**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Кафедра «Информатика и программное обеспечение»**

**«У Т В Е Р Ж Д А Ю»**

Зав. кафедрой «И и ПО», к.т.н., доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подвесовский А.Г.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2013г.

**ПРОГРАММНЫЙ УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС.   
МОДУЛЬ АДАПТИВНОГО ТЕСТИРОВАНИЯ.**

**ДИПЛОМНАЯ РАБОТА**

Документы текстовые

Всего \_\_\_\_ листов в папке

**Руководитель**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. Азарченков А.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

**Консультанты:**

по экономической части

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. Демиденко И.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

по организационной части

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. Филин С.С.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

**Нормоконтролер**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. Лагерев Д.Г.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

**Студент**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Евграфова Е.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2013 г.

Брянск 2013

**БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**на дипломную работу**

**Студенту\_\_Евграфовой Елены Владимировны\_\_\_группы** ***\_\_08-ПО1\_\_\_\_\_\_\_\_\_***

1. **Тема работы:** *Программный учебно-методический комплекс. Модуль адаптивного тестирования*

**Утверждена приказом по БГТУ №** **\_\_284-3\_\_** **от \_\_ 19.04 \_\_ 2013г.**

1. **Срок сдачи дипломной работы**  *июнь* 2013г.
2. Рекомендуемые инструментальные средства PHP, MySQL \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Содержание папки текстовых документов (перечень разделов)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Аналитическая часть. 1.1. Общие сведения об электронном общении в организациях,1.2. Средства распространения графической информации,1.3. Обзор программных средств,1.4. Моделирование процессов использования ПО для распространения изображений,1.5. Постановка задачи дипломной работы.

2. Техническое задание. 2.1. Введение,2.2. Полное описание,2.3. Специфические требования,2.4. Этапы разработки,2.5. Порядок контроля и приемки.

3. Исследовательская часть. 3.1. Цели и задачи исследования,3.2. Форматы изображений.

4. Конструкторская часть. 4.1. Выбор средств разработки,4.2. Архитектура программного средства,4.3. Алгоритм работы проектируемой системы.

5. Экспериментальная часть. 5.1. Назначение тестирования,5.2. Тестирование в нормальных условиях, 5.3. Проверка программной системы в экстремальных условиях,5.4. Проверка программной системы в исключительных ситуациях,5.5. Основные выводы

6. Техническая документация. 6.1. Руководство пользователя,6.2. Руководство администратора.

7. Экономическая часть

8. Безопасность и экологичность проекта

5. Перечень плакатов с указанием их названия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Цели (1 лист)

2. Аналоги (2 листа)

3. Исследование существующих форматов изображений (1 лист)

4. Общая архитектура приложения (1 лист)

5. Алгоритм работы модуля адаптивного тестирования (1 лист)

6. Модель представления данных (1 лист)

7. Алгоритм работы приложения (1 лист)

8. Интерфейс приложения (1 лист)

9. Экономические показатели (1 лист)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Итого 11 листов

6. Особые замечания \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013г.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подвесовский А.Г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Азарченков А.А.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013г.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Евграфова Е.В.

# Аннотация

На основании задания дипломной работы разработан модуль адаптивного тестирования.

В аналитической части тестирование рассмотрено как средство контроля знаний, изучены существующие методы тестирования, в частности адаптивное тестирование. Проведён обзор и анализ существующих программных систем-аналогов, выделены их достоинства и недостатки, а также сделан вывод об актуальности создания модуля.

В техническом задании определены основание для разработки и назначение разработки. Описаны основные требования к программной системе и этапы ее разработки.

В исследовательской части проведено исследования существующих алгоритмов адаптивного тестирования, разработан собственный адаптивный алгоритм.

В конструкторской части рассмотрена архитектура программной системы, выполнено обоснование выбора языка и средств разработки, проведено моделирование системы, базы данных и интерфейса.

В части технической документации разработаны необходимые документы, сопровождающие программный продукт.

В экспериментальной части описано проведенное тестирование разработанной системы в нормальных условиях, нагрузочное тестирование и тестирование в исключительных ситуациях.

В экономической части дипломной работы приведены расчеты затрат на создание программной системы и проведена оценка эффективности ее внедрения.

В организационной части приводятся рекомендации по охране труда и технике безопасности.

**СОДЕРЖАНИЕ**

[Аннотация 2](#_Toc356465536)

[Введение 6](#_Toc356465537)

[1 Аналитическая часть 8](#_Toc356465538)

[1.1 Общие сведения об электронном образовании 8](#_Toc356465539)

[1.2 Тестирование как средство контроля знаний 9](#_Toc356465540)

[1.3 Методы тестирования 11](#_Toc356465541)

[1.4 Адаптивное тестирование 12](#_Toc356465542)

[1.5 Обзор программных средств 14](#_Toc356465543)

[1.6 Моделирование процессов адаптивного тестирования 21](#_Toc356465544)

[1.6.1 Модель вариантов использования 22](#_Toc356465545)

[1.6.2 Функциональная модель IDEF0 23](#_Toc356465546)

[1.7 Постановка задачи дипломной работы 2](#_Toc356465547)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc356465548)

[2.1 Введение 3](#_Toc356465549)

[2.1.1 Основание для разработки 3](#_Toc356465550)

[2.1.2 Назначение 3](#_Toc356465551)

[2.1.3 Определения, сокращения, аббревиатуры 3](#_Toc356465552)

[2.2 Полное описание 4](#_Toc356465553)

[2.2.1 Перспектива 4](#_Toc356465554)

[2.2.2 Функция изделия 4](#_Toc356465555)

[2.2.3 Характеристика пользователей 5](#_Toc356465556)

[2.2.4 Ограничения 5](#_Toc356465557)

[2.3 Специфические требования 5](#_Toc356465558)

[2.3.1 Требования к внешним интерфейсам. 5](#_Toc356465559)

[2.3.2 Функциональные требования 5](#_Toc356465560)

[2.3.3 Требования к рабочим характеристикам 6](#_Toc356465561)

[2.4 Этапы разработки 6](#_Toc356465562)

[2.5 Порядок контроля и приемки 7](#_Toc356465563)

[3 Исследовательская часть 8](#_Toc356465564)

[3.1 Исследование алгоритмов адаптивного тестирования 8](#_Toc356465565)

[3.2 Жесткий алгоритм тестирования 8](#_Toc356465566)

[3.3 Мягкий алгоритм тестирования 9](#_Toc356465567)

[3.4 Комбинированный алгоритм 10](#_Toc356465568)

[3.5 Динамическая модель адаптивного тестирования 11](#_Toc356465569)

[3.6 Модель Раша (Rasch) 13](#_Toc356465570)

[3.7 Разработка собственного алгоритма тестирования 15](#_Toc356465571)

[3.8 Выводы 18](#_Toc356465572)

[4 Конструкторская часть 20](#_Toc356465573)

[4.1 Выбор средств разработки 20](#_Toc356465574)

[4.1.1 Выбор языка программирования 20](#_Toc356465575)

[4.1.2 Выбор СУБД 23](#_Toc356465576)

[4.1.3 Выбор среды разработки 24](#_Toc356465577)

[4.1.4 Выбор web-сервера 27](#_Toc356465578)

[4.2 Архитектура программного средства 29](#_Toc356465579)

[4.3 Моделирование работы автоматизированной системы 32](#_Toc356465580)

[4.3.1 Модель размещения 32](#_Toc356465581)

[4.3.2 Модель хранения данных 33](#_Toc356465582)

[4.3.3 Модель потоков данных 35](#_Toc356465583)

[4.4 Проектирование пользовательского интерфейса 40](#_Toc356465584)

[4.5 Алгоритм работы проектируемой системы 43](#_Toc356465585)

[5 Экспериментальная часть 45](#_Toc356465586)

[5.1 Назначение тестирования 45](#_Toc356465587)

[5.2 Тестирование в нормальных условиях 45](#_Toc356465588)

[5.3 Проверка программной системы в экстремальных условиях 47](#_Toc356465589)

[5.4 Проверка системы в исключительных ситуациях 47](#_Toc356465590)

[5.5 Основные выводы 47](#_Toc356465591)

[6 Техническая документация 48](#_Toc356465592)

[6.1 Руководство пользователя 48](#_Toc356465593)

[6.2 Руководство администратора 49](#_Toc356465594)

[ЗаключЕние 54](#_Toc356465595)

[Список литературы 55](#_Toc356465596)

# Введение

Под электронным образованием обычно понимают учебные методы, в которых в качестве учебного посредника выступает компьютер. Это – целенаправленная учебная деятельность, которая включает в себя одновременно обучения и изучения, а для общения с аудиторией использует, главным образом, IT-технологии.

