



## СОДЕРЖАНИЕ

Введение.....	4
1 Техническое задание на разработку системы адаптивного управления процессом выдачи учебных заданий.....	6
1.1 Техническое задание на разработку .....	6
1.2 Обзор литературных источников и интернет источников в области систем адаптивного управления процессом выдачи учебных заданий .....	7
1.3 Сравнительные характеристики программных средств, предназначенных для реализации системы адаптивного управления процессом выдачи учебных заданий .....	17
1.4 Схемы алгоритмов работы программного продукта .....	28
2 Программные средства для разработки подсистемы адаптивного управления процессом выдачи учебных заданий.....	30
2.1 Выбор программных продуктов для решения задачи разработки подсистемы адаптивного управления процессом выдачи учебных заданий .....	30
2.2 Структура программы .....	31
3 Разработка программного обеспечения системы адаптивного управления процессом выдачи учебных заданий.....	33
3.1 Процесс разработки и структура программного обеспечения .....	33
3.1.1 Разработка программы.....	33
3.1.2 Архитектура системы .....	35
3.1.3 Структура таблиц базы данных .....	37
3.1.4 Взаимодействие модулей .....	39
3.2 Разработка основных модулей программы.....	43

					<b>ДП - 230101.65 - ПЗ</b>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Монин Л.В.			Разработка подсистемы адаптивного управления процессом выдачи учебных заданий	Лит.	Лист	Листов
Провер.		Редькина А.В.					2	83
Н. Контр.		Иванов В.И.				<b>ВТ</b>		
Утв.		Легалов А.И.						



## ВВЕДЕНИЕ

В связи с развитием компьютерных технологий в настоящее время широкое распространение получили системы адаптивного обучения. Адаптивное обучение представляет собой технологическую систему форм и методов, способствующую эффективному индивидуальному обучению. Это система лучше других учитывает уровень и структуру начальной подготовленности, оперативно отслеживает результаты текущей подготовки, что позволяет рационально подбирать задания и упражнения для дальнейшего быстрого продвижения.

Адаптивное обучение – это модель, основной идеей которой является «подстройка» под способности, знания, умения и даже настроение каждого ученика. За время развития адаптивного обучения достигнуты немалые успехи. И связано это прежде всего с развитием цифровых технологий.

Компьютеризация образования позволяет уменьшить непроизводительные затраты живого труда преподавателей, сохранить методический потенциал профессоров старшего поколения, многократно использовать результаты овеществленного труда в форме компьютерных обучающих и контролирующих программ. Это может также помочь преподавателям заниматься технологией современного учебного процесса, в котором ведущая роль отводится не столько и не только обучающей деятельности преподавателя, сколько учению самих студентов и разработанности технологической платформы адаптивной системы обучения.

На сегодня адаптация является одним из наиболее развивающихся направлений в системах электронной поддержки обучения, поэтому исследования и разработка компьютерных обучающих систем является актуальными направлениями в области образования.

Вместе с тем, при всех достижениях, адаптивные системы обучения недостаточно разработаны для интернета.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4



Целью данного дипломного проекта является разработка программного обеспечения, предназначенного для использования в области интернет-технологий, подсистемы адаптивного управления процессом выдачи учебных заданий, в котором реализованы алгоритмы адаптивного управления процессом обучения.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

# 1 Техническое задание на разработку системы адаптивного управления процессом выдачи учебных заданий

## 1.1 Техническое задание на разработку

Основной целью является разработка программного обеспечения для системы адаптивного управления процессом выдачи учебных заданий (САУВУЗ), позволяющего автоматически выдавать задания на решение задач с учетом уровня подготовленности студентов. Программное обеспечение использует базу данных, где хранятся тексты задач, которые упорядочены следующим образом:

- все задачи каждой темы составляют последовательность от простого к сложному;
- задач одинаковой сложности, относящихся к одной теме, может быть несколько;

Выдача новой задачи производится системой управления только после получения правильного решения более простой задачи.

Разрабатываемый программный продукт должен:

- обеспечивать авторизацию администратора с полными правами.
- обеспечивать авторизацию преподавателя, при этом, преподаватель должен иметь права на добавление новых заданий, сортировку их по уровню сложности, просмотр и оценку выполненных заданий, и общую оценку знаний каждого студента;
- обеспечивать авторизацию студентов с правами, которые позволяют получать учебные задания, вводить результаты их выполнения, а так же просматривать полученные задания;
- содержать таблицу пользователей;
- содержать таблицу со списком заданий определенного уровня сложности по каждой теме для выдачи студентам с возможностью

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

просмотра и изменения;

- содержать таблицу со списком выполненных заданий с возможностью просмотра и ввода оценки по каждому заданию;
- содержать таблицу выданных заданий;
- содержать таблицу со списком студентов, для каждого из которых должно быть указано количество выполненных заданий, количество успешно выполненных заданий, количество невыполненных заданий, средняя оценка и общий уровень обученности студентов с возможностью просмотра и ввода уровня обученности;
- следующее задание студенту должно выдаваться только после успешного выполнения предыдущего;
- если студент получил неудовлетворительную оценку за задание, то ему выдается задание меньшего уровня сложности;
- если студент получил неудовлетворительную оценку за задание первого уровня сложности, то следующее задание ему может быть выдано только через 24 часа;
- проект должен быть реализован с применением программных средств, предназначенных для разработки программ для интернета.

## **1.2 Обзор литературных источников и интернет источников в области систем адаптивного управления процессом выдачи учебных заданий**

Вопросы адаптивного обучения рассмотрены в статье «Адаптивное обучение и адаптивный тестовый контроль» [6]. Показано, что главной причиной появления адаптивных систем индивидуального обучения стала недостаточная информированность о реальном уровне знаний учеников и естественные различия в их способностях усвоить предлагаемые знания.

Адаптивный тестовый контроль и адаптивное обучение представляют собой современный компьютерный вариант реализации известного

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		7

педагогического принципа индивидуализации обучения, одного из самых важных для подготовки квалифицированных специалистов. Этот принцип не мог быть реализован сколько-нибудь эффективным образом в условиях традиционных классно-урочных форм занятий - лекционных и групповых. В современном варианте адаптивные формы появились недавно, как результат взаимосвязи трех прогрессивных тенденций: компьютеризации образования, возрастания роли самообучения и развития теории педагогических измерений.

Одной из первых активно применять технологии анализа данных в образовании стала компания Knewton [7]. В результате была создана адаптивная образовательная платформа, которую можно подключить к любой современной системе управления учебным процессом (LMS).

Методология Knewton строится вокруг двух основных понятий: технологии планирования образовательной траектории и сложной модели оценки студента. Такой подход в корне отличается от большинства “адаптивных приложений”, которые по сути применяют адаптивный подход к единственной точке, в которой измеряются знания студентов. Примером такого “слабоадаптивного” подхода является диагностический экзамен, по результатам которого компьютер определяет какой контент будет показан студенту в дальнейшем.

В статье «Авторские средства проектирования интеллектуальных обучающих систем» [8] рассматривается средство проектирования обучающих систем МОНАП-II и математическая модель управления процессом обучения.

Адаптивное управление процессом обучения основано на стабилизации трудности учебных задач.

В статье «Организация адаптивного обучения навыкам алгоритмической природы» [9] показано, что большая часть задач обучения может быть отнесена к классу задач обучения навыкам алгоритмической природы. Их основной характеристикой является наличие соответствующего

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

алгоритмического предписания, описывающего путь решения задач из заданной предметной области.

Для указанного класса задач обучения разработана настраиваемая на конкретную предметную область подсистема модели обучения в автоматизированных обучающих системах.

Задача обучения формулируется как задача управления, в которой обучаемый является объектом управления, а автоматизированная обучающая система (АОС) - источником управления. АОС является источником адаптивного управления. Эта система на каждом шаге обучения идентифицирует уровень знаний обучаемого и на основе этой информации вырабатывает управляющие воздействия (задания), адекватные знаниям обучаемого, учитывая при этом предысторию обучения. Идентификация знаний и определение свойств заданий обеспечивается наличием в АОС подсистемы модели обучения.

Универсальная подсистема модели обучения написана на языке ассемблера. Часть программных компонент разработана с использованием макросредств ассемблера и может быть настроена на конкретные условия применения по параметрам, определяемым пользователем (разработчиком или тем, кто сопровождает АОС).

Процесс обучения представляет собой управляемый и контролируемый процесс выполнения учебных заданий обучаемым, на каждом шаге обучения в статье выделяются четыре этапа:

1. Выдача учебного задания с требуемыми свойствами;
2. Ввод ответа обучаемого, контроль правильности выполнения задания и выдача подкреплений, выдача соответствующих объяснений в случае появления ошибок;
3. Идентификация знаний обучаемого и принятие решения:
  - о продолжении обучения,
  - о достижении цели обучения,
  - об аварийном окончании обучения (в этом случае обучаемый

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

отсылается к преподавателю или к соответствующей литературе);

4. Определение необходимых данных для выдачи очередного задания при вынесении решения о необходимости продолжения обучения.

Рассматриваемая модель обучения обеспечивает полную формализацию, а на ее основе и автоматизацию третьего и четвертого этапов шага обучения.

Наличие многих тестирующих и обучающих программ не обеспечивает адаптацию к студенту [1], что значительно снижает эффективность обучения. Этот недостаток устраняется в компьютерных адаптивных системах обучения. Именно компьютерные адаптивные системы обучения в той или иной степени учитывая индивидуальные особенности учащихся, предоставляют каждому из них возможность наиболее быстро и качественно получить новые знания. Такие системы могут эффективно использоваться как для проведения дистанционного обучения, так и для интеграции в традиционный (очный) учебный процесс.

В работе Л.В. Зайцевой [2] рассмотрены уровни адаптации в обучающих системах:

- адаптация к студентам как категории пользователей;
- адаптация к группе студентов;
- адаптация к отдельному студенту.

Первый уровень адаптации предусматривает адаптацию к каждой категории пользователей компьютерной системы обучения в зависимости от их потребностей и обычно реализуется созданием специального интерфейса для каждого класса пользователей. Такой подход характерен для любых компьютерных систем. В интеллектуальных обучающих системах учащемуся необходимо предоставить следующие возможности: обучение, проверка знаний, упражнения, помощь и справочная информация, вопросы преподавателю, конференции, студенческие форумы, электронные методические пособия, ввод комментариев по ходу занятия и др.

Адаптация к группе студентов обеспечивает адаптацию в зависимости

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		10

от выбранной специальности, образовательной программы, возраста и психологической направленности личности. Этот уровень адаптации базируется, в первую очередь, на решении вопросов как и чему учить. Ответ на первый вопрос определяет цели обучения, т.е. объем необходимых знаний, умений и навыков, и степень их освоения. Решение второго вопроса («как учить ?») определяет выбор методов обучения, наиболее подходящих для группы учащихся, а также способов представления информации. На выбор методов обучения и способов представления информации влияют как возраст обучаемого, так и его психологическая направленность личности (ориентация на себя, на задачу, на взаимодействие). На третьем уровне достигается максимальная степень адаптации к учащемуся, т.к. он основан на учете личностных характеристик студента, его предшествующих и текущих знаний, умений и навыков, опыта, способностей и т.п.

