватель кафедры программного обеспечения
тел.: 8(912)-926-5789; e-mail: e.a.pavlova@utmn.ru
М.С. Воробьева, канд. тех. наук, доцент кафедры программного обеспечения
тел.: 8(3452)-46-14-84; e-mail: m.s.vorobyeva@utmn.ru
ФГБОУ ВО «Тюменский государственный университет»
Институт математики и компьютерных наук

SUPPORTIVE ALGORITHM DEVELOPMENT FOR ACTIVE LEARNING METHODS BASED ON MULTI-USER INTERACTION.

Elena A. Pavlova, assistant Professor,
Marina S. Vorobyeva, cand. tech. sci., associate Professor,
Department of Software Development
Institute of Mathematics and Computer Science
Tyumen State University

Аннотация. В статье описан алгоритм формирования вариантов заданий разного уровня сложности, с разными параметрами, по различным темам и с дополнительными возможностями, приведён алгоритм формирования проверочных заданий во время сеанса тестирования.

Ключевые слова. Система автоматизированного формирования индивидуальных заданий, электронное обучение, моделирование сложных объектов.

Внедрение интерактивных форм обучения — одно из важнейших направлений совершенствования подготовки студентов в современном вузе. Данные исследований подтверждают, что использование активных и интерактивных подходов является наиболее эффективным путём, способствующим обучению студентов и формированию необходимых компетенций. Информатизация образования приводит к необходимости разработки и внедрения инновационных образовательных методик и технологий, способствующих появлению новых форм обучения.

Успешное освоение программирования требует от студента выполнения большого числа лабораторных работ, а от преподавателя – составления за короткое время пакета индивидуальных заданий разной сложности с параметрами, обеспечивающими адаптацию заданий к различным уровням освоения дисциплины (темы). При выполнении работы студенты, с одной стороны, желают минимизировать время решения задач, с другой стороны — набрать количество баллов, соответствующее уровню их знаний и притязаний по поводу оценки.

Важную роль в изучении программирования играет контроль знаний, включающий в себя проверочные работы, тестирование различного вида. Одним из популярных методов контроля знаний является компьютерное тестирование различных видов, но составление большого числа таких оценочных средств требует много времени, кроме того, при создании компьютерных систем контроля знаний важное значение имеет автоматическая генерация вариантов заданий непосредственно во время сеанса тестирования.

Комплексное решение этих проблем может быть реализовано с помощью системы, включающей в себя модуль автоматизированного формирования и распределения пакета индивидуальных заданий, учитывающий результаты сдачи

предшествующих работ, и модуль автоматической генерации набора тестовых заданий по программированию, с вариантами ответов на задания, сформированными во время сеанса тестирования, выбором языка, на котором будут создаваться задания, проверкой результатов.

Модуль автоматизированного формирования пакета индивидуальных заданий

Создание и распределение генерируемых заданий предполагает прохождение нескольких стадий.

На первом этапе выбирается подмножество шаблонов, на основании которых будут сгенерированы индивидуальные варианты заданий.

На втором этапе формируется полный текст работ, состоящий из одного или нескольких шаблонов заданий, с указанием общего количества баллов за работу и за каждое задание в отдельности.

На третьем этапе автоматически комплектуется индивидуальный пакет заданий из полного текста работ, учитывая несколько показателей: индивидуальные особенности пользователя, сложность, максимальное число баллов за каждое задание и работу в целом, уровень выполнения предыдущей работы. При генерации набора используется алгоритм, учитывающий приоритет включения заданий в работу и нормирующий сумму баллов за задания.

На четвёртом этапе по всем выполненным работам формируется сводная ведомость с разбиением студентов на группы и определением рекомендаций по изучению тем.

Модуль генерации интерактивных заданий

При реализации данного модуля нужно учесть возможность получения заданий на нескольких языках программирования.

В работе использован подход, основанный на создании транслятора. В этом случае учебное задание пишется на адаптированном псевдокоде, удобном для трансляции на целевой язык.

Пусть G^k – множество характеристик сложности объекта, Q^k – множество показателей уровня интерактивности объекта, Z^k – множество структур учебных заданий, состоящих из базовых элементов (множество A), основных зависимостей (множество B), функциональных элементов (множество C), функциональных зависимостей (множество D).

$$z_i^k = \begin{pmatrix} a_1 & \dots & a_r \\ b_1 & \dots & b_p \\ c_1 & \dots & b_m \\ d_1 & \dots & d_s \end{pmatrix}, a_j \in A, b_j \in B, c_j \in C, d_j \in D \qquad (1)$$

Множество объектов учебного проверочного задания описывает неоднородную систему T, где t_i^k – объект учебного проверочного задания на языке программирования k. Данный объект можно описать набором характеристик $t_i^k = \{z_i^k, g_i^k, q_i^k\}, z_i^k \in Z^k, g \in G^k, q_i^k \in Q^k$.

Характеристиками объекта могут быть: уровень сложности задания, уровень интерактивности задания, совокупность основных, функциональных элементов и функциональных зависимостей, которые определяются в зависимости от показателя сложности задания.

На рисунке показан алгоритм формирования тестового задания, исходными данными для которого являются учебные задания на псевдокоде.