На сегодняшний день создание качественных и эффективных электронных образовательных ресурсов (ЭОР) является одной из главных задач в области информатизации образования. Если брать во внимание «эволюцию» электронных продуктов, то на смену текста графическим приходят высоко интерактивные, мультимедийно насыщенные электронные образовательные ресурсы. При этом необходимо обеспечить возможность их сетевого распространения.

ЭОР можно разделить на три уровня.

Самые простые ЭОР – текстографические. Они отличаются от книг в основном формой предъявления текстов и иллюстраций: материал представляется на экране компьютера, а не на бумаге. Но его очень легко распечатать, т.е. перенести на бумагу [1].

ЭОР следующего уровня тоже текстографические, но имеют существенные отличия в навигации по тексту. Страницы книги мы читаем последовательно, осуществляя, таким образом, так называемую линейную навигацию. При этом довольно часто в учебном тексте встречаются термины или ссылки на другой раздел того же текста. В таких случаях книга не очень удобна: нужно разыскивать пояснения где-то в другом месте, листая множество страниц. В ЭОР же это можно сделать гораздо комфортнее: указать незнакомый термин и тут же получить его определение в небольшом дополнительном окне или мгновенно сменить содержимое экрана при указании так называемого ключевого слова (либо словосочетания). В данном случае навигация по тексту является нелинейной (вы просматриваете фрагменты текста в произвольном порядке, определяемом логической связностью и собственным желанием) [1].

Третий уровень ЭОР – это ресурсы, целиком состоящие из визуального или звукового фрагмента. Отличия от книги здесь очевидны: ни кино, ни анимация (мультфильм), ни звук в полиграфическом издании невозможны. Но, с другой стороны, стоит заметить, что такие ЭОР по существу не отличаются от аудио- и видео-продуктов, воспроизводимых на бытовом CD-плейере [1].

Актуальным вопросом является использование информационных технологий при разработке автоматизированных систем обучения и контроля знаний. Использование подобного типа систем в образовательном процессе позволит применить новые адаптивные алгоритмы тестового контроля, использовать в тестовых заданиях мультимедийные возможности компьютеров, уменьшить объем бумажной работы, ускорить процесс подсчета результатов, снизить затраты на организацию и проведение тестирования.

***Цель*** – реализация алгоритма адаптивного тестирования в рамках учебно-методического комплекса «УМК-А» c помощью Web-технологий.

***Задачи:***

* Исследование предметной области;
* Разработка технического задания;
* организация работы сайта на основе базы данных и разграничения прав доступа к имеющейся информации. Доступ к ресурсам сайта возможен при регистрации или авторизации пользователя.

***Объект исследования*** – алгоритмы адаптивного тестирования, применимые в системах электронного образования.

***Предмет исследования*** – реализация алгоритма адаптивного тестирования в рамках учебно-методического комплекса «УМК-А».

# Аналитическая часть

## Общие сведения об электронном образовании

Образование – одна из важнейших сфер человеческой деятельности, которая обеспечивает формирование интеллектуального потенциала общества, и поэтому в систему образования привлекают современные информационные и коммуникационные технологии, основанные на компьютерных сетях. Бурное развитие информатики и информационных технологий ставит перед образованием проблему использования новых технических средств, совершенствования образовательных методик. Перед учеными и педагогами стоит задача оптимизации объективного процесса информатизации образования.

Нужно сказать, что использование в учебном процессе компьютерных технологий имеет оптимальный порог, за пределами которого дополнительное насыщение ведет к снижению позитивного эффекта на успеваемость. Наиболее эффективно компьютерные технологии влияют на успеваемость, когда основная цель их использования – поддержание способности к умственному восприятию и переработке информации, а не просто модернизация представления учебного материала.

Однако стоит заметить, что новые технологии только тогда могут быть эффективны в образовании, когда они не вписываются в уже существующую образовательную систему, а входят как элемент в новую систему образования [2].

Интенсивное развитие процесса информатизации образования влечет за собой расширение сферы применения электронных образовательных ресурсов. Например, можно вполне определенно выделить такие успешно и активно развивающиеся направления в образовании, как реализация возможностей программных средств учебного назначения в качестве средства обучения, объекта изучения, средства управления, средства коммуникации, средства обработки информации.

Возможности современной вычислительной техники в значительной степени адекватны педагогическим и методическим потребностям образования:

* вычислительные – быстрое и точное преобразование любых видов информации (числовой, текстовой, графической, звуковой и др.);
* трансдьюсерные – способность компьютера к приему и выдаче информации в самой различной форме (при наличии соответствующих устройств);
* комбинаторные – возможность запоминать, сохранять, структурировать, сортировать большие объемы информации, быстро находить необходимую информацию;
* графические - представление результатов своей работы в четкой наглядной форме (текстовой, звуковой, в виде рисунков и пр.);
* моделирующие - построение информационных моделей (в том числе и динамических) реальных объектов и явлений.

## Тестирование как средство контроля знаний

По сравнению с традиционными методами контроля знаний студентов компьютерный контроль обладает явными преимуществами. Одно из них – сокращение времени, необходимого для проверки и оценки качества заданий, выполненных обучаемыми, а также на анализ полученных данных. Немаловажно и то, что компьютерный контроль в отличие от традиционных методов проверки усвоения учебного материала может использоваться в группах студентов любой целесообразной численности.

К основным преимуществам тестирования можно отнести следующее:

* удобство использования;
* снижение трудоемкости;
* исключение ошибок при проверке (у компьютеров не бывает снижения внимательности по причине усталости);
* оценивание результатов тестирования осуществляется мгновенно;
* возможность формирования тестов, различных по уровню обученности испытуемых;
* каждый тестируемый выбирает самостоятельный темп работы с тестом;
* при компьютерном тестировании легко ввести временные ограничения или временное отслеживание процесса тестирования;
* использование мультимедийных компонентов и графических изображений высокого качества и т. д.

Среди недостатков тестирования можно отметить трудность формулировки ответов на вопросы по предметам, связанным с общими законами развития природы, общества (философия, история, экономика, некоторые разделы астрономии, физики и т.д.), для которых свойственны неоднозначные ответы, и двоичной логики "да/нет" – недостаточно. Также, с помощью тестирования невозможно контролировать творческие знания. Таким образом, проверка понимания и умений в некоторых областях гуманитарных, общественно-политических дисциплин вне прерогативы компьютерного тестирования. Для этого существуют иные методы оценки.

Существенный недостаток тестирования, косвенно связанный с вышеперечисленными проблемами, это то, что обучаемые привыкают к готовым формулировкам, и теряют (не приобретают) способность свободно и грамотно излагать мысли.

## Методы тестирования

В большинстве источников, когда речь заходит о тестовых технологиях в учебном процессе, тест рассматривается как элемент контроля.

Компьютерное тестирование – это средство, которое позволяет с минимальными затратами времени преподавателя объективно проверить знание большого количества студентов. Компьютерные тесты положительно воспринимаются студентами. Преимуществом компьютерного тестирования является автоматическая проверка результатов и исключение влияния человеческого фактора. Еще одним преимуществом компьютерных тестов является то, что в процессе проверки знаний студенты видят в преподавателе не оппонента, а союзника.

Различают классическое тестирование и тестирование, построенное на основе адаптивных алгоритмах. К классическому тестированию относятся тесты, с заранее известной последовательностью заданий. При написании таких тестов, испытуемый, независимо от своих знаний, пройдет заданное изначально количество вопросов, при этом имея возможность вернуться к предыдущим вопросам и изменить свой ответ. В отличие от классического тестирования, адаптивное основывается на таком алгоритме, который позволяет выявить оценку знаний тестируемого при минимальных затратах времени и количестве заданных вопросов.

Важнейшими критериями адаптивных тестов являются:

* Действенность теста: полнота, всесторонность проверки, пропорциональность представления всех элементов изучаемых знаний, умений;
* надежность теста: характеризуется стабильностью, устойчивостью показателей при повторных измерениях с помощью того же теста или его равноценного заменителя. Надежность тестов обученности значительно зависит от трудности их выполнения, которая в свою очередь определяется соотношением правильных и неправильных ответов на тестовые вопросы;
* дифференцированность теста: способность теста отделить тех, кто усвоил материал на необходимом уровне, от тех, кто заданного уровня не достиг.