Модель обучаемого [2] является информационной базой для принятия решения о педагогических воздействиях на обучаемого. Она содержит достаточно полную информацию об обучаемом: уровень его знаний, умений и навыков; способность к обучению; способность выполнения заданий (умеет ли он использовать полученную информацию); уровень мотивации и другие параметры. Модель студента динамична, т.е. изменяется в процессе прохождения курса, в ходе работы с системой. Главное ее назначение: хранение информации об обучаемом; отбор и анализ необходимых данных для последующей адаптации системы; оценка уровня знаний обучаемого.

Модель обучаемого в общем случае, является вектором  $M = \{M1, M2, M3, \dots, Mn\}$  и может включать следующие компоненты, как правило, векторы: M1 – предыстория обучения; M2 – результаты текущей работы с курсом (тип выполненных заданий, время выполнения заданий и т.д.); M3 – личностные психологические характеристики (тип и направленность личности, мотивация, репрезентативная система, способность к обучению, уровень беспокойства-тревоги, особенности памяти и др.); M4 – опыт работы с компьютерной системой; M5 – общий уровень подготовленности и др. [3].

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

Способы реализации адаптивной технологии обучения в информационных обучающих системах рассмотрены в статье «Реализация адаптивной технологии обучения в информационной обучающей системе» [4].

В статье рассматривается информационная обучающая система, поддерживающая формирование адаптивных педагогических программных средств различного типа. Основные функциональные возможности системы:

- наличие средств формирования и представления учебного материала;
- наличие средств формирования и представления успеваемости обучаемого;
- адаптация к предметной области и обучаемому.

Разработанная система была апробирована на группе студентов. Результаты показали, что разработанная система хорошо коррелируется с оценками педагога.

Процесс подготовки будущего специалиста как объект управления рассмотрен в статье «Теоретические основы управления адаптивными интеллектуальными образовательными системами» [12]. Рассмотрена обучающая система, состояние которой определяется, например, такими показателями как уровень сложности и доступности предъявляемого материала, временем, затраченным конкретным обучаемым на усвоение нового материала, сложности и значимости практических задач, оказывает воздействие  $X$  на будущего специалиста, желающего получить знания по выбранной профессии, с целью перехода в состояние  $Y$ , для дальнейшего взаимодействия с внешней рыночной средой (Рисунок - 1.1).

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12





взаимодействия между преподавателем и учениками, хотя подходит и для организации традиционных дистанционных курсов, а так же поддержки очного обучения.

Moodle переведена на десятки языков, в том числе и на русский и используется в 197 странах мира.

*Преимущества системы дистанционного обучения Moodle.*

Основным преимуществом системы дистанционного обучения Moodle является возможность ее бесплатного использования. При этом функциональность системы не уступает коммерческим аналогам.

Еще одним важным преимуществом является то, что она распространяется в открытом исходном коде, что позволяет адаптировать ее под специфику задач, которые должны быть решены с ее помощью.

Встроенные средства разработки дистанционных курсов позволяют снизить стоимость разработки учебного контента и решить проблемы совместимости разработанных дистанционных курсов с СДО.

Также к плюсам системы Moodle следует отнести легкость инсталляции, а также обновления при переходе на новые версии.

*Некоторые недостатки.*

Moodle имеет значительный недостаток: в системе не предусмотрены группы уровня сайта (далее - «учебные группы»), что делает очень сложным учет студентов разных специальностей. Группы в Moodle существуют не для управления правами доступа к курсам, а для разделения групп слушателей в одном курсе. Чтобы одни слушатели не видели активность других. Группы создаются внутри курса и не могут быть перенесены в другие.

Кроме этого, оценками слушателя можно оперировать только внутри курса. Нет возможности составить итоговую ведомость, например, по всем дисциплинам семестра, да и само понятие семестра в базовой версии системы отсутствует.

Из сказанного можно сделать вывод, что Moodle является системой, ориентированной на западную модель обучения: изучение одного курса

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

несколькими группами слушателей, в то время как для организации и управления учебным процессом отечественного ВУЗа, система дистанционного обучения должна быть ориентированной на приоритетное использование учебных групп.

2. *TrainingWare Class* - это первая российская СДО с открытым кодом. Компания "Корпоративные Системы Обучения" (ВСС Group) объявила платформу *TrainingWare Class* свободно распространяемым программным обеспечением (СПО) с открытым исходным кодом. Теперь эту систему дистанционного обучения (СДО) смогут бесплатно использовать и совершенствовать сторонние пользователи и разработчики.

*TrainingWare Class* — это технологическая платформа для автоматизации процессов обучения и аттестации пользователей. Она обеспечивает взаимодействие между преподавателем и учениками в процессе обучения, разработку курсов и тестов, поддержку очного обучения и, что особенно востребовано, автоматизированную аттестацию пользователей. *TrainingWare Class* позволяет формировать индивидуальный подход к обучению и автоматизировать рутинную работу учителя.

На сегодняшний день *TrainingWare Class* является единственной свободно распространяемой российской СДО, разработанной, в отличие от зарубежных аналогов, с учетом требований отечественного образования.

За счет масштабируемости решений на базе *TrainingWare Class* могут создаваться комплексные системы автоматизации учебных процессов и системы мониторинга обучения уровня района, города или региона, а также формироваться единые библиотеки учебно-методических материалов, создаваемые участниками педагогических сообществ и социальных сетей.

3. *Claroline LMS* - это платформа для электронного электронного обучения (eLearning) и электронной деятельности (eWorking), позволяющая учителям создавать эффективные онлайн-курсы и управлять процессом обучения и совместными действиями на основе веб-технологий.

Переведённая на 35 языков, *Claroline LMS* обладает обширным

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15



адаптирована для университетской среды, что выражается в поддержке большого количества учеников и курсов. Dokeos же больше ориентирован на профессиональную клиентуру, например, на персонал предприятия;

— организовать (скорее, выставить на продажу) набор дополнительных сервисов для платформы. Название Dokeos относится как к приложению, так и к сообществу, которое предлагает набор различных сервисов к платформе: хостинг, интегрирование контента, разработка дополнительных модулей, техническая поддержка и т.д.

Dokeos бесплатен, поскольку лицензия Claroline (GNU/GPL) предполагает, что ветки подпадают под ту же лицензию. Поскольку ветка была выделена недавно, оба приложения сейчас относительно похожи друг на друга, хотя некоторые различия в эргономике, построении интерфейса, функционале уже начинают проявляться.

### **1.3 Сравнительные характеристики программных средств, предназначенных для реализации системы адаптивного управления процессом выдачи учебных заданий**

Поскольку целью данного проекта является разработка программного обеспечения для реализации системы адаптивного управления процессом выдачи учебных заданий методами интернет-технологий, целесообразно проанализировать программные средства, предназначенные для решения подобных задач.

Приведем характеристики основных языков программирования, предназначенных в первую очередь для разработки программного обеспечения для создания интернет сайтов. Можно отметить следующее:

- Сценарные языки позволяют разработчикам сцеплять вместе различные пакеты программ, а также согласовывать полученные в результате системы.

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		17





Согласно самому красивому из них, название "perl" произошло от слова "pearl" (жемчужина).

Основной особенностью языка считаются его богатые возможности для работы с текстом, реализованные при помощи регулярных выражений (regular expressions). Перл также знаменит огромной коллекцией дополнительных модулей CPAN.

### *PHP*

PHP — интерпретируемый сценарный язык программирования, созданный для генерации HTML-страниц на веб-сервере и работы с базами данных. В области веб-программирования PHP является на сегодняшний день одним из самых распространённых технологий (наряду с Perl, ASP/.NET и Python) благодаря простоте, скорости выполнения и богатой функциональности. PHP распространяется свободно. Синтаксис языка похож на синтаксис C++. PHP поддерживается подавляющим большинством представителей сетевого хостинга.

Название "PHP" представляет собой самоповторяющуюся (рекурсивную) аббревиатуру и расшифровывается как "PHP: HypertextPreprocessor", или "PersonalHomePage". PHP был создан в качестве надстройки над Perl для облегчения разработки веб-страниц. За свою жизнь PHP значительно изменялся. Одной из сильнейших сторон PHP является возможность расширения ядра. Интерфейс написания расширений привлек к PHP множество сторонних разработчиков, работающих над своими модулями, что дало PHP возможность работать с огромным количеством баз данных, протоколов, поддерживать большое число API. PHP поддерживает ООП (деструкторы, открытые, закрытые и защищённые члены и методы, final-члены и методы, интерфейсы и клонирование объектов). PHP поддерживает XML. Является наиболее распространённым языком веб-программирования. Его используют сотни тысяч веб-программистов.

### *JavaScript*

JavaScript - это язык программирования от компании Netscape, который

										Лист
										20
Изм.	Лист	№ докум.	Подпись	Дата	ДП - 230101.65 - ПЗ					





Сценарий PHP может состоять из 10 000 строк или из одной строки - все зависит от специфики задачи. При этом не надо подгружать библиотеки и указывать специальные параметры компиляции. Механизм PHP просто начинает выполнять код после первой экранирующей последовательности (<?) и продолжает выполнение до того момента, когда он встретит парную экранирующую последовательность (?>). Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

PHP - язык, который может быть встроен непосредственно в html -код страниц, которые, в свою очередь будут корректно обрабатываться PHP - интерпретатором. Можно использовать PHP для написания CGI-сценариев и избавиться от множества неудобных операторов вывода текста. Можно непосредственно формировать HTML документы в PHP-файле.

Очень важное преимущество PHP заключается в его эффективности. PHP не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью.

По некоторым оценкам, большинство PHP-сценариев (особенно не очень больших размеров) обрабатываются быстрее аналогичных им программ, написанных на Perl и других языках. Однако, чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут работать значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных web-приложений.

Поскольку PHP является встраиваемым языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя PHP обычно рекомендуется использовать в сочетании с HTML, он с таким же успехом интегрируется и в JavaScript, WML, XML и другие языки. Кроме того, хорошо структурированные приложения PHP легко расширяются по мере необходимости.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22









- Расширения - существует возможность расширения функционала за счет сохранения своих процедур.
- Объектность - PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого

*Недостатки PostgreSQL:*

- Производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL
- Популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество.
- Хостинг - в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

Опять-таки в силу простоты и достаточной функциональности для целей проекта наиболее подходит СУБД MySQL.

Остановимся более подробно на преимуществах MySQL.

MySQL является быстрой и стабильной системой и отличным сервером баз данных. Система MySQL находится в открытом доступе и выполнена как в версии бесплатного программного обеспечения, так и в коммерческом исполнении.

MySQL поддерживает такие возможности как, транзакции, блокирование на уровне строк, внешние ключи, подзапросы и полно-текстовый поиск.

Система MySQL хорошо проверена и надежна. Она используется целым рядом очень требовательных заказчиков, таких как Yahoo!, Finance, Slashdot и Бюро переписи США.