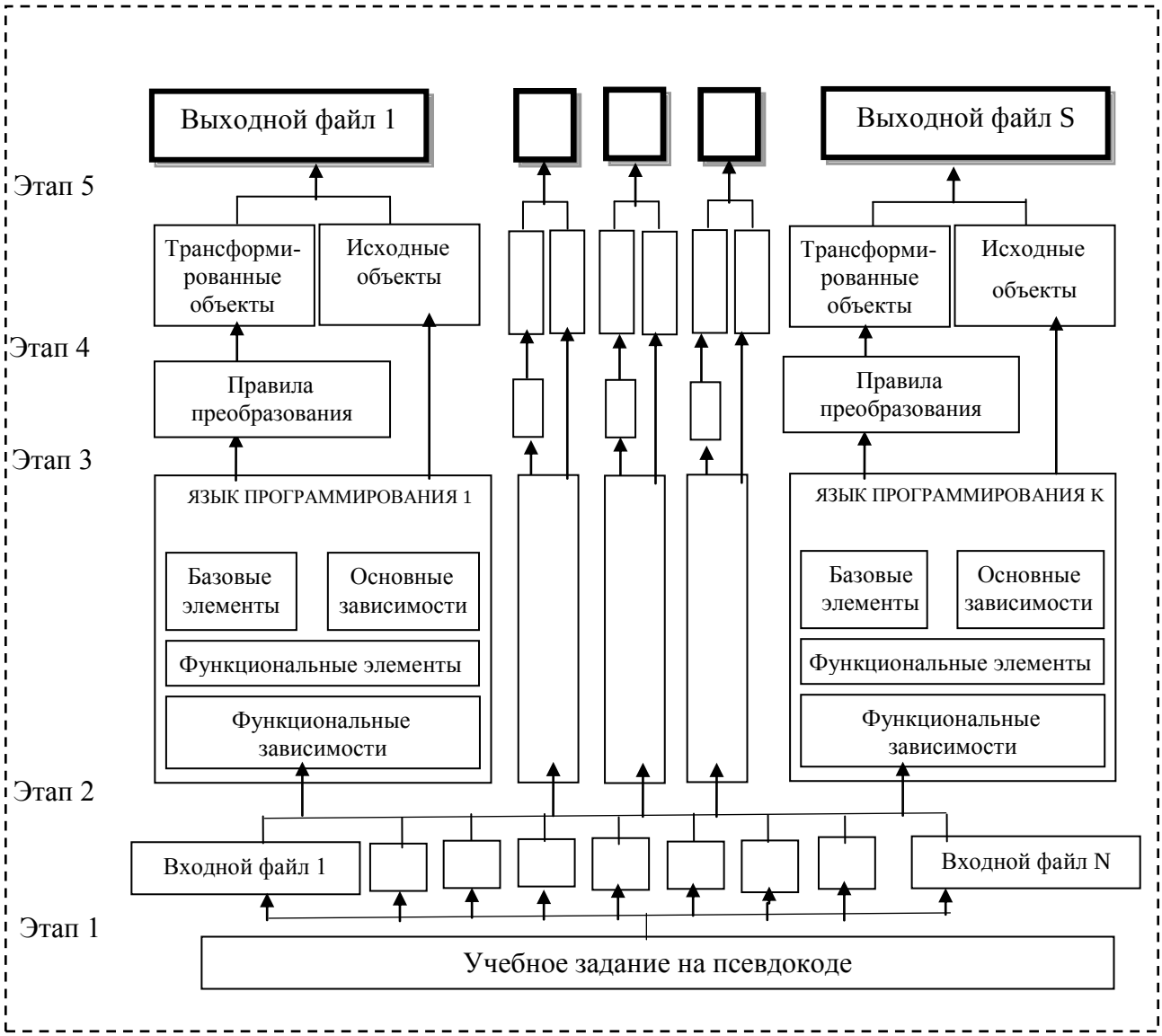

Рисунок. Схема формирования тестового задания

по программированию

Формирование тестового задания состоит из пяти этапов.

Этап 1 «Генерирование входных данных». На данном этапе алгоритмы, которые будут включены в тестовые задания, записываются на псевдокоде, и формируются входные файлы.

Этап 2 «Трансляция в язык программирования». Этот этап подразумевает перевод алгоритмов из входных файлов на выбранный язык программирования, при этом в структуре файла выделяются следующие объекты: базовые элементы, основные зависимости, функциональные элементы, функциональные зависимости.

Этап 3 «Модифицирование объектов» подразумевает искажение некоторых объектов (или их частей) по заданным правилам. Правила и их количество определяется уровнем сложности и интерактивности задания.

Этап 4 «Распределение объектов». На этом этапе базовые элементы, основные зависимости, функциональные элементы, функциональные зависимости сортируются на трансформированные и исходные.

Этап 5 «Сборка учебного проверочного задания». На этом этапе происходит объединение трансформированных и исходных объектов и формирование тестовых заданий с учётом выбранного языка программирования, уровня сложности, интерактивности, правил трансформирования объектов.

Внедрение в учебный процесс системы, содержащей модуль формирования индивидуальных лабораторных работ и модуль, генерирующий тестовые задания, способствует развитию электронного обучения, так как известно, что адекватная поддержка студентов современными ИТ побуждает интерес студентов к самостоятельной работе, к поиску лучших решений при выполнении лабораторных работ, повышает уровень освоения языка программирования.

Список литературы

1. Зорин Ю.А. Использование алгоритмов комбинаторной генерации при построении генераторов тестовых заданий // Дистанционное и виртуальное обучение, 2013, №6. – С. 54-59.

2. Лаптев В.В., Толасова В.В. Генерация вариантов заданий для лабораторных работ по программированию // Вестник Архангельского государственного университета, 2010, №1. – С. 127-131.