## Адаптивное тестирование

Адаптивное тестирование определяется в «Стандартах образовательного и психологического тестирования» как «последовательная форма тестирования, при которой последующие задания теста выбираются в зависимости от ответов на предыдущие задания». Отсюда компьютеризированное адаптивное тестирование — это адаптивное тестирование, проведение которого облегчается благодаря использованию компьютера.

Адаптивное тестирование основано на принципах последовательного анализа, впервые изложенных А. Вальдом [9]. Основной целью этой процедуры является создание теста либо в когнитивной, либо в аффективной области, состоящего из заданий, которые не выходят за пределы способностей индивидуума. Это достигается путем привлечения группы экспертов в данной области или опытных создателей тестов для разработки банка заданий, соответствующих измеряемой области. Каждое задание из этого банка затем ранжируется от низкого к высокому уровню трудности на основе информации о пропорции тестируемых, дающих на него правильный ответ, или пропорции ответов на какой-либо пункт определенным способом.

С началом внедрения компьютерных технологий и применения теории «задание-ответ» (IRT) в качестве инструмента анализа заданий была создана основа для использования компьютеров в адаптивном тестировании. Процедура выбора заданий и порядка их предъявления тестируемому существенно облегчалась машинными операциями.

Стратегия заключается в предъявлении тестируемому следующего по порядку задания умеренной трудности, которое выбирается компьютером из общего банка заданий в зависимости от результата предыдущего ответа. Задания в этом банке градуируются с использованием IRT-методов, реализованных на уровне программного обеспечения. В этом смысле можно сказать, что тест «адаптирует сам себя» к уровню способностей тестируемого. Конечной желаемой целью процедур компьютеризированного адаптивного тестирования является разработка теста, который обнаруживал бы практически одинаковую валидность и надежность в отношении всего диапазона тестируемого содержания или измеряемых конструктов.

К числу важных преимуществ компьютеризованного адаптивного тестирования можно отнести.

1. Тестирование является более качественным и объективным способом оценивания, его объективность достигается путем стандартизации процедуры проведения, проверки показателей качества заданий и тестов целиком.
2. Тестирование – более справедливый метод, оно ставит всех учащихся в равные условия, как в процессе контроля, так и в процессе оценки, практически, исключая субъективизм преподавателя. По данным английской ассоциации NEAB, занимающейся итоговой аттестацией учащихся Великобритании, тестирование позволяет снизить количество апелляций более чем в три раза, сделать процедуру оценивания одинаковой для всех учащихся вне зависимости от места проживания, типа и вида образовательного учреждения, в котором занимаются учащиеся.
3. Тесты – это более объёмный инструмент, поскольку тестирование может включать в себя задания по всем темам курса, в то время как на устный экзамен обычно выносится 2-4 темы, а на письменный – 3-5. Это позволяет выявить знания учащегося по всему курсу, исключив элемент случайности при вытаскивании билета. При помощи тестирования можно установить уровень знаний учащегося по предмету в целом и по отдельным его разделам.
4. Тест – это более точный инструмент, так, например, шкала оценивания теста из 20 вопросов, состоит из 20 делений, в то время, как обычная шкала оценки знаний — только из четырёх.
5. Тестирование более эффективно с экономической точки зрения. Основные затраты при тестировании приходятся на разработку качественного инструментария, то есть имеют разовый характер. Затраты же на проведение теста значительно ниже, чем при письменном или устном контроле. Проведение тестирования и контроль результатов в группе из 30 человек занимает полтора два часа, устный или письменный экзамен — не менее четырёх часов.
6. Тестирование — это более мягкий инструмент, они ставят всех учащихся в равные условия, используя единую процедуру и единые критерии оценки, что приводит к снижению предэкзаменационных нервных напряжений.

## Обзор программных средств

Немаловажным вопросом является оценка качества программных средств тестирования. Качество обучающих средств является одним из условий успешной интеграции компьютерных технологий в учебный процесс.

GMAT (The Graduate Management Admission Test) – тест, результаты которого засчитывают более 1500 бизнес-школ по всему миру. Этот стандартизированный компьютерный тест позволяет бизнес-школам максимально оценить уровень подготовки кандидатов к дальнейшему обучению в области бизнеса и менеджмента при поступлении на программы MBA или другие пост бакалаврские программы по менеджменту. Ежегодно тестирование проходят более 200 тысяч человек.

Особенностью GMAT (рис. 1.1) является то, что тест включает в себя проверку навыков речи, письма и математического счета. Сложность заключается в том, что тест принимается на английском, и, следовательно, требует высокого уровня знания английского языка.

GMAT разработан с целью проверки навыков, необходимых в сфере менеджмента. Однако он не требует специальных знаний и не касается узких областей специализации.

Тест GMAT уникален тем, что по его результатам можно определить потенциальные возможности человека, прогнозировать его успехи в ходе обучения в бизнес-школе.

Тест разработан в США, но используется и в других странах. Существует ряд аналогов GMAT, но популярны они, как правило, в неанглоязычных странах. [3]

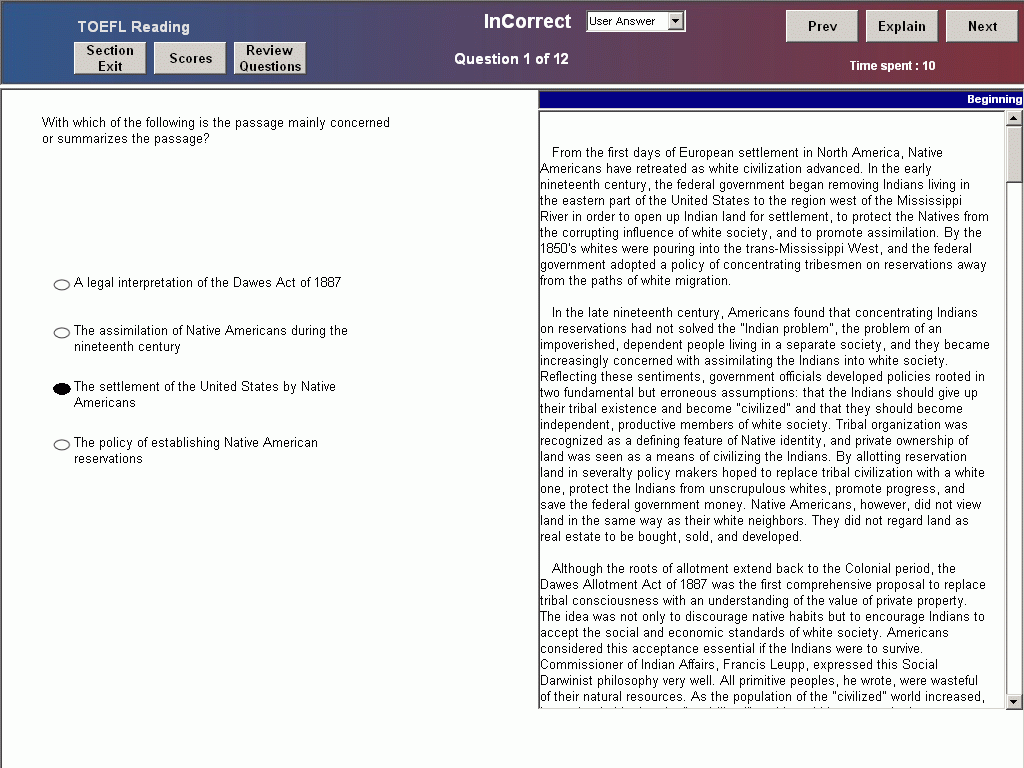


Рис. 1.1.Окно программы GMAT

Первый вопрос каждого раздела относится к средней группе сложности. Если вы отвечаете правильно, уровень сложности следующего вопроса повышается. Вернуться к предыдущему заданию невозможно, а переход к следующему вопросу происходит только после ответа на текущий [3].

GMAT не подходит для кафедры «Информатика и программное обеспечение», поскольку предполагает:

* знание английского языка на высоком уровне;
* направленность тестируемого на дальнейшее обучение в сфере экономики;
* прохождения тестирования занимает достаточно длительное время, порядка трех часов;
* необходимость в привлечении третьих лиц при определении окончательного результата.

Программа АСТ-Тест Player (рис. 1.2) предназначена для прохождения тестов. Функционал программы определяется параметрами конкретного теста и настройками системы тестирования.

АСТ-Тест Player – комплекс программ для компьютерного тестирования. Клиентский компонент данной системы представлена модулем для прохождения тестов, т.е. основной функцией «АСТ-теста» является контроль, а не формирование навыков и умений.