По сравнению с конкурентами MySQL выигрывает в производительности и невысоким требованиям к дисковому пространству.

					ДП - 230101.65 - ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

## 1.4 Схемы алгоритмов работы программного продукта

Структурная схема работы программного обеспечения для создания интернет сайта представлена на рисунке 1.2

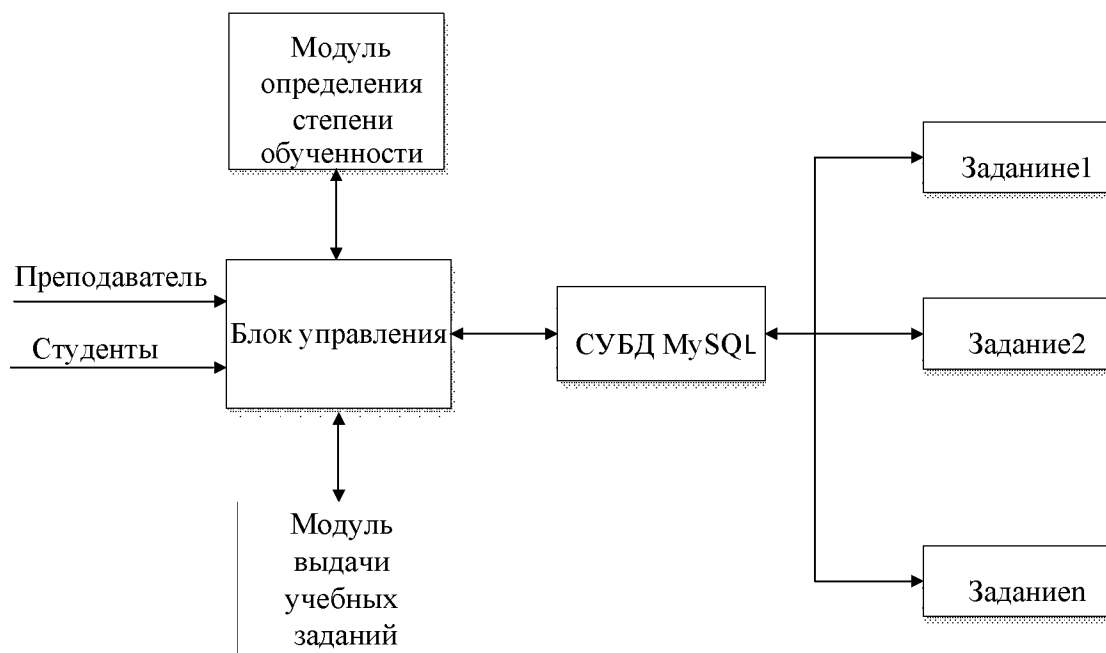


Рисунок 1.2 - Структурная схема работы системы адаптивного управления процессом выдачи учебных заданий

Учебные задания хранятся в СУБД MySQL. При этом используется несколько таблиц базы данных. Каждая таблица предназначена для хранения определенного набора данных. Все задачи в таблице заданий отсортированы по принципу «от простого к сложному». Задач одинаковой сложности, может быть несколько.

Управление всем процессом работы производится в блоке управления. Выдача новой задачи производится подсистемой только после получения правильного решения более простой задачи.

Обученность студента устанавливается в модуле определения степени обученности.

Преподаватель имеет возможность добавлять новые учебные задания,



выставлять оценки за задачи и оценивать уровень знаний студента.

Каждый из указанных модулей (блоков) содержит несколько программных модулей, каждый из которых решает свою задачу в рамках алгоритма работы системы.

Эскизная блок-схема алгоритма работы клиентской части программы приведена в приложении А.

После авторизации определяется тип пользователя, который вошел в систему. Если это системный администратор, то, обладая полными правами, он может выполнить любую операцию в системе, корректировать таблицы и задания.

Если это студент, он может получить учебное задание для выполнения, при этом полученное задание заносится в таблицу выданных заданий и в таблицу выполненных заданий. После выполнения учебного задания студент с помощью специальной формы вводит ответ на задание, который записывается в соответствующее поле таблицы выданных заданий и в соответствующее поле таблицы выполненных заданий.

Преподаватель имеет возможность просмотреть таблицу выполненных заданий и выставить оценки по каждому заданию, которые также записываются в таблицу выданных заданий для оцениваемого студента. Он может добавлять в таблицу заданий новые задания с помощью формы ввода и выставлять общий уровень знаний студентов в таблице оценки знаний студентов.

На этапе оценки знаний студента преподаватель определяет общий уровень обученности студента, например «низкий», «средний», «достаточный», «высокий» на основании среднего балла за все задания. Этот уровень заносится в таблицу оценки знаний студентов через специальную форму.

В таблице оценки знаний студентов хранятся количество выполненных каждым студентом заданий, количество выполненных и не выполненных заданий, средняя оценка по заданиям и общий уровень обученности.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

Каждый студент может получить новое учебное задание при условии успешного выполнения предыдущего задания, например, если оценка за него «удовлетворительно» или выше. При этом, при успешном выполнении предыдущего задания ему выдается задание следующего уровня сложности. Если оценка за предыдущее задание неудовлетворительная, следующее задание выдается с меньшим уровнем сложности. В том случае, если это было задание с первым уровнем сложности, следующее задание может быть получено только через 24 часа.

## **2 Программные средства для разработки подсистемы адаптивного управления процессом выдачи учебных заданий**

### **2.1 Выбор программных продуктов для решения задачи разработки подсистемы адаптивного управления процессом выдачи учебных заданий**

Сравнительный анализ языков веб-программирования, проведенный в гл. 1 показал, что наиболее приемлемым для разработки является сценарный язык PHP. Он широко распространен, находится в свободном доступе, несложен в освоении и достаточно функционален. Более подробно преимущества PHP рассмотрены в разделе 1.3.

Проект будет выполняться на локальном компьютере. При необходимости, всегда можно перенести модули на какой-либо хостинг в интернете.

Для того, чтобы начать разработку программного обеспечения на локальном компьютере, на этом компьютере был установлен локальный веб-сервер Apache, а также программные продукты PHP, MySQL, PHPExpertEditor.

После установки необходимых программных средств, была выполнена разработка программного обеспечения системы адаптивного управления

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		30

процессом выдачи учебных заданий.

## 2.2 Структура программы

Структурная схема связей между модулями программы на рисунке 2.1

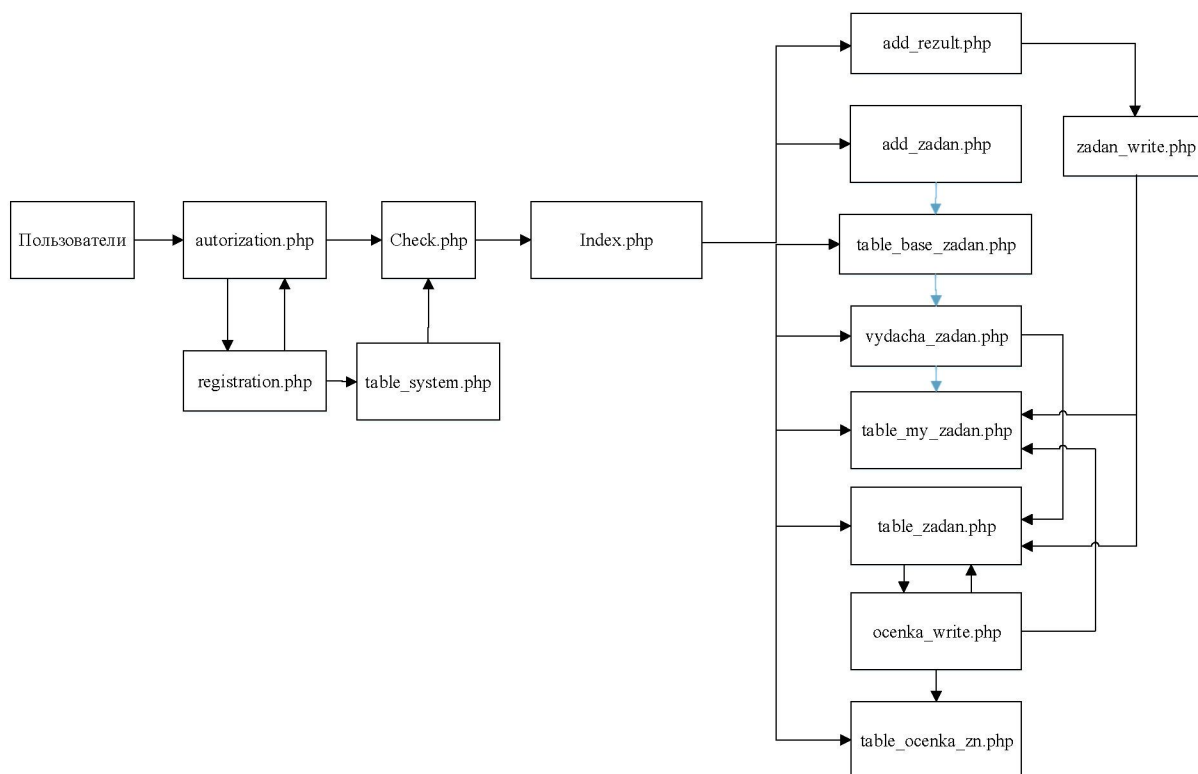


Рисунок 2.1 - диаграмма модулей программы

Основные модули программы:

authorization.php – авторизация пользователей;

registration.php – регистрация новых пользователей;

Check.php – проверка логина и пароля пользователя;

table\_system.php – просмотр и редактирование таблицы пользователей;

Index.php – главный модуль;

table\_base\_zadan.php – просмотр и редактирование таблицы заданий;

vydacha\_zadan.php – выдача задания студенту;

zadan\_write.php – запись данных задания в таблицу выданных заданий

и в таблицу выполненных заданий;

table\_my\_zadan.php – просмотр и редактирование таблицы выданных заданий;

add\_rezult.php – ввод результата выполнения задания;

add\_zadan.php – добавление нового задания;

table\_zadan.php – просмотр и редактирование таблицы выполненных заданий;

ocenka\_write.php – занесение данных о результатах выполнения заданий студентом в таблицу оценки знаний и в таблицу выданных заданий;

table\_ocenka\_zn.php – просмотр и редактирование таблицы оценки знаний студентов.

После авторизации и проверки логина и пароля пользователь попадает на главную страницу сайта (Index.php). Если авторизовался преподаватель, он может добавлять новые задания в базу (add\_zadan.php), просматривать выполненные задания и выставять оценки за каждое задание (table\_zadan.php). Он также может оценивать общий уровень знаний студентов (table\_ocenka\_zn.php) и редактировать таблицу заданий (table\_base\_zadan.php).

Если вошел студент, он получает задание определенного уровня сложности (vydacha\_zadan.php). Он может посмотреть свои задания (table\_my\_zadan.php). После выполнения задания, студент вводит ответ на задание (add\_rezult.php), при этом данные ответа записываются в таблицу полученных заданий (table\_my\_zadan.php) и в таблицу выполненных заданий (table\_zadan.php), при этом для записи используется модуль (zadan\_write.php).

Администратор входит в систему под логином «admin» и системным паролем. Он может редактировать любую базу данных и выполнять роль любого пользователя.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

### **3 Разработка программного обеспечения системы адаптивного управления процессом выдачи учебных заданий**

#### **3.1 Процесс разработки и структура программного обеспечения**

##### **3.1.1 Разработка программы**

Разработка программного обеспечения выполнялась исходя из требований технического задания и выбранных для разработки программных продуктов.

Для того, чтобы реализовать требуемый алгоритм работы, необходимо было разработать совокупность модулей, связать их между собой, а также создать несколько таблиц базы данных MySQL.

Хотя язык PHP и предполагает написание текста программ в редакторе «Блокнот» при разработке, был использован редактор PHPExpertEditor, который имеет внутренний HTTP-сервер и отладчик, для того, чтобы выполнять, проверять и отлаживать сценарии PHP, производить проверку синтаксиса PHP, используя внутренний браузер, эксплорер кода, файловый эксплорер, FTP-клиент, эксплорер проектов и шаблоны кода. Кроме того есть возможность запускать модули в любом из браузеров, установленных на компьютере. Пример работы с программным кодом приведен на рисунке 3.1.

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		33



```

<html>
<head>
<title></title>
</head>
<body>
<?php
echo "Старт";
require_once "mysql_connect_doska.php";
echo "<br />Начали";
//\$sql="DROP TABLE ann_kras";
$sql="DROP TABLE vupoln";
mysql_query($sql);
// Создаем таблицу с определенным набором полей
$sql="CREATE TABLE vupoln (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
nomer int(2) NOT NULL, zadanie text NOT NULL, ur int(2) NOT NULL default '1' , otvet_prav
text NOT NULL,
student char(20) NOT NULL, otvet_stud text NOT NULL, oценка_prep char(10) NOT NULL,
UNIQUE KEY `id` (`id`))";
mysql_query($sql);
echo "<br />Вставили";
?>
</body>
</html>

```

Рисунок 3.3 - Создание таблицы vupoln

Отладка каждого модуля производилась путем запуска модуля в браузере Google Chrome из редактора PHPExpertEditor.

### 3.1.2 Архитектура системы

Архитектура системы представлена на рисунке 3.4.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35







- my\_zadan – таблица выданных заданий;
- vupoln – таблица выполненных заданий;
- ocenka\_zn – оценка знаний студентов.

Таблица пользователей «rolz» имеет следующие поля:

fio – фамилия, имя отчество пользователя, тип varchar(50) ;

grupp – группа, тип varchar(20);

pass – пароль, тип varchar(10);

login – логин, тип varchar(15).

Структура таблицы, в которой хранятся задания «base\_zadan» следующая:

tema– тема задания, тип varchar(20);

zadanie – текст задания, тип text;

ur – уровень сложности задания, тип int(2);

otvet\_prav – правильный ответ, тип text;

Номер задания определяется параметром «id».

Таблица выданных заданий «my\_zadan» имеет следующие поля:

student – логин студента (совпадает с фамилией), тип varchar(20);

nomer – номер задания, тип int(2);

tema – тема задания, тип varchar(20);

zadanie – текст задания, тип text;

ur – уровень сложности задания, тип int(2);

otvet\_stud – ответ студента, тип text;

ocenka\_prep – оценка преподавателя, тип varchar(10);

time – отметка времени, тип int(11);

Таблица «vupoln» предназначена для хранения информации о выполнении заданий студентами. При выдаче задания заполняются поля student, nomer, tema, zadanie. Таблица имеет поля:

nomer – номер задания, тип int(2);

tema – тема задания, тип varchar(20);

zadanie – текст задания, тип text;

					ДП - 230101.65 - ПЗ	Лист
						38
Изм.	Лист	№ докум.	Подпись	Дата		



Такая связь реализована, например, в главном модуле «Index.php» (Рисунок 3.7):

```
&#9679;&nbsp;&nbsp;&nbsp;<a  
href="vydacha_zadan.php?login=<?=$login?>&tema=<?=$tema?>">Получить  
задание</a></font>
```

Рисунок 3.7 - Связь между модулями по ссылке

В этом фрагменте по ссылке «Получить задание» из главного модуля «Index.php» запускается модуль «vydacha\_zadan.php». При этом в модуль «vydacha\_zadan.php» передаются переменные \$login и \$tema. Таким же образом организованы связи на других страницах системы.

2. Связь через форму ввода можно продемонстрировать на примере формы ввода задания из модуля «add\_zadan.php» (Рисунок 3.8):

```
<div style=" margin-left: 270px ">  
<fontface="Arial" size="4">  
Добавление нового задания  
</font>  
</div>  
//-----Форма ввода задания, обработчик zadan_write.php-----  
<form action="zadan_write.php" method="post">  
<?echo "<input type=hidden name='login' value='".$login.">";?>  
<?echo "<input type=hidden name=them value='".$tema.">";?>  
<div style=" margin-left: 35px ">&nbsp;&nbsp;&nbsp;</div><font face="Arial" size="2">  
    <font face="Arial">  
<br />  
<div style=" margin-left: 135px" >  
<font color="#000080">Тема</font></div>  
<div style="margin-top: -15px; margin-left: -10px" >  
<!--<input type="INT" name="them" id="them" size="20" >-->  
<?php  
//-----Подключаемся к базе-----  
require_once "mysql_connect_doska.php";  
//-----Выпадающее меню-----  
require_once "menu_tema_add.php";  
?></div>  
<div style=" margin-left: 80px; margin-top: 0px;" >  
<fontcolor="#000080">Уровень задания</font></div>  
<div style="margin-top: -22px; margin-left: 190px" >  
<input type="INT" name="ur" id="theme" size="2" ></div>
```

										Лист
										40
Изм.	Лист	№ докум.	Подпись	Дата	ДП - 230101.65 - ПЗ					

```

<br />
<div style=" margin-left: 90px; margin-top: -0px; " ><font face="Arial" size="2">
    <fontcolor="#000080">Текст задания</font></div>
<div style="margin-top: -12px; margin-left: 190px" >
<textarea name="zadan" rows=10 cols=50 ></textarea></div>
<br />
<div style=" margin-left: 70px " ><font face="Arial" size="2">
    <fontcolor="#000080">Правильный ответ</font></div>
<div style="margin-top: -15px; margin-left: 190px" >
<textarea name="otvet_zadan" rows=5 cols=50 ></textarea></div>
<br />
</p>
<div align="left" style="margin-left: 270px">
//-----Кнопка ввода-----
<input type="submit" value="Добавить задание" /></div>
</font></font></form>

```

Рисунок 3.8 - Код формы ввода нового задания

Для этой формы задан обработчик «zadan\_write.php», который запускается при нажатии на клавишу «Добавить задание» и записывает введенные данные в таблицу «base\_zadan». В модуль «zadan\_write.php» из формы передаются параметры them, ur, zadan, otvet\_zadan, содержащие введенные в форму данные.

В модуле «zadan\_write.php» эти переменные получают путем выполнения следующих операторов (Рисунок 3.9):

```

$tema = $_POST['them'];
$ur = $_POST['ur'];
$zadan = $_POST['zadan'];
$otvet_zadan = $_POST['otvet_zadan'];

```

Рисунок 3.9 - Получение переменных, содержащих данные, введенные в форму

3. Связь через вложенность модулей можно продемонстрировать на примере модуля «Check.php», фрагмент кода которого приведен на рисунке 3.10.



## 3.2 Разработка основных модулей программы

### 3.2.1 Подсистемы и авторизация пользователей

В системе адаптивного управления процессом выдачи учебных заданий (САУВУЗ) предусмотрены три типа пользователей:

1. Студент;
2. Преподаватель;
3. Администратор.

Каждый из пользователей обладает определенным набором прав, которые приведены в таблице 1.

Таблица 1 - права пользователей

Пользователь	Пункты меню на главной странице пользователя	Доступные для редактирования таблицы базы данных
Администратор	Все	Все
Преподаватель	«Добавить новое задание», «Таблица заданий», «Просмотр выполненных заданий», «Оценка знаний студентов»,	Поле <code>ocenka_prep</code> таблицы « <code>vypoln</code> » Все поля таблиц « <code>ocenka_zn</code> », « <code>base_zadan</code> »
Студент	«Получить задание», «Мои задания», «Ввести результаты выполнения»	Нет

Разработанное программное обеспечение представляет собой совокупность трех подсистем: «Преподаватель», «Студент», «Администратор».

Каждая из подсистем предназначена для решения задач в зависимости от типа пользователя. Подсистемы взаимосвязаны между собой и используют общие таблицы информационной базы.

После запуска программы пользователь попадает на страницу авторизации (Рисунок 3.11).

СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ  
ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ  
ЗАДАНИЙ

Авторизация

Логин: Преподаватель

Пароль: .....

OK

Регистрация

Рисунок 3.11 - Авторизация пользователей

По ссылке «Регистрация» открывается форма для регистрации нового пользователя (Рисунок 3.2), которая вызывается из модуля registration.php. Данные, введенные в форму регистрации записываются в таблицу «polz».

Регистрация пользователя

Фамилия Имя Отчество

Группа

Логин

Пароль

Зарегистрироваться

Рисунок 3.12 - Регистрация пользователя

Проверка введенных логина и пароля производится в модуле Check.php. При этом, если введенные логин и пароль найдены в таблице «polz», пользователь авторизуется, в противном случае выдается сообщение «Неверный логин и пароль».



### 3.2.2 Главная страница и интерфейс пользователя

В САУВУЗ реализован адаптивный интерфейс пользователя. В зависимости от того, кто авторизовался в системе интерфейс главной страницы может принимать три различных вида:

1. Интерфейс для студента (Рисунок 3.13);
2. Интерфейс для преподавателя (Рисунок 3.14);
3. Интерфейс для администратора (Рисунок 3.15).



**СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ ЗАДАНИЙ**

**Иванов Сергей Петрович**

[Авторизация](#)

Тема

Студент может получить задание и ввести результат его выполнения. Новое задание выдается только после успешного выполнения предыдущего задания.

тел. 8-928-122-87-21 E-mail: [monin@mail.ru](mailto:monin@mail.ru)

- [Получить задание](#)
- [Мои задания](#)
- [Ввести результаты выполнения](#)

Рисунок 3.13 - Интерфейс для студента

									Лист
									45
Изм.	Лист	№ докум.	Подпись	Дата					

ДП - 230101.65 - ПЗ



**СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ ЗАДАНИЙ**

**Преподаватель**

Авторизация      Студент       Тема

Преподаватель добавляет задания в информационную базу. Он может просмотреть выполненные задания и выставить оценки по каждому заданию, а также оценить уровень знаний студента.

тел. 8-928-122-87-21    E-mail: monin@mail.ru

- [Добавить новое задание](#)
- [Таблица заданий](#)
- [Просмотр выполненных заданий](#)
- [Оценка знаний студентов](#)

Рисунок 3.14 - Интерфейс для преподавателя



**СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ ЗАДАНИЙ**

**Администратор**

Авторизация

Студент может получить задание и ввести результат его выполнения. Новое задание выдается только после успешного выполнения предыдущего задания.