Функции приложения:

* обеспечение тестовых испытаний;
* создание библиотеки пользователей и тестов;
* создание расписания тестирования;
* контроль за ходом тестирования;
* получение результатов тестирования;
* диагностика работы системы;[4]
* возможность динамического формирования на основе банка тестовых заданий неограниченного количества вариантов тестов, различающихся по составу заданий, тематической структуре, способу оценивания результатов и алгоритмов тестирования.[5]

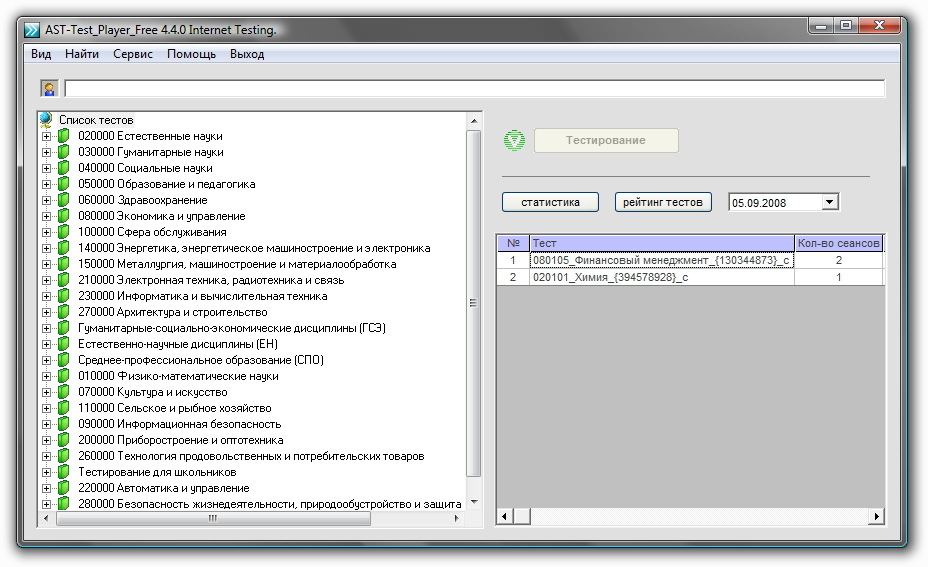


Рис. 1.2. Окно программы, предоставляющее выбор теста через список, сгруппированный по специальностям.

Возможности:

* авторизация по логину/паролю;
* выбор теста из списка;
* отображение инструкции по тесту;
* задания с аудио-содержанием;
* 4 формы заданий;
* виды оценки: процент, класс, балл, на основе меры трудности;
* ограничение по времени: задание, тест;
* пропуск заданий / изменение ответа на задание;
* отображение признака верно/неверно после ответа на задание;
* восстановление прерванного сеанса тестирования;
* отображение результата тестирования;
* просмотр неверно выполненных заданий;
* просмотр успеваемости по разделам теста;
* отображение истории сеансов тестирования;
* тестирование в локальной сети или через Internet;
* работа в терминальном режиме.[6]

ACT-Тест Player содержит ряд преимуществ, которые не плохо было бы иметь в разрабатываемом продукте для кафедры «Информатика и программное обеспечение». Однако система предполагает интерактивный урок при изучении какой-либо области и дальнейший контроль полученных знаний.

Moodle — система управления курсами (электронное обучение), также известная как система управления обучением или виртуальная обучающая среда (рис. 1.3). Представляет собой свободное веб-приложение, предоставляющее возможность создавать сайты для онлайн-обучения.

Первый способ организации адаптивного тестового контроля в Moodle. Допустим, имеется базовый тест средней трудности (на «4»). При выполнении теста на 100% учащийся переходит к тесту Moodle на «5», при определенном количестве неправильных ответов (скажем 75% правильных) – переход к тесту Moodle на «3».

Такую схему можно реализовать с помощью элемента Moodle «Лекция». В поле Зависит можно выбрать зависимость от другого элемента Moodle «Лекция-Занятие». Так для теста на «3» выбирается Базовый тест и пройден на 75%, а для теста на «5» выбирается также Базовый тест, но пройден на 100%. Все эти три элемента «Лекция-Занятие» (базовый тест, тест на «3» и тест на «5» будут состоять только из тестов).

Такой механизм можно реализовать как в Moodle 1.9.x, так и Moodle 2.x. Кроме того, в Moodle 2.x. можно устанавливать зависимость от процента прохождения предыдущих тестов. Для Moodle 1.9.x есть специальный плагин под название QuizPort.

Он работает только с тестами [Hot Potatoes](http://moodlefree.ru/groups/hot-potatoes-besplatnaya-programma-dlya-sozdaniya-testov) и Qedos. Главное достоинство этого модуля – возможность устанавливать зависимость от прохождения других тестов.



Рис. 1.3. Обучающая система Moodle

Анализируя представленные системы тестирования, хотелось бы подчеркнуть, что важным аспектом разработки тестируемых сред является:

* в клиентском компоненте наглядный пользовательский интерфейс в различных форматах;
* в административном компоненте возможность добавления пользователей на основе авторизации, создание расписания тестирования для каждого пользователя;
* в конструктивном компоненте возможность добавлять и редактировать БД вопросов;
* в серверном компоненте надежность хранения учебных материалов и базы данных пользователей и доступность сервера через сеть Интернет.

Каждая из рассмотренных нами систем располагает полезными и удобными возможностями, но не отвечает в полной мере потребностям кафедры «Информатика и программное обеспечение» в плане возможности использования адаптивного тестирования в необходимом объеме.

В табл. 1.1 представлены достоинства и недостатки выше перечисленных программ-аналогов.

Таблица 1.1

Сравнительная характеристика

|  |  |  |  |
| --- | --- | --- | --- |
| Программы    Критерии | GMAT | АСТ- Тест Player | Moodle |
| Область применения | Область бизнеса и менеджмента при поступлении на программы MBA или другие пост бакалаврские программы по менеджменту. | Учебные заведения. | Учебные заведения. |
| Назначение | Проверка навыков, необходимых в сфере менеджмента. | Контроль знаний. | Управление обучением и виртуальная обучающая среда. |
| Особенности | включает в себя проверку навыков речи, письма и математического счета. | Использование мультимедийных заданий. | Возможность устанавливать зависимость от процента прохождения предыдущих тестов. |
| Недостатки | Знание английского языка на высоком уровне.  Направленность тестируемого на дальнейшее обучение в сфере экономики.  Тестирование занимает достаточно длительное время.  Необходимость в привлечении третьих лиц при определении окончательного результата. | Проведение интерактивного урока перед прохождением тестирования. | В системе не предусмотрены учебные группы, что делает очень сложным учет студентов разных специальностей. |

В результате анализа предметной области было выяснено, что создание сайта кафедры с возможностью прохождения предметного адаптированного тестирования (дистанционно, например, с целью тренировки) или стационарно (с последующим знанием обучаемого) является актуальным.

## Моделирование процессов адаптивного тестирования

Моделирование процессов можно разделить на два подхода:

* объектно-ориентированный;
* структурный.

В рамках объектно-ориентированного подхода система разбивается на набор объектов, соответствующих объектам реального мира, взаимодействующих между собой путем посылки сообщений.

Отличительной особенностью является объединение в объекте как атрибутивных данных (характеристики, свойства), так и поведения (функции, методы). Второй особенностью является объединение в объекте как атрибутивных данных (характеристики, свойства), так и поведения (функции, методы). В функционально-ориентированных системах функции и данные хранятся (существуют) отдельно.)

В объектно-ориентированном подходе иерархия выстраивается с использованием двух отношений: композиции и наследования. При этом в объектно-ориентированном подходе «объект-часть» может включаться сразу в несколько «объектов-целое». Таким образом, модуль в данном подходе представляется в виде ориентированного графа, т. е. с помощью более общей структуры.

Одной из методологий объектно-ориентированного подхода является модель вариантов использования.

При анализе и проектировании структурным подходом принято называть метод исследования системы, основанный на представлении ее в виде иерархии взаимосвязанных функций. Описание системы начинается с ее общего обзора и затем детализируется, приобретая иерархическую структуру со всё большим числом уровней. Разбиение на уровни абстракции производится с ограничением числа элементов на каждом из них. Описание каждого уровня включает в себя только существенные для этого уровня элементы. Процесс разбиения продолжается вплоть до конкретных процедур, дальнейшая детализация которых не имеет смысла. При этом автоматизируемая система должна сохранять целостное представление, в котором все составляющие ее компоненты взаимоувязаны.