Преподаватель добавляет задания в информационную базу. Он может просмотреть выполненные задания и выставить оценки по каждому заданию, а также оценить уровень знаний студента.

Администратор имеет доступ ко всем таблицам базы данных, может изменять любую из них, а также контролировать алгоритм работы системы. Он может выполнять любую роль

тел. 8-928-122-87-21    E-mail: monin@mail.ru

- [Получить задание](#)
- [Мои задания](#)
- [Ввести результаты выполнения](#)
- [Добавить новое задание](#)
- [Таблица заданий](#)
- [Просмотр выполненных заданий](#)
- [Оценка знаний студентов](#)
- [Редактирование таблицы пользователей](#)
- [Редактирование выполненных заданий](#)
- [Редактирование таблицы выданных заданий](#)
- [Редактирование таблицы оценки знаний](#)

Рисунок 3.15 - Интерфейс для администратора

Формирование вида главной страницы производится в модуле Index.php.

Для всех интерфейсов имеется ссылка «Авторизация». Остальные ссылки формируются в зависимости от типа пользователя.

На главной странице студента имеются ссылки:

- Получить задание;
- Мои задания;
- Ввести результаты выполнения.

В интерфейсе имеется форма с выпадающим меню «Тема» для выбора темы заданий.

Для преподавателя доступны ссылки:

- Добавить новое задание;
- Таблица заданий;
- Просмотр выполненных заданий;
- Оценка знаний студентов.

Выпадающие меню «Студент» и «Тема» предназначены для выбора студента и темы заданий.

На странице администратора выводятся перечисленные выше ссылки и добавляются новые пункты:

- Редактирование таблицы пользователей;
- Редактирование выполненных заданий;
- Редактирование таблицы выданных заданий;
- Редактирование таблицы оценки знаний.

### 3.2.3 Добавление заданий в информационную базу

Задания для студентов хранятся в таблице базы данных «base\_zadan».

Добавлять новые задания в таблицу «base\_zadan» может только преподаватель или администратор. Для добавления разработана специальная форма (Рисунок 3.16), которая вызывается из модуля add\_zadan.php.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

Добавление нового задания

Тема

Уровень задания

Текст задания

Правильный ответ

Рисунок 3.16 - Добавление нового задания

В форме указывается уровень сложности задания («Уровень задания»), текст задания («Текст задания») и правильный ответ («Правильный ответ»). Тема выбирается из выпадающего меню. Если тема задана на главной странице пользователя, то она автоматически подставляется в форму.

В САУВУЗ реализована возможность пакетного ввода заданий, поэтому после нажатия кнопки «Добавить задание», задание добавляется в таблицу «base\_zadan» и пользователь попадает на страницу (Рисунок 3.17), где может выбрать – продолжить ли ему ввод заданий или вернуться на главную страницу. Для записи задания предназначен модуль zadan\_write.php

**Задание добавлено**

Рисунок 3.7 - Страница записи задания

					<b>ДП - 230101.65 - ПЗ</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		48





Тема География  
Задание № 19 для Иванов  
Куда впадает река Волга ?

Рисунок 3.19 - Получение задания студентом

В модуле `vydacha_zadan.php` реализован следующий алгоритм выдачи задания:

- Задания выбираются случайным образом для определенного уровня сложности;
- Уровень сложности зависит от результатов выполнения предыдущих заданий студентом;
- Если студент получает задание впервые, он получает задание первого уровня;
- Если студент успешно выполнил предыдущее задание, он получает задание с более высоким уровнем сложности (уровень увеличивается на единицу);
- Если студент за предыдущее задание получил оценку «неуд», ему выдается задание с меньшим уровнем сложности, (уровень уменьшается на единицу);
- Если студент неудачно выполнил первое задание, то получить новое он сможет только через 24 часа;
- Если студент не ввел ответ на полученное задание, новое задание ему не выдается.

После выдачи задания, его параметры записываются в таблицы `my_zadan` и `vypoln`.

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		50

### 3.2.5 Ввод результатов выполнения задания

Для ввода ответа на задание студент на своей странице выбирает тему из выпадающего меню и переходит по ссылке «Ввести результат выполнения» (Рисунок 3.20).

[На главную страницу](#)

Ввод результатов выполнения задания

Номер задания	<input type="text" value="21"/>
Тема	<input type="text" value="География"/>
Текст задания	<input type="text" value="Крупнейшее течение в Атлантическом океане"/>
Ответ на задание	<input type="text" value="Гольфстрим"/>

Рисунок 3.20 - Ввод результатов выполнения задания

Задание для ввода выбирается в модуле «Index.php» из списка выданных заданий студенту по заданной теме. Параметры задания передаются в форму ввода (модуль «add\_rezult.php»). При этом автоматически заполняются поля «Номер задания», «Тема» и «Текст задания».

После нажатия кнопки «Добавить результат в базу данных» ответ студента заносится в поле `otvet_stud` таблицы «`vypoln`» и в поле `otvet_stud` таблицы «`my_zadan`».

### 3.2.6 Выставление оценок за задания

Преподаватель проверяет правильность выполнения заданий и выставляет оценки по каждому заданию перейдя по ссылке «Просмотр

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

выполненных заданий» на своей странице. При этом, можно задать отбор по студенту и теме в меню «Студент» и «Тема». Для целей просмотра и ввода оценки используется модуль table\_zadan.php.

После запуска table\_zadan.php на экране отображается таблица выполненных заданий студентов «vuroIn» показанная на рисунке 3.21 без отбора.

[На главную страницу](#)

id	nomer	tema	zadanie	ur	otvetprav	student	otvetstud	ocenka_prep	
51	19	География	Куда впадает река Волга ?	1	В Каспийское море	Иванов	В море	хор	Изменить
Удалить									
58	3	Экономика	Затраты предприятия ?	1	Себестоимость	Сидоров	Себестоимость	отл	Изменить
Удалить									
39	3	Экономика	Затраты предприятия ?	1	Себестоимость	Иванов	Зарплата	неуд	Изменить
Удалить									
40	25	Математика	10+79	1	89	Иванов	89	отл	Изменить
Удалить									
43	19	География	Куда впадает река Волга ?	1	В Каспийское море	Петров	В море	уд	Изменить
Удалить									
45	21	География	Крупнейшее течение в Атлантическом океане	1	Гольфстрим	Петров	Гольфстрим		Изменить
Удалить									
46	3	Экономика	Затраты предприятия ?	1	Себестоимость	Петров	Себестоимость	хор	Изменить
Удалить									
47	25	Математика	10+79	1	89	Петров	89	отл	Изменить
Удалить									
49	19	География	Куда впадает река Волга ?	1	В Каспийское море	Сидоров	В Атлантический океан	уд	Изменить
Удалить									

Рисунок 3.21 - Таблица выполненных заданий студентов

Оценка за задание вводится в поле ocenka\_prep. Предусмотрены следующие оценки:

- отл – отлично;
- хор – хорошо;
- уд – удовлетворительно;
- неуд – неудовлетворительно.

При нажатии на кнопку «Изменить» строка таблицы записывается в



базу данных. При этом оценка записывается в поле `ocenka_prep` таблиц «`vuroln`» и «`my_zadan`» и запускается обработка данных студента, реализованная в модуле «`ocenka_write.php`». При этом анализируются данные выполнения заданий студентом из таблицы «`vuroln`», подсчитывается общее количество выполненных заданий, количество заданий, выполненных с положительными и отрицательными оценками и средняя оценка за задания.

При расчете средней оценки принимаются следующие соответствия:

- «неуд» – 2;
- «уд» – 3;
- «хор» – 4;
- «отл» – 5.

Рассчитанные данные записываются в поля `kol_z`, `ud`, `neud` и `ocenka_sred` таблицы «`ocenka_zn`».

### 3.2.7 Оценка уровня обученности студентов

Для обобщенной оценки знаний студентов в базе данных создана таблица `ocenka_zn`.

Записи в таблицу `ocenka_zn` добавляются при регистрации нового пользователя.

Для просмотра и редактирования таблицы `ocenka_zn` используется модуль `table_ocenka_zn.php`.

Преподаватель перейдя по ссылке «Оценка знаний студентов» раздела «Преподавателю» главной страницы системы получает возможность выставить уровень обученности каждого студента (Рисунок 3.22).

Для просмотра и редактирования таблицы `ocenka_zn` используется модуль `table_ocenka_zn.php`.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53







```
// Подключаемся к СУБД MySQL.
mysql_connect("localhost", $user, $pass)
or die("Could not connect: ".mysql_error());
@mysql_query('CREATE DATABASE $db');
mysql_select_db($db)
or die("Could not select database: ".mysql_error());
?>
```

Вспомогательный модуль `otvet_write.php` предназначен для записи ответа на задание студента в таблицы «`vypoln`» и «`my_zadan`».

При успешной записи ответа студента выдается сообщение (Рисунок 3.25).

**Результат выполнения  
добавлен в базу данных**

Рисунок 3.25 - Сообщение об успешной записи ответа

Аналогичным образом выдается сообщение о вводе неверного логина и пароля (`error.php`).

### **3.3.2 Тестирование алгоритма выполнения заданий и оценки знаний студентов**

Для проверки правильности работы программного обеспечения было проведено тестирование алгоритма выполнения заданий и оценки знаний студентов.

С помощью модуля `add_zadan.php` в базу данных было добавлено три новых задания (пользователь «Преподаватель») с уровнем сложности 1 :

					<b>ДП - 230101.65 - ПЗ</b>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		57





По теме «Прочее» (Рисунок 3.27):

[На главную страницу](#)

id	tema	zadanie	ur	otvet_prav	
43	Прочее	Идут рядом два человека, один из них - отец сына другого. Как такое может быть?	1	Это отец и мать ребенка	Изменить
<input type="button" value="Удалить"/>					
44	Прочее	Меня колесо своей машины, человек уронил все четыре гайки его крепления в решетку канализационного	2	Мальчик предложил отвернуть по одной гайке с каждого из трех колес и	Изменить
<input type="button" value="Удалить"/>					
47	Прочее	Лист бумаги прямоугольной формы перегнули пополам шесть раз. В средней части этого сложенного листа	3	128 отверстий;	Изменить
<input type="button" value="Удалить"/>					

Рисунок 3.27 - Таблица заданий студентов по теме «Прочее»

По теме «Экономика» (Рисунок 3.28):

[На главную страницу](#)

id	tema	zadanie	ur	otvet_prav	
2	Экономика	Показатель эффективности предприятия	1	Рентабельность	Изменить
<input type="button" value="Удалить"/>					
3	Экономика	Затраты предприятия ?	1	Себестоимость	Изменить
<input type="button" value="Удалить"/>					
23	Экономика	Затраты предприятия на выпуск продукции	1	Себестоимость	Изменить
<input type="button" value="Удалить"/>					
46	Экономика	Рентабельность. Что это такое ?	1	Рентабельность. Что это такое ?	Изменить
<input type="button" value="Удалить"/>					
28	Экономика	Поступление товара. Бухгалтерская проводка ?	2	Д41 К60	Изменить
<input type="button" value="Удалить"/>					
29	Экономика	Экономика должна быть экономной ?	2	Да	Изменить
<input type="button" value="Удалить"/>					
30	Экономика	УСН. Платится ли НДС ?	2	Нет	Изменить
<input type="button" value="Удалить"/>					

Рисунок 3.28 - Таблица заданий студентов по теме «Экономика»

Студент Козлов, войдя в систему впервые, получил задание по математике 1 уровня (Рисунок 3.29) и ввел ответ на него (Рисунок 3.30).



Тема Математика

Задание № 42 для Козлов

Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

Рисунок 3.29 - Получение задания студентом Козловым

**Ввод результатов выполнения задания**

Номер задания	<input type="text" value="42"/>
Тема	<input type="text" value="Математика"/>
Текст задания	<div style="border: 1px solid gray; padding: 5px;">Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.</div>
Ответ на задание	<div style="border: 1px solid gray; padding: 5px;">Не знаю</div>

Рисунок 3.30 - Ввод ответа на задание студентом Козловым

Преподаватель за это задание поставил оценку «неуд» (Рисунок 3.31).

id	nomer	tema	zadanie	ur	otvetprav	student	otvetstud	ocenka_prep	
<input type="text" value="53"/>	<input type="text" value="42"/>	<input type="text" value="Математика"/>	<div style="border: 1px solid gray; padding: 2px;">Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.</div>	<input type="text" value="1"/>	<div style="border: 1px solid gray; padding: 2px;">"сто" - 100; "миллион" - 1000000</div>	<input type="text" value="Козлов"/>	<div style="border: 1px solid gray; padding: 2px;">Не знаю</div>	<input type="text" value="неуд"/>	<input type="button" value="Изменить"/>
<input type="button" value="Удалить"/>									

Рисунок 3.31 - Выставление оценки Козлову

При попытке получить новое задание по теме «Математика» студент Козлов получает сообщение (Рисунок 3.32).

**Вы неправильно выполнили задание. Получить новое сможете 30 Май 2016 18:20:36**

Рисунок 3.32 - Попытка Козлова получить новое задание

Студент Смирнов первоначально получил задание 1 уровня по теме «Прочее» (Рисунок 3.33).

[На главную страницу](#)

Тема Прочее

Задание № 43 для Смирнов

Идут рядом два человека, один из них - отец сына другого. Как такое может быть?

Рисунок 3.33 - Получение задания Смирновым

После ввода ответа «*Это отец и мать ребенка*» он получил от преподавателя оценку «отл». Следующее задание по теме «Прочее» он получает с уровнем сложности 2. (Рисунок 3.34).

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

Тема Прочее Уровень 2

Задание № 44 для Смирнов

Меняя колесо своей машины, человек уронил все четыре гайки его крепления в решетку канализационного стока, откуда достать их было невозможно. Он уже решил, что застрял здесь, но проходивший мимо мальчик подсказал ему очень дельную мысль, которая позволила ему поехать дальше. В чем состояла его идея?

Рисунок 3.34 - Получение следующего задания Смирновым

По этому заданию Смирнов получил оценку «неуд».

Следующее задание он получает с уровнем сложности, уменьшенным на 1(Рисунок 3.35).

Тема Прочее Уровень 1

Задание № 48 для Смирнов

Кто открыл Америку ?

Рисунок 3.35 - Получение задания Смирновым меньшего уровня

Задания, выданные студенту Смирнову можно посмотреть на странице студента, установив отбор по теме «Прочее» (Рисунок 3.36).

id	student	nomer	tema	zadanie	ur	otvet_stud	ocenka_prep	time
26	Смирнов	43	Прочее	Идут рядом два человека, один из них - отец сына другого. Как такое может быть?	1	Это отец и мать ребенка	отл	1464523441
35	Смирнов	48	Прочее	Кто открыл Америку ?	1			1464543249
28	Смирнов	44	Прочее	Меняя колесо своей машины, человек уронил все четыре гайки его крепления в решетку канализационного	2	Открутить все гайки	неуд	1464525985

Рисунок 3.36 - Задания, выданные Смирнову



## ЗАКЛЮЧЕНИЕ

При выполнении дипломного проекта были рассмотрены вопросы разработки программного обеспечения для создания системы адаптивного управления процессом выдачи учебных заданий студентам, в котором реализованы алгоритмы адаптивного управления процессом обучения. Программное обеспечение предназначено для использования в области интернет-технологий.

Обзор литературных и интернет источников в области систем адаптивного управления процессом выдачи учебных заданий показал, что имеется достаточное количество публикаций и разработок в области адаптивного управления обучением. Вместе с тем эта тема недостаточно разработана в области интернет-технологий.

В работе проведен обзор и анализ существующих программных средств в области интернет-технологий для решения задач разработки систем адаптивного управления обучением. Показано, что оптимальным решением для разработки программного обеспечения является выбор основным языком программирования сценарного языка PHP и использование системы управления базами данных MySQL.

Перед началом разработки была составлена эскизная схема алгоритма работы клиентской части программы и структура программы с указанием основных модулей.

При создании программного обеспечения были разработаны основные модули программы, которые позволяют реализовать следующие функции:

- Авторизация пользователей;
- Добавление заданий в информационную базу;
- Выдача более сложного задания студенту при условии успешного выполнения предыдущего задания;
- Выдача задания с меньшим уровнем сложности студенту если

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

предыдущее задание было выполнено неудачно.

- Получение задания через 24 часа, если за первое задание получена оценка «неуд»;
- Ввод результата выполнения задания;
- Выставление оценок за задания;
- Оценка уровня обученности студентов;
- Редактирование таблиц базы данных.

Созданы уникальные интерфейсы для пользователей типа «Студент», «Преподаватель» и «Администратор».

Разработана и реализована система прав для пользователей «Студент», «Преподаватель», «Администратор» и ряд вспомогательных модулей, необходимых для правильной работы системы.

При тестировании программного обеспечения было проверено насколько правильно реализованы поставленные задачи в алгоритме работы.

Однако в данном проекте есть важный недостаток - слабая адаптивность. Все темы доступны студентам сразу на выбор, а преподаватель не может настраивать порядок изучения материала. В будущем эту недоработку можно будет исправить, ввиду доступности системы для редактирования и грамотного построения модульной структуры программы.

Программное обеспечение, которое создано при выполнении проекта было перенесено на хостинг в сети интернет и доступно по адресу <http://maker-s.ru/autorization.php>. Таким образом, доступ к системе имеют пользователи, находящиеся в любом месте мира.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ракова, О. А. Компьютерная адаптивная обучающая система для диагностики обученности студентов / О.А. Ракова // Молодой ученый -2010- №8. Т. 1 - С. 169-175.

2. Зайцева, Л. В. Методы и модели адаптации к учащимся в системах компьютерного обучения / Л. В. Зайцева // EducationalTechnology&Society 6(4) 2003. ISSN 1436-4522.-С. 204-211.

3. Zaitseva, L. V. Student models in Computer-based Education / L. V. Zaitseva L., C. Boule // Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies. ICALT 2003 – Athens, Greece, 2003, С. 451.

4. Денисова, И. Ю. Реализация адаптивной технологии обучения в информационной обучающей системе / И. Ю. Денисова, М. В. Баканова // Известия ПГУ им. В.Г. Белинского . 2012. №28: [Электронный ресурс] <http://cyberleninka.ru/article/n/realizatsiya-adaptivnoy-tehnologii-obucheniya-v-informatsionnoy-obuchayuschey-sisteme> (дата обращения: 13.03.2016).

5. Moodle -инструкция для преподавателя: [Электронный ресурс] [http://sdo.taom.ru/file.php/1/Instrukcii/Moodle\\_Instrukcija\\_dlja\\_prepodavatelja.pdf](http://sdo.taom.ru/file.php/1/Instrukcii/Moodle_Instrukcija_dlja_prepodavatelja.pdf).

6. Адаптивное обучение и адаптивный тестовый контроль: [Электронный ресурс] <http://testolog.narod.ru/Theory41.html>.

7. Адаптивное обучение, или несколько слов о Кnewton / Блог компании NewProfessionsLab / Хабрахабр: [Электронный ресурс] <https://habrahabr.ru/company/npl/blog/244539/>.

8. МОНАП-II - Авторские средства проектирования интеллектуальных обучающих систем: [Электронный ресурс] <http://www.dfki.de/~sosnovsky/papers/MONAP.pdf>.

9. Организация адаптивного обучения навыкам алгоритмической природы: [Электронный ресурс] <http://swsys.ru/index/php?page=article&id=1377/>.

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67



10. Обзор сценарных языков: [Электронный ресурс] <http://www.script-coding.com/Browse.html>.
- 11.PHP.SU–Преимущества PHP Основы PHP: [Электронный ресурс] <http://www.php.su/php/?oport>.
12. Теоретические основы управления адаптивными интеллектуальными образовательными системами: [Электронный ресурс] [http://www.rusnauka.com/10\\_NPE\\_2008/Pedagogica/30072.doc.htm](http://www.rusnauka.com/10_NPE_2008/Pedagogica/30072.doc.htm)
13. Котеров, Д. В. PHP 5 / Д. В. Котеров, А. Ф. Костарев // Санкт-Петербург.: БХВ-Петербург, 2005. – 1120 с.: ил.
14. Учебник по PHP. Самоучитель, справочник PHP: [Электронный ресурс] [http://www.softtime.ru/bookphp/g11\\_1.php](http://www.softtime.ru/bookphp/g11_1.php)
15. Учебник PHP. Онлайн учебник PHP для начинающих|PHP720: [Электронный ресурс] <http://php720.com/>
16. PHP.RU – Сообщество PHP-программистов: [Электронный ресурс] <https://php.ru/>
17. Что такое MySQL. Взаимодействие с PHP: [Электронный ресурс] [http://www.softtime.ru/bookphp/g112\\_1.php](http://www.softtime.ru/bookphp/g112_1.php)
18. MySQL.RU :.. Одобрено лучшими российскими программистами: [Электронный ресурс] <http://www.mysql.ru/docs/man/>
19. MySQL: Справочное руководство: [Электронный ресурс] <http://phpclub.ru/mysql/doc/>
20. СУБД MySQL. Учебник. Обучение MySQL: [Электронный ресурс] <http://www.webmasterwiki.ru/MySQL>
21. Современный учебник JavaScript: [Электронный ресурс] <http://learn.javascript.ru/>
22. Курсы — HTML Academy: [Электронный ресурс] <https://htmlacademy.ru/courses>
23. HTML для чайников: [Электронный ресурс] <http://www.site-do.ru/html/>

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68



24. HTML справочник | [html.manual.ru](http://html.manual.ru): [Электронный ресурс]  
<http://html.manual.ru/>

25. Apache.RU - Russian Apache Web-Server: [Электронный ресурс]  
<http://www.apache.ru/docs/>

26. Apache сервер: что такое и как настроить: [Электронный ресурс]  
<https://ru.hostings.info/chto-takoe-apache.html>

27. Установка и настройка Apache HTTP Server 2.2.22. Подключение модулей Apache, создание виртуального хоста: [Электронный ресурс]  
<http://php-myadmin.ru/learning/instrument-apache.html>