Одной из методологий структурного анализа и проектирования является модель IDEF0.

### Модель вариантов использования

UML (Unified Modeling Language) – стандартный язык для написания моделей анализа, проектирования и реализации объектно-ориентированных программных систем. Может использовать для визуализации, спецификации, конструирования и документирования результатов программных проектов.

Диаграмма Use Case определяет поведение системы с точки зрения пользователя. Рассматривается как главное средство для первичного моделирования динамики системы, используется для выяснения требований к разрабатываемой системе, фиксации этих требований в форме, которая позволит проводить дальнейшую разработку.

Достоинствами модели вариантов использования являются такие аспекты, как:

* определяет пользователей и границы системы;
* определяет системный интерфейс;
* является основой для написания пользовательской документации;
* хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

В Русской литературе Use Case часто называют диаграммами прецедентов или диаграммами вариантов использования.[7]

Приведем диаграмму вариантов использования (рис.1.3). Студент и Администратор являются актерами, причем Администратор является потомком пользователя, поэтому может использовать все возможности предусмотренные пользователю. Помимо этого Администратор может формировать новое тестирование. Студент имеет такие возможности, как прохождение тестирования и просмотр результатов пройденных тестов.

### Функциональная модель IDEF0

Методология IDEF0 (SADT – методология структурного анализа и проектирования) представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели системы. Является одной из самых известных методологий анализа и проектирования систем [19]. Она способна наиболее полно отражать такие характеристики, как управление, обратная связь и ресурсы. Другая ее особенность заключается в том, что она развивалась как язык описания функционирования систем общего вида.

Модель IDEF0 позволяет:

* описывать любые системы, а не только информационные;
* создать описание системы и ее внешнего окружения до определения окончательных требований к ней. Иными словами, с помощью данной методологии можно постепенно выстраивать и анализировать систему даже тогда, когда трудно еще представить ее воплощение.

Таким образом, IDEF0 может применяться на ранних этапах создания широкого круга систем. В то же время она может быть использована для анализа функций существующих систем и выработки решений по их улучшению.

|  |  |
| --- | --- |
|  | Рис. 1.3. Диаграмма вариантов использования |

Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Модель может содержать 4 типа диаграмм [20, 21, 22, 23]:

* контекстную диаграмму;
* диаграммы декомпозиции;
* диаграммы дерева узлов;
* диаграммы только для экспозиции (for exposition only, FEO).

Контекстная диаграмма (диаграмма верхнего уровня), являясь вершиной древовидной структуры диаграмм, показывает назначение системы (основную функцию) и ее взаимодействие с внешней средой. В каждой модели может быть только одна контекстная диаграмма. После описания основной функции выполняется функциональная декомпозиция, т. е. определяются функции, из которых состоит основная.

Далее функции делятся на подфункции и так до достижения требуемого уровня детализации исследуемой системы. Диаграммы, которые описывают каждый такой фрагмент системы, называются диаграммами декомпозиции. После каждого сеанса декомпозиции проводятся сеансы экспертизы – эксперты предметной области указывают на соответствие реальных процессов созданным диаграммам. Найденные несоответствия устраняются, после чего приступают к дальнейшей детализации процессов.

Диаграмма дерева узлов показывает иерархическую зависимость функций (работ), но не связи между ними. Их может быть сколько угодно, поскольку дерево можно построить на произвольную глубину и с произвольного узла.

Диаграммы для экспозиции строятся для иллюстрации отдельных фрагментов модели с целью отображения альтернативной точки зрения на происходящие в системе процессы (например, с точки зрения руководства организации).

Функциональная модель системы тестирования представлена на рис 1.4-1.5.

Как видно из рис. 1.4 основным процессом для разрабатываемой модели является процесс «Тестирование».

Процесс тестирования разбивается на следующие этапы (рис. 1.5):

1. Выбор тестирования (адаптивное).
2. Выбор темы.
3. Прохождение тестирования.



Рис.1.4 Диаграмма первого уровня функциональной модели



Рис.1.5 Диаграмма второго уровня функциональной модели

## Постановка задачи дипломной работы

В настоящее время наличие доступа к интернету не является необычным, скорее наоборот – это норма. Поэтому важно использовать открывающиеся возможности применения интернет-ресурсов для реализации на практике возможностей дистанционного (при необходимости) или стационарного обучения.

***Цель*** – реализация алгоритма адаптивного тестирования в рамках учебно-методического комплекса «УМК-А» c помощью Web-технологий.

***Задачи:***

* Исследование предметной области;
* Разработка технического задания;
* организация работы сайта на основе базы данных и разграничения прав доступа к имеющейся информации. Доступ к ресурсам сайта возможен при регистрации или авторизации пользователя.

***Объект исследования*** – алгоритмы адаптивного тестирования, применимые в системах электронного образования.

***Предмет исследования*** – реализация алгоритма адаптивного тестирования в рамках учебно-методического комплекса «УМК-А».

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разрабатываемый модуль сайта относится к классу распределённых информационных систем. Разработка ведётся в соответствии с концепцией MVC.

## Введение

### Основание для разработки

Программное обеспечение разрабатывается в рамках дипломной работы. Основанием для разработки является задание на выполнение дипломной работы, выданное дипломным руководителем к.т.н. А.А. Азарченковым, на основе приказа и.о. БГТУ проф. \_Рудакова И.А.\_ № 284-3 от 19.04.2013.

### Назначение

Данное программное средство предназначено для автоматизации определенной деятельности кафедры любого ВУЗа. Рассмотрим данное средство на конкретном примере – кафедра «ИиПО» Брянского Государственного Технического Университета.

Данный программный продукт рассчитан на любой класс пользователей, так как для него рассчитан простой, интуитивно понятный интерфейс, для работы с данным программным продуктом не требуется никаких определенных навыков или дополнительных знаний.

### Определения, сокращения, аббревиатуры

ПО – программное обеспечение.

ПС – программное средство.

ОС – операционная система.

Web-сайт – совокупность web-страниц, объединенных некоторой общей идеей и связанных между собой гиперссылками.

Web-страница – один файл, содержащий код HTML, и имеющий расширение htm, html, php, phtml.

PHP – это язык, предназначенный для быстрого создания динамических web-страниц.

Адаптивное тестирование – диалоговая тестовая программа, которая изменяет порядок предъявления тестовых заданий в зависимости от результатов выполнения испытуемым уже предъявленных заданий.

## Полное описание

### Перспектива

Программное средство должно иметь клиент-серверную архитектуру, построенную на базе web технологий. Должна использовать бесплатную СУБД для возможности расширения функциональности не отраженной в данных ТЗ.

### Функция изделия

Функция обработки данных – обрабатывает ответы на вопросы определенного уровня, в зависимости от возвращаемого результата отбирает последующие задания.

Функция ввода данных – позволяет внесение новых данных и сохранение их в базу данных. Ввод данных при авторизации.

Функция просмотра предыдущих результатов тестирования – предназначена для возможности последующего анализа ранее пройденного тестирования.

### Характеристика пользователей

Пользователь должен уметь пользоваться любым известным браузером, уметь ориентироваться в стандартных элементах управления web страницами.

### Ограничения

Работа с браузерами Google Chrome 26, Mozilla Firefox 20.0.

## Специфические требования

### Требования к внешним интерфейсам.

Интерфейс пользователя. Реализовать графический интерфейс пользователя, представляющий собой страницу, содержащую поля для входа в систему. После авторизации пользователь имеет возможность перехода на страницу, в которую включен выбор дисциплины для прохождения тестирования, и соответственно прохождение самого тестирования. Есть возможность просмотра результатов ранее пройденных тестирований.

### Функциональные требования

Программа предоставляет доступ студенту со следующими возможностями:

* авторизация;
* доступ к базе данных;
* просмотр информации о предполагаемых тестах;
* выбор дисциплины;
* прохождение тестирования выбранной дисциплины;
* просмотр результатов ранее пройденных тестирований;
* составляющий «каркас» тестирования (время тестирования, количество вопросов различных уровней и т.п.).

Администратор имеет права доступа ко всем возможностям сайта. Он регистрирует пользователей. Имеет права доступа к внесению и редактированию информации, составляющей «каркас» тестирования, а также информацию о предполагаемых тестах. Администратор может просматривать результаты ранее пройденных тестирований.