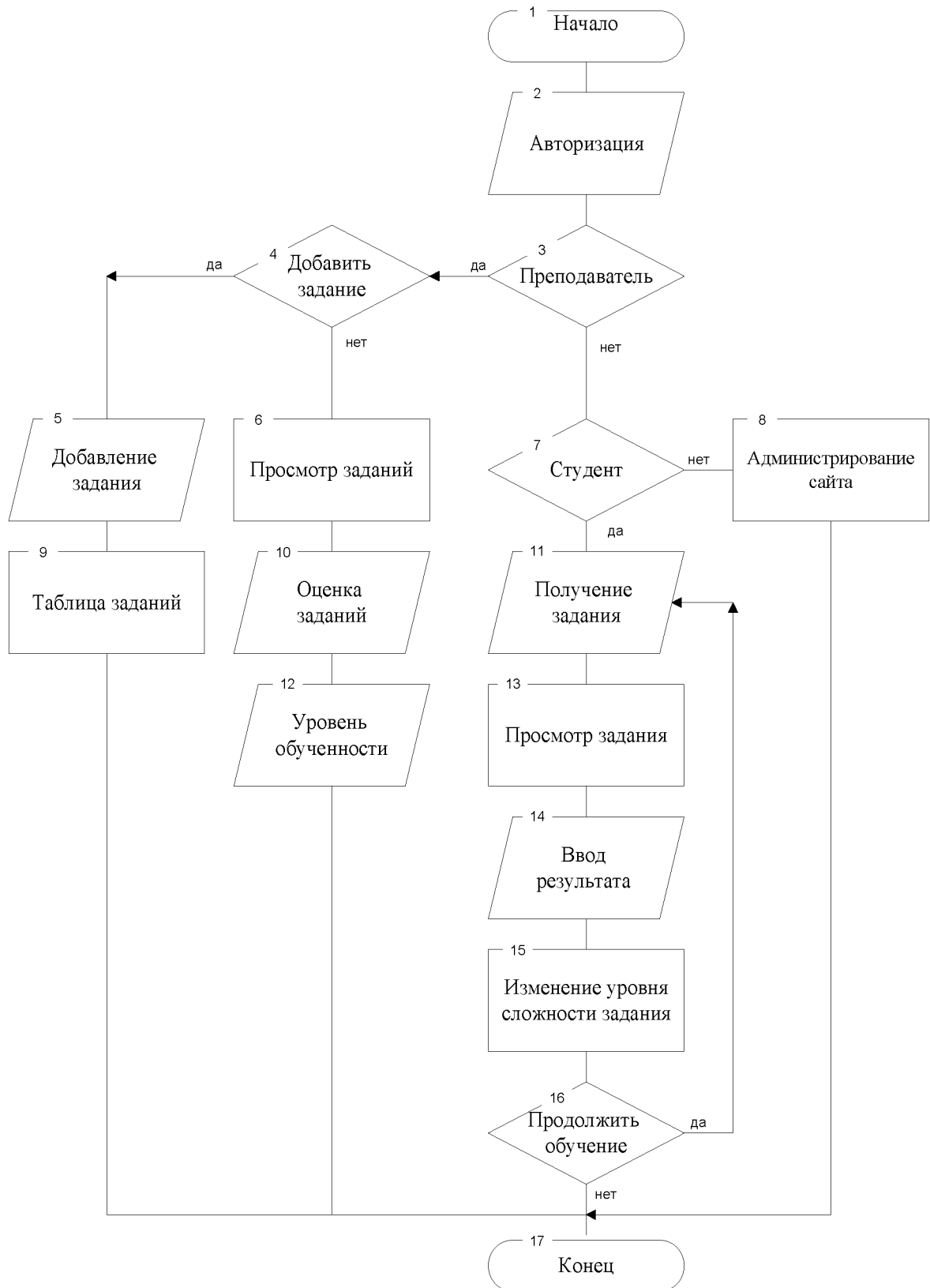
28. PHP Expert Editor v4.3 Rus - Твойсофт: [Электронный ресурс]  
<http://www.yoursoft.org.ua/raznoe/php-expert-editor-v4.3-rus.html>

29. Инструкции - Программное обеспечение Web-сервера. -  
Установка и настройка PHP Expert Editor: [Электронный ресурс]  
<http://xoops.ws/modules/instruction/page.php?id=290>

										Лист
										69
Изм.	Лист	№ докум.	Подпись	Дата	ДП - 230101.65 - ПЗ					

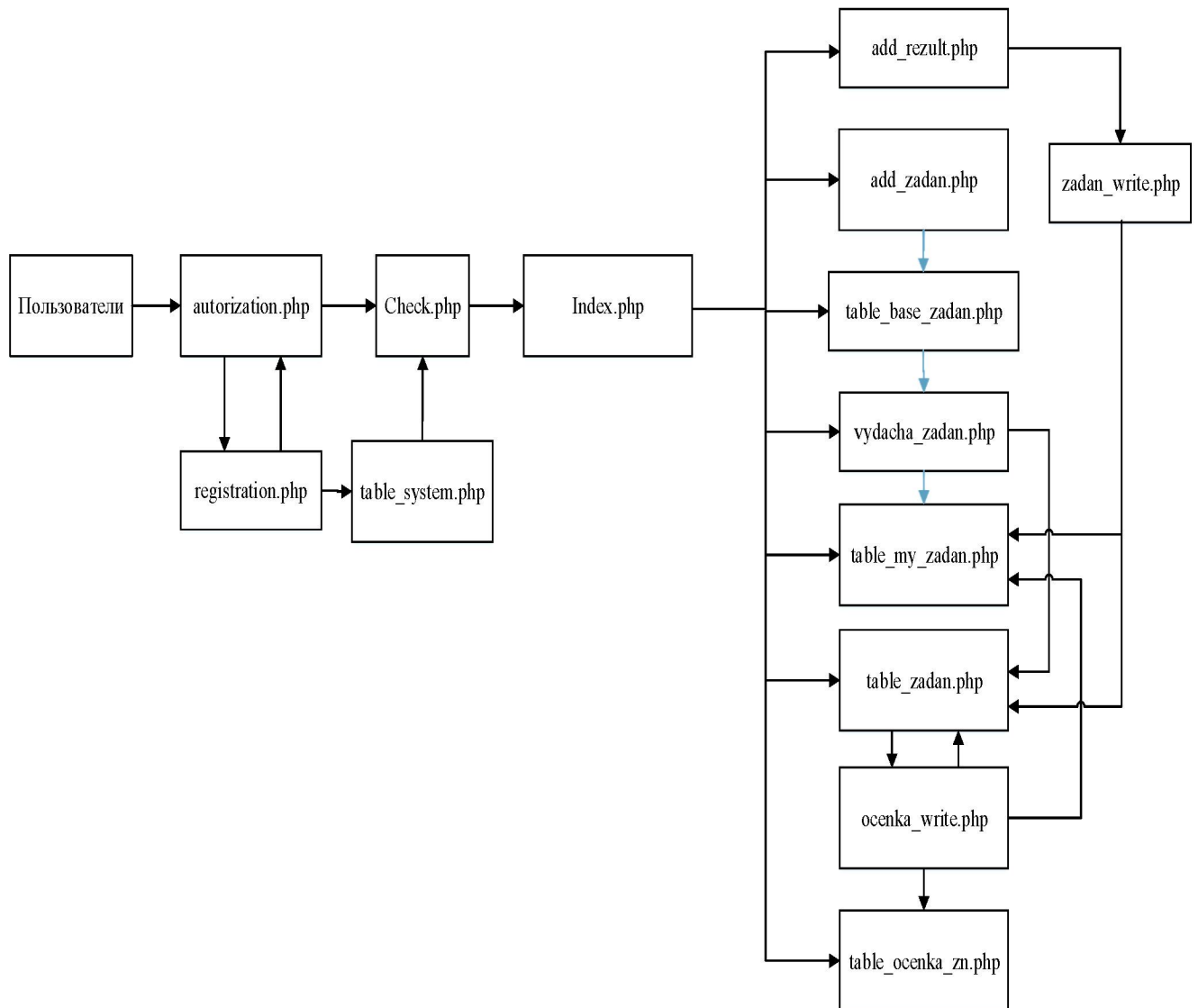
# ПРИЛОЖЕНИЕ А

## Блок-схема алгоритма работы клиентской части программы



## ПРИЛОЖЕНИЕ Б

### Структурная схема связей между модулями подсистемы



Изм.	Лист	№ докум.	Подпись	Дата

ДП - 230101.65 - ПЗ

Лист

71

## ПРИЛОЖЕНИЕ В

### Листинг программы

#### Модуль авторизации пользователей authorization.php

```
<html>
<head>
<title>Система адаптивного управления процессом выдачи учебных заданий</title>
<link rel="stylesheet" type="text/css" href="phpmyadmin.css.php">
</head>
<body class="loginform">
<div class="container">
<br /><br /><br /><br /><br /><br /><br /><br />
<h1 style="color: #000080">
СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ
ЗАДАНИЙ </h1>
<br>
<br>
//-----Формируем форму авторизации, обработчик Check.php-----
<form method="post" action="Check.php" name="login_form" target="_top" class="login">
<fieldset >
<legend align="center">
<font color="#000080">
Авторизация</font></legend>
<div class="item">
<p style="margin-top: 10px; margin-bottom: 5px">
<label for="input_username">Логин:</label>
// -----Ввод логина-----
<input type="text" name="login" id="input_username" value="" size="24" class="textfield">
</div>
<div class="item" style="margin-bottom: 5px">
<label for="input_password">Пароль:</label>
//-----Ввод пароля-----
<input type="password" name="pass" id="input_password" value="" size="24"
class="textfield">
</div>
<input type="hidden" name="server" value="1"></fieldset>
<fieldset class="tblFooters">
// -----Кнопка ввода-----
<input value="ОК" type="submit">
</fieldset>
</form>
<br />
//-----Если надо зарегистрироваться-----
<a href="registration.php">
```

									Лист
									72
Изм.	Лист	№ докум.	Подпись	Дата					

```

Регистрация
</a>
</div>
</body>
</html>

```

## Модуль регистрации нового пользователя registration.php

```

<html>
<head>
<title>Система адаптивного управления процессом выдачи учебных заданий</title>
<link rel="stylesheet" type="text/css" href="phpmyadmin.css.php">
</head>
<body bgcolor="#D2E6E6">
<div >
<br /><br />
<div style=" margin-left: 270px ">
<font face="Arial" size="4">
<br /><br /><br /><br /><br />
Регистрацияпользователя
</font>
</div>
<br />
//-----Форма регистрации, обработчик reg_write.php-----
<form action="reg_write.php" method="post">
<div style=" margin-left: 35px ">&nbsp;</div><font face="Arial" size="2">
<font face="Arial">
<div style=" margin-left: 100px " ><font face="Arial" size="2">
<font color="#000080">Фамилия Имя Отчество</font></div>
<div style="margin-top: -22px; margin-left: 270px" >
<input size="70" type="text" name="fio"></div>
<br /><br />

<div style=" margin-left: 200px" >
<font color="#000080">Группа</font></div>
<div style="margin-top: -22px; margin-left: 270px" >
<input type="text" name="grupp" id="theme" size="20" ></div>
<br /><br />
<div style=" margin-left: 200px" >
<font color="#000080">Логин</font></div>
<div style="margin-top: -22px; margin-left: 270px" >
<input type="text" name="login" id="theme" size="20" ></div>
<br /><br />
<div style=" margin-left: 200px" >
<font color="#000080">Пароль</font></div>
<div style="margin-top: -22px; margin-left: 270px" >
<input type="text" name="pass" id="theme" size="20" ></div>

```

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

```

</p>
<div align="left" style="margin-left: 270px">
//-----Кнопка ввода-----
<input type="submit" value="Зарегистрироваться" /></div>
    </font></font>
<?php
?>
</div>
</body>
</html>

```

## Модуль проверки логина и пароля пользователя Check.php

```

<html>

<head>
<title>Система адаптивного управления процессом выдачи учебных заданий</title>
</head>
<body>
<?php
$login=htmlspecialchars(stripslashes($_POST['login']));

$password=htmlspecialchars(stripslashes($_POST['pass']));
// -----Проверяем логин и пароль-----
if (((($login == "admin")&&($password == "sys")) || (($login == "teach")&&($password ==
"prp")))) {require_once "Index_1.php";}
else {
$login=$_POST['login'];
require_once "mysql_connect_doska.php";
//-----Запрос к базе-----
$r = mysql_query("SELECT * FROM polz where (login='$login'&&pass='$password') ORDER
BY id DESC")
or die(mysql_error());
//-----Ищем в массиве-----
for ($arr_guest=array(); $row=mysql_fetch_assoc($r); $arr_guest[]=$row);
//-----Что найдено-----
$count=sizeof($arr_guest);
//-----Пользователь есть-----
if ($count==1) {
//-----Запускаем гл модуль-----
require_once "Index_1.php";
}
//-----Не найден, ошибка-----
else {require_once "error.php";}
}
?>
</body>
</html>

```

					ДП - 230101.65 - ПЗ	Лист 74
Изм.	Лист	№ докум.	Подпись	Дата		

## Главный модуль программы Index.php

```
<html>
<head>
<meta http-equiv="Content-Language" content="ru">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ
ЗАДАНИЙ</title>
//-----Стили-----
<style>
A {
    TEXT-DECORATION: underline
}
A1:link {
    COLOR: #206еса
}
A:visited {
    COLOR: #0000ae
}
A:hover {
    COLOR: #800000; TEXT-DECORATION: underline
}
A:active {
    COLOR: #cc0000
}
</style>
</head>

<body bgcolor="#ECF4F4">

<div align="center">
<table border="0" width="1024" height="1" align="center">
    <tr>
    <td height="0" width="1024">

<table border="0" width="574" height="89">
    <tr>
        <td height="89" width="574">
            </td>
    </tr>
</table>
<table border="0" width="1034" height="0" cellpadding="10" cellspacing="0">
    <tr>
        <td height="22" width="1024" colspan="2" valign="bottom">

            <h1 style="font-size:15.0pt;font-family:Arial; margin-bottom:0px"
            valign="bottom" align="center"><FONT color=#B70000>СИСТЕМА
АДАПТИВНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ВЫДАЧИ УЧЕБНЫХ ЗАДАНИЙ
```





```

require_once "menu_tema.php";
}
elseif ($login == "teach") {$stud=$_GET['stud'];
$tema=$_GET['tema'];
require_once "menu_polz.php"; //-----Выпадающее меню пользователи-----
echo '<div style="margin-bottom: 10px; MARGIN: -47px 0px 10px 0px" vAlign="middle"
align="center">';
//-----Выпадающее меню тема-----
require_once "menu_tema.php";
echo '</div>';
//----Ищем логин по ФИО-----
$r = mysql_query("SELECT * FROM polz where (fio='$stud')")
or die(mysql_error());
$rows=mysql_num_rows($r);
$arr_guest=mysql_fetch_array($r);
$log_stud = $arr_guest["login"];
}
?>
</font></td>
</tr>

<?
if ($login <> "teach") {
//-----Формируем страницу для студента-----
?>
<tr>
  |
```

					ДП - 230101.65 - ПЗ	Лист 77
Изм.	Лист	№ докум.	Подпись	Дата		

```

        $no = "";
        $r = mysql_query("SELECT * FROM my_zadan where
(student='$login'&&tema='$stema'&&otvet_stud='$no') ORDER BY id DESC")
        or die(mysql_error());
        $rows=mysql_num_rows($r);
        $arr_guest=mysql_fetch_array($r);
        $nomer = $arr_guest["nomer"];
        $zadanie = $arr_guest["zadanie"];
        $z = $zadanie;
        $z = htmlspecialchars(stripslashes($z));
        $z = urlencode($zadanie);
        ?>
                &#9679;&nbsp;&nbsp;&nbsp;<a
href="add_rezult.php?login=<?=$login?>&tema=<?=$stema?>&id=<?=$id?>&nomer=<?=$nomer?>&zadanie=<?=$z?>">Ввести результаты выполнения</a></font>
        </tr>
<?
}

        if (($login == "teach") || ($login == "admin")) {
//-----Страница для преподавателя-----
?>
        <tr>
                <td height="198" width="496" background="image/Фон_1.jpg"
valign="middle">
                <p style="line-height: 150%; margin-top:-20px" align="left"><b><font
face="Arial">
                        <font color="#6600CC">Преподаватель добавляет задания в
информационную базу. Он может просмотреть выполненные задания
                        и выставить оценки по каждому заданию, а также оценить
уровень знаний студента.</font></b></font></td>
                <td height="198" width="498" background="image/Фон_2.jpg"
valign="middle">
                <font face="Arial" >
                <p style="line-height: 150%; margin-left:165px; margin-top:-15px"><b><font
face="Arial" size="3">
                        &#9679;&nbsp;&nbsp;&nbsp;<a
href="add_zadan.php?login=<?=$login?>&tema=<?=$stema?>&stud=<?=$stud?>">Добавитьно
воезадание</a></font>
                <p style="line-height: 150%; margin-left:165px"><b><font face="Arial"
size="3">
                        &#9679;&nbsp;&nbsp;&nbsp;<a
href="table_base_zadan.php?stud=<?=$stud?>&log_stud=<?=$log_stud?>&login=<?=$login?>
&tema=<?=$stema?>">Таблицазаданий</a></font>
                <p style="line-height: 150%; margin-left:165px"><b><font face="Arial"
size="3">
                        &#9679;&nbsp;&nbsp;&nbsp;<a href="vyd_zadan.php">Выдатьзаданиестуденту</a></font>

```

					ДП - 230101.65 - ПЗ	Лист 78
Изм.	Лист	№ докум.	Подпись	Дата		







```

}
else {
require_once "otkaz_1.php";
}
}
else { //----одно задание с неуд-----

$r = mysql_query("SELECT * FROM my_zadan where (student='$login'&&tema='$tema')
ORDER BY id DESC")
    or die(mysql_error());
    $rows=mysql_num_rows($r);
    $arr_guest=mysql_fetch_array($r);
//-----Рассчитываем время выдачи-----
$t_zadan = $arr_guest['time'] + 14400;
$t_zadan_vyd = $t_zadan + 86400;
$t_tec = time() + 14400;

if ($t_tec < $t_zadan_vyd) { // -----не прошли сутки-----
require_once "otkaz_7.php";
}
else { //выдать с уровнем l
ran($url, $login, $tema);
$d = date("d = F = Y ", 1); //-----Можно выдавать !-----
}

}
}
elseif (($rows > 0) and ($ocenka_prep <> 'неуд') and ($ocenka_prep <> "")) {
//-----успешно выполнено выдать следующего уровня-----
$r1 = mysql_query("SELECT * FROM vypoln where (student='$login'&&tema='$tema')
ORDER BY id DESC")
    or die(mysql_error());
    $arr_guest=mysql_fetch_array($r1);
    $url = $arr_guest["ur"];
    $r2 = mysql_query("SELECT * FROM my_zadan where (student='$login'&&tema='$tema')
ORDER BY id DESC")
    or die(mysql_error());
    $arr_guest=mysql_fetch_array($r2);
    $id = $arr_guest["id"];
    $url = $url + 1;
ran($url, $login, $tema); //-----выдаем задание на уровень больше-----
}
elseif (($rows > 0) and ($otvet_stud <> "") and ($ocenka_prep == "")) {
require_once "otkaz_3.php"; //-----оценка не выставлена-----
}

function ran($url, $login, $tema) {

$r = mysql_query("SELECT * FROM base_zadan where (tema='$tema'&&ur='$url') ORDER
BY id ASC")

```

					ДП - 230101.65 - ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		82

```

    or die(mysql_error());
$rows_1=mysql_num_rows($r);

if ($rows_1 == 0) {require_once "otkaz_4.php";
}
else {
for ($j = 0; $j < $rows_1; $j++) {
mysql_data_seek($r, $j);
$arr_guest=mysql_fetch_array($r);
$id = $arr_guest["id"];
$arr[$j] = $id;
}
$ran = array_rand($arr);
$id_ran = $arr[$ran];
$r = mysql_query("SELECT * FROM base_zadan where (id='$id_ran') ORDER BY id ASC")
    or die(mysql_error());
$rows=mysql_num_rows($r);
$arr_guest=mysql_fetch_array($r);
$idn = $arr_guest["id"];
$nomer = $idn;
$zadan = $arr_guest["zadanie"];
$student = $login;
$otvet_zadan = $arr_guest["otvet_prav"];
?>
<br /><b>Тема<? Print $tema?>Уровень<? Print $ur1?></b><br />
<br /><b>Задание № <? Print $idn?>для<?=$login?></b><br />
<?php
Print "<br />".$zadan."<br />";

$r = mysql_query("SELECT * FROM vypoln ORDER BY id DESC")
    or die(mysql_error());
//-----Пишем в базу-----
$sql = "INSERT INTO vypoln (tema, nomer, zadanie, ur, otvet_prav, student, otvet_stud,
ocenka_prep) VALUES ('$tema', '$nomer', '$zadan','$ur1','$otvet_zadan','$student',
'$otvet_stud', '$ocenka_prep)";
mysql_query($sql)
or die("Ошибказаписи: ".mysql_errno().": ".mysql_error()."<BR>");
$t = time()+14400;
//-----Пишем в базу выданных-----
$sql = "INSERT INTO my_zadan (student, tema, nomer, zadanie, ur, time, otvet_stud,
ocenka_prep) VALUES ('$login','$tema', '$nomer', '$zadan','$ur1','$t',
'$otvet_stud', '$ocenka_prep)";
mysql_query($sql)
or die("Ошибказаписи: ".mysql_errno().": ".mysql_error()."<BR>");
}
} // Конец функции
?>

</body>
</html>

```

					<i>ДП - 230101.65 - ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		83