«Каркас» тестирования включает в себя вопросы низкого уровня сложности, вопросы среднего уровня сложности, вопросы высокого уровня сложности, время прохождения тестирования, количество всех вопросов на определенном уровне.

Результаты ранее пройденных тестирований содержат ФИО тестируемого, дату прохождения тестирования и результат.

Профиль каждого студента содержит информацию о дате и результатах пройденных ранее тестированиях. В зависимости от предыдущих результатов (среднего балла), тестирование начинается с соответствующего уровня. При первичном прохождении теста, вопросы начинаются со среднего уровня.

### Требования к рабочим характеристикам

Число одновременно поддерживаемых пользователей не ограничено, однако вход с одного ПК разным пользователям не возможен. Также не возможен вход пользователя, который уже авторизован на данный момент на другом ПК.

## Этапы разработки

Программное обеспечение модуля системы должно разрабатваться в следующем порядке:

* анализ предметной области;
* разработка технического задания;
* проектирование системы;
* реализация системы;
* тестирование законченной программы.

## Порядок контроля и приемки

Для всестороннего контроля работы системы необходимо разработать специальные наборы тестовых данных, результаты обработки которых в полной мере отразят работоспособность системы. Для проверки правильности работы программы должно быть проведено тестирование всех режимов работы.

# Исследовательская часть

## Исследование алгоритмов адаптивного тестирования

Анализ алгоритмов адаптивного тестирования позволяет выделить следующие варианты.

1. Алгоритм тестирования «жесткий».
2. Алгоритм тестирования «мягкий».
3. Комбинированный алгоритм.
4. Динамическая модель адаптивного тестирования.
5. Статистическая модель Раша (Rasch).

### Жесткий алгоритм тестирования

«Жесткий» алгоритм тестирования основывается на формировании области незнания, т.е. вопросы будут задаваться по темам, которые пользователь не знает, или знает плохо. Таким образом, пользователь наказывается за незнание какой-либо темы. Вначале в области незнания находятся все темы теста. По мере прохождения опроса, область может динамически меняться:

* если пользователь отвечает на вопрос правильно, то тема, к которой принадлежит вопрос, удаляется из области;
* если пользователь отвечает на вопрос неправильно или частично правильно, то тема остается в области;
* если область пуста, то она заполняется снова всеми темами теста.

Если вопросы из области незнания закончились, то система задает вопросы из других тем.

Тестирование прекращается в одном из следующих случаев:

* пройден максимальный уровень сложности;
* пользователь не сумел преодолеть минимальный уровень сложности;
* истекло время на прохождение теста или преодоление уровня сложности;
* исчерпан лимит вопросов.

Достоинства такого подхода следующие: при данном подходе полностью исключается удача при выборе очередного вопроса. Таким образом, пользователю недостаточно знать несколько тем, чтобы хорошо пройти тест, так как в случае верного ответа система начнет задавать вопросы из других тем, которые пользователь может не знать. А если он не знает какой-либо темы, то он может застрять на ней надолго и испортить себе оценку неправильными ответами. Недостатком такого подхода является то, что если пользователь знает предмет хорошо, но несколько тем знает плохо, то система, выяснив это, будет «валить» пользователя, и он не сможет получить хорошую оценку. Ситуация усугубляется тем, что если пользователю сразу попался вопрос из темы, которую он плохо знает, то он может сразу получить плохую оценку, даже если знает в совершенстве все остальные темы.

### Мягкий алгоритм тестирования

«Мягкий» алгоритм тестирования основывается на формировании области знания, т.е. вопросы будут задаваться по темам, которые пользователь знает хорошо. Таким образом, пользователь не наказывается за незнание какой-либо темы.

Вначале в области знания находятся все темы теста. По мере прохождения опроса, область может динамически меняться:

* если пользователь отвечает на вопрос неправильно, то тема, к которой принадлежит вопрос, удаляется из области;
* если пользователь отвечает на вопрос правильно или частично правильно, то тема остается в области;
* если область пуста, то она заполняется снова всеми темами теста.

Если вопросы из области знания закончились, то система задает вопросы из других тем.

Прекращение тестирования и выставление оценки осуществляется аналогично предыдущему алгоритму.

Достоинства такого подхода следующие: при данном подходе полностью исключается возможность застрять на какой-нибудь теме и, следовательно, испортить себе оценку неправильными ответами.

Недостатком такого подхода является то, что если пользователь знает предмет плохо или знает всего несколько тем, то система будет «вытягивать» пользователя и у него всегда есть шанс получить хорошую оценку, даже если он ее не заслуживает. Ситуация усугубляется тем, что если пользователю сразу попался вопрос из темы, которую знает, то он может сразу получить хорошую оценку, даже если вообще не знает все остальные темы.

### Комбинированный алгоритм

Комбинированный алгоритм сочетает в себе достоинства предыдущих алгоритмов. Пользователь не наказывается за незнание какой-либо темы, но и не поощряется чрезмерно. В данном алгоритме нет области знания и незнания – есть просто область тем, из которых выбираются вопросы.

Вначале в области находятся все темы теста. По мере прохождения опроса, область может динамически меняться:

* если пользователь отвечает на вопрос неправильно и у него мало баллов, то тема, к которой принадлежит вопрос, удаляется из области;
* если пользователь отвечает на вопрос правильно и у него много баллов, то тема, к которой принадлежит вопрос, также удаляется из области;
* если область пуста, то она заполняется снова всеми темами теста.

Если вопросы из области тем закончились, то система задает вопросы из других тем.

Прекращение тестирования и выставление оценки осуществляется аналогично предыдущему алгоритму.

Таким образом, пользователь не штрафуется, если у него мало баллов, но и не поощряется, если баллов много.

### Динамическая модель адаптивного тестирования

Разработана модель адаптивного тестирования знаний, ориентированная не только на определение уровня подготовленности пользователей, но и позволяющая динамически изменять число вопросов в зависимости от числа их уровней трудности, при сохранении надежности получаемых результатов не ниже чем при линейном тестировании.

Эффективное управление тестированием знаний предполагает, что для пользователей с различным уровнем подготовленности будут предложены вопросы различного уровня трудности, т.е. будет проведено адаптивное тестирование. При адаптивном тестировании каждый из пользователей получает свое количество вопросов и выходит на определенный уровень трудности, после чего тестирование заканчивается, а результаты переносятся на метрическую шкалу.

Адаптивное тестирование позволяет решить две задачи: уменьшение времени тестирования и уменьшение загрузки каналов связи и компьютерного оборудования.

Суть предлагаемой модели адаптивного тестирования заключается в следующем: пусть имеется -уровней трудности вопросов (число вопросов на каждом уровне имеет значение только для уточнения уровня знаний). Разобьем -уровней на отрезков по уровней трудности вопросов на каждом (с округлением до целого).

Проведем верхнюю оценку минимального числа вопросов, которое потребуется для определения уровня знаний условного пользователя. Самое большое количество вопросов (при старте с самого нижнего уровня) должно быть задано пользователю, подготовленность которого находится вблизи самого большого уровня трудности вопросов (на последнем отрезке). Для того чтобы дойти до последнего отрезка, потребуется задать *x*-вопросов (по числу отрезков) и не более чем – вопросов, чтобы уточнить его уровень на последнем отрезке. Таким образом, функция *f(x) –* числа вопросов будет иметь вид: . Данная функция имеет минимум при , а максимальное число вопросов будет не более чем,.

На рис.3.1. представлена графическая схема алгоритма уточнения уровня подготовленности пользователя после первого неверного ответа (точка А), – отрезок номер *k*, – отрезок номер *k+1*. Стрелками над линией указаны переходы после правильных ответов, под линией – при ошибочных.



Рис.3.1. Графическая схема алгоритма уточнения   
уровня подготовленности пользователя

Для проверки эффективности использования предлагаемого алгоритма адаптивного тестирования были проведены эксперименты, 75 студентам был предложен тест состоящий из 36 заданий. При проведении эксперимента студентам было предложено пройти данный тест, как в линейной, так и в адаптивной форме.

Расчет коэффициента корреляции Пирсона дает величину *r*=0,88, это говорит о том, что между баллами, полученными при линейном тестировании и адаптивном существует сильная корреляционная связь, а значит, результаты, полученные при использовании адаптивного тестирования являются не менее валидными, чем обычного линейного.

Отметим, что при адаптивном тестировании каждому испытуемому задается в среднем 9,2 заданий против 36 при линейном, что не превышает верхней оценки .



### Статистическая модель Раша (Rasch)

Основной математической моделью IRT является однопараметрическая логистическая функция Раша, называемая характеристической кривой, имеющая вид:

,

где – вероятность правильного ответа испытуемых любого уровня подготовленности на задание определённого уровня трудности под номером j; – уровень трудности конкретного, j-го задания проектируемого теста; – уровень знаний, латентная переменная.

Чем выше крутизна функции P, тем уже интервал, на котором это задание работает. Таким образом, возникла мысль об улучшении модели Раша за счет введения в выражение второго параметра . Параметр даёт информацию о задании с точки зрения оценки его дифференцирующей способности, на заданном интервале. Геометрически значение параметра выражается крутизной характеристической кривой, аналитически – значением производной функции в точке перегиба. После введения в выражение параметра получается двухпараметрическая модель:

.

Эмпирические пределы значений для параметра – от минус 2,80 до плюс 2,80. Возможно также добавление параметра C, определяющего вероятность угадывания правильного ответа.

Возникает проблема адекватной оценки трудности тестовых заданий и начального уровня знаний студентов. Начальная оценка уровня знаний испытуемого определяется по формуле

,

где – доля правильных ответов i-го испытуемого, – доля неправильных ответов. Аналогично определяется начальная оценка уровня трудности задания теста

,

где – доля правильных ответов на j-е задание теста, – доля неправильных ответов.

В силу действия различных случайных факторов оценки параметров θ и , полученные на нескольких выборках, будут, конечно, различаться. Если объем выборки достаточно велик, то можно ставить вопрос о вычислении устойчивых значений параметров θ и δ, которые будут наиболее эффективными оценками и могут быть приняты в качестве объективных оценок параметров и δ .

При любом проведении процесса тестирования результаты вычисления – статистических оценок , и – статистических оценок будут отличаться от существующих точных значений. По своему смыслу оценки являются определенными функциями исходных случайных значений элементов матрицы ответов , состоящей из N-строк и K-столбцов и поэтому сами являются случайными величинами. Таким образом, возникает вопрос о нахождении математических ожиданий и дисперсий этих случайных величин. Необходимо чтобы математическое ожидание соответствующих оценок совпадало с соответствующими точными значениями, а дисперсия оценки была бы минимальной. Статистическая оценка уровня подготовленности и уровня трудности будут являться несмещенными оценками, если их математическое ожидание при любом объеме выборки испытуемых будет равно самому оцениваемому параметру. На практике обычно используют асимптотически несмещенную оценку, математическое ожидание которой стремится к истинному значению оцениваемого параметра, при неограниченном увеличении объема выборки. Статистическая оценка эффективна если при заданной выборке, она имеет возможную наименьшую дисперсию D\* при неполной информации, возможно, получить лишь оценку с D>D\*. Если отношение D/D\*→1, при увеличении выборки, то оценка называется асимптотически эффективной. Статистическая оценка состоятельна, если несмещенная оценка не является эффективной, но при увеличении объема выборки ее дисперсия уменьшается. Несмещенность, эффективность и состоятельность являются независимыми свойствами, характеризующими оценки с разных сторон. Задача отыскания эффективных несмещенных оценок имеет особо важное значение при обработке результатов малых выборок испытуемых. Для получения оценки параметров δ и θ применяются метод моментов или метод наибольшего правдоподобия. Рассмотрим подробнее второй метод.

## Разработка собственного алгоритма тестирования

На основе изученных алгоритмов, основанных на адаптивном тестировании, мною был разработан собственный алгоритм, который будет использоваться в системе «УМК-А».

Для начала прохождения тестирования необходимо авторизоваться, выбрать дисциплину и тему, по которой будут тестироваться знания студентов. Тест начинается с некоторого уровня сложности, который определяется путем средней оценки от предыдущих результатов. В случае, первичного прохождения теста, вопросы начинаются со среднего уровня. Если средней оценкой является 3, то тестирование начинается с низкого уровня. Если же средняя оценка – 4 или 5, то тестирование начинается со среднего уровня сложности. Помимо сложности, каждый вопрос имеет свое время. Если человек, не отвечает в течении указанного времени, система прекращает работу и выводит на экран результаты.

На каждом уровне испытуемому задается последовательно 3 вопроса, возможности вернуться к предыдущему заданию – нет. Для каждого уровня разрешено только две попытки, т.е. на определенном уровне сложности, студент может ответить на три вопроса дважды, в противном случае тестирование заканчивается и выводится результат. Если испытуемый прошел средний уровень сложности с первой попытки и перешел на высокий уровень, который он не смог пройти, то при возвращении на средний уровень у студента количество попыток не считается заново, т.е. у него остается только одна попытка. В данной системе имеется особенность: со среднего уровня на низкий, человек может вернуться, если тестирование начинал со среднего уровня сложности. Начиная тестирования с низкого уровня, такого перехода не имеется, т.к. человек перешел на средний, значит низкий уровень он уже прошел и минимальные знания у него имеются. С высокого уровня на средний перейти можно, здесь устанавливается граница – насколько хорошо он усвоил знания.

Начиная тестирование с низкого уровня, студент отвечает на 3 вопроса. Если испытуемый ответил правильно на все вопросы с первой попытки, то система выдает вопросы высокого уровня. Если же количество правильных ответов равно двум, то система задает вопросы среднего уровня сложности. В противном случае, испытуемому дается вторая попытка на низком уровне. Если испытуемый отвечает на все вопросы, но со второй попытки, то переходит на средний уровень сложности.

При прохождении тестирования на среднем уровне сложности, не имеет значения, начинается тестирование на основе средней оценки или же студент перешел с низкого уровня. Если испытуемый успешно ответил на 2-3 вопроса, то ему задаются вопросы высокого уровня сложности. В противном случае, у студента имеется еще одна попытка и система выдает ему еще три вопроса сложного уровня.

Прохождение тестирования на высоком уровне заканчивается, если студент ответил на все вопросы верно. Если нет, то система задает вопросы среднего уровня.

На рис. 3.2. представлен раскрытый алгоритм адаптивного тестирования в системе «УМК-А».



Рис. 3.2. Алгоритм адаптивного тестирования

Результат тестирования зависит от полностью пройденного уровня сложности. Если тест был завершен

* на низком уровне, то оценка 2;
* на среднем уровне, т.е. с низкого уровня испытуемый перешел на средний, но дальше не продвинулся, то оценка 3;
* на высоком уровне, в случае малого количества правильных ответов в первой попытке, то оценка 4;
* на высоком уровне, в случае всех правильных ответов, то оценка 5.

При использовании двух попыток, система высчитывает, вопросы какой сложности задавать дальше, на основе количества правильных ответов данной серии заданий, а не в зависимости от общего числа заданных вопросов.

Существует вероятность, что в базе данных будет недостаточное количество вопросов для двух попыток. В данном случае, система выдаст сообщение, что тестирование не может быть начато.

## Выводы

В исследовании приведены некоторые из существующих алгоритмов адаптивного тестирования. Многие из них имеют в своей основе математический аппарат, однако рассматривая их более детально можно получить вывод, что все они построены на ряде умозаключений конкретных авторов и не способны отобразить объективной информации. В связи с тем, что сегодня не существует объективных методов оценивания знаний на основе адаптивных тестов, целесообразно разработать собственный алгоритм. Преимущество, которого заключается в гибкости управления его работой, четким представлением его работы. Кроме этого собственный алгоритм может в будущем получить существенное развитие за счет накопления статистической информации, в отличии от существующих, которые уже используют какие-то статистические данные.

# Конструкторская часть

## Выбор средств разработки

### Выбор языка программирования

PHP — скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект не является свободным и распространяется под собственной лицензией [16].

Область применения PHP [16] сфокусирована на написание скриптов, работающих на стороне сервера. таким образом, PHP способен выполнять всё то, что выполняет любая другая программа CGI. Например, обрабатывать данных форм, генерировать динамические страницы. Но PHP способен выполнять и множество других задач.

Перечислим некоторые преимущества этого языка:

* PHP способен выдавать любые текстовые данные, такие, как XHTML и другие XML – файлы;
* PHP способен осуществлять автоматическую генерацию файлов и сохранять их в файловой системе сервера, вместо того, чтобы отдавать клиенту, организуя, таким образом, кэш динамического содержания, расположенный на стороне сервера;
* поддержку широкого круга баз данных и операционных систем, а так же большого числа веб – серверов;
* у интерпретатора упрощенно обнаружение ошибок во время выполнения программы. В случае сбоя интерпретатор сразу же выведет сообщение, если что-то не так;
* можно не заботиться об освобождении выделенной памяти.

Язык JAVA – объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems.

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее.

Достоинство подобного способа выполнения программ – в полной независимости байт – кода от операционной системы и оборудования, что позволяет выполнять Java – приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

К недостаткам концепции виртуальной машины относят то, что исполнение байт–кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java.

Основные возможности языка:

* Автоматическое управление памятью.
* Расширенные возможности обработки исключительных ситуаций.
* Богатый набор средств фильтрации ввода/вывода.
* Набор стандартных коллекций, таких как массив, список, и т. п..
* Наличие простых средств создания сетевых приложений (в том числе с использованием протокола RMI).
* Наличие классов, позволяющих выполнять HTTP-запросы и обрабатывать ответы.

Язык VBScript – cкриптовый язык программирования, интерпретируемый компонентом Windows Script Host. Он широко используется при создании скриптов в операционных системах семейства Microsoft Windows.

Скрипты на языке VBScript чаще всего используются в следующих областях, использующих продукцию Microsoft:

* автоматизация администрирования систем Windows;
* серверный программный код в страницах ASP;
* клиентские скрипты в браузере Internet Explorer.

Такой тип сценариев обычно используется для:

* создания сложных сценариев;
* использования объектов из других приложений и библиотек;
* скрытия окон в ходе выполнения сценария;
* шифрования логики сценария.

В основном VBS – сценарии применяются для обработки данных, управления системой, работы с учетными записями пользователей и компьютеров, взаимодействия с офисными приложениями, работы с базами данными и прочих сложных задач.

VBS – сценарий – это обычный текстовый файл с именем \*.VBS, который легко править в блокноте, а запускать на исполнение – двойным щелчком мыши или вызовом по имени в консоли.

Cценарии не компилируются, а интерпретируются. То есть для обработки скрипта в системе должен присутствовать интерпретатор языка VBS, и таких интерпретатора в Windows есть даже две штуки: оконный WScript и консольный CScript – оба интерпретатора – это Windows Script Host (WSH).

После сравнения характеристик, достоинств и недостатков был выбран язык PHP, из-за следующих преимуществ:

* Наличие понятной и доступной документации
* Простота в установке и настройке.
* Поддержка БД – MySQL.
* Поддержка популярными веб-серверами Apache и nginx.
* Свобода в выборе хостинга, т.к. подавляющее большинство хостингов используют платформу FreeBSD + Apache + PHP + MySQL.

### Выбор СУБД

Система управления базами данных (СУБД) — специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных. Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор.

На сегодняшний момент существует довольно приличное количество баз данных – это и MS SQL Server, и Interbase, и Oracle с его колоссальными возможностями и Sybase и Postgres и еще большой ряд других СУБД, но для web-ориентированных приложений все таки лучше остановиться на MySQL – как на более простой и в, то же время достаточно функциональной базе.

Microsoft SQL Server – система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для от небольших и средних по размеру баз данных до крупных баз данных масштаба предприятия, конкурирует с другими СУБД в этом сегменте рынка.

В SQL Server 2008 встроена поддержка .NET Framework. Благодаря этому, хранимые процедуры БД могут быть написаны на любом языке платформы .NET, используя полный набор библиотек, доступных для .NET Framework, включая Common Type System (система обращения с типами данных в Microsoft .NET Framework).

Ограничения:

* 1 поддерживаемый процессор (но может быть установлен на любой сервер).
* 1 Гб адресуемой памяти.
* 4 Гб максимальный размер базы.
* Через интерфейс нет возможности экспорта/импорта данных.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы [17].

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

В качестве СУБД была выбрана MySQL, так как является популярным средством в сети Интернет [18]. MySQL является наиболее производительной, разработана для работы с web-серверами, имеет более простой интерфейс при создании и работе с базами данных.

### Выбор среды разработки

PHP Expert Editor - удобный редактор для PHP, Perl, Python, HTML, Java Script и других файлов с поддержкой UTF-8. Программа разработана специально для PHP мастеров и имеет встроенный отладчик PHP. PHP Expert Editor имеет встроенный HTTP сервер и позволяет запускать PHP, Perl, Python, скрипты. Есть возможность использовать любой внешний HTTP сервер.

Преимущества PHP Expert Editor:

* Встроенный браузер;
* Встроенный FTP-клиент с поддержкой SFTP;
* Авто сохранение;
* Проверка синтаксиса PHP;
* Запуск скриптов и просмотр результата во встроенном или внешнем браузере;
* Отладчик;
* Для запуска и отладки PHP скриптов можно использовать встроенный или любой внешний HTTP-сервер;
* Поддержка всех известных Content-Type. Возможность отлаживать скрипты, которые генерируют различный контент, например, картинки;
* Быстрая вставка всех функций PHP с подсказкой параметров;
* Настраиваемые Шаблоны Кода для быстрой вставки часто употребляемых фрагментов;
* Экспорт исходного текста в HTML и RTF с подсветкой синтаксиса;
* Поддержка справки PHP с возможностью поиска по ключевому слову в текущей позиции;
* Отображение ошибок и предупреждений в соответствии с настройками PHP
* Поддержка Perl, Python, Ruby, Tcl. Есть возможность использовать несколько интерпретаторов, не только PHP.

Macromedia Dreamweaver (сейчас - Adobe Dreamweaver) - профессиональный редактор HTML для проектирования, написания кода и поддержки сайтов, web-страниц и приложений сети. Практикуете ли вы написание кода HTML вручную или предпочитаете работать в визуальной среде редактирования, Dreamweaver обеспечивает вас полезными инструментальными средствами, чтобы увеличить ваш опыт создания web-страниц и сайтов.

Визуальные возможности редактирования в Dreamweaver позволяют вам быстро создавать страницы без того, чтобы писать код вручную. Вы можете просматривать все ваши элементы сайта или активы и быстро вставлять их из панели непосредственно в документ. Вы можете упрощать ваши действия, создавая и редактируя изображения в Macromedia Fireworks или в другом графическом редакторе, затем импортируя их непосредственно в Dreamweaver или добавляя объекты Macromedia Flash.

Dreamweaver также обеспечивает полно-функциональную среду написания кода, которая включает инструментальные средства редактирования кода (например, раскраска кода или проверка закрытия тегов) и справочная информация по HTML, CSS, JavaScript, язык разметки ColdFusion (CFML), Microsoft Active Server Pages (ASP) и страницы JavaServer (JSP).

Zend Studio - профессиональное средство для разработки приложений на PHP, позволящее производить полный цикл работ над проектом:

* создание;
* редактирование;
* отладку;
* анализ;
* оптимизацию.

Пакет Zend Studio включает в себя все необходимые инструменты, включая HTTP/PHP сервер с поддержкой расширений.

Среда разработки Zend (ZDE) – это мощная интегрированная платформа для создания и поддержки приложений PHP. В состав ZDE входит полноценный редактор и ведущее в отрасли средство для обработки программного кода, а также интегрированный отладчик последнего поколения. Эти инструменты заметно сокращают сроки разработки приложений и позволяют гораздо быстрее выпустить продукт на рынок.

Существует огромное количество программ для разработки сайтов. Из всех существующих сред для написания дипломной работы была выбрана **PHP Expert Editor**, так как она позволяет запускать PHP, Perl, Python, скрипты. А также использовать любой внешний HTTP сервер. Присутствует проверка синтаксиса PHP, встроенный браузер, FTP клиент с поддержкой SFTP, поддержка проектов, настраиваемые шаблоны кода, настраиваемая подсветка кода, и многие другие функции значительно облегчают работу разработчика.

### Выбор web-сервера

Веб-сервер — это сервер, обслуживающий запросы к одному или нескольким сайтам Всемирной паутины (веб-сайтам). Клиент, в роли которого обычно выступает веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы — это HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP. Веб-сервером называют как программное обеспечение, выполняющее эти функции, так и непосредственно компьютер, на котором это программное обеспечение работает.

**Apache** является кроссплатформенным ПО, поддерживая операционные системы GNU/Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т.д. Поддерживает IPv6.

Ядро Apache включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей. Ядро (в отличие от модулей) полностью разрабатывается Apache Software Foundation, без участия сторонних программистов.

Существует множество модулей, добавляющих к Apache поддержку различных языков программирования и систем разработки. К ним относятся:

* PHP (mod\_php).
* Python (mod\_python).
* Ruby (apache-ruby).
* Perl (mod\_perl).
* ASP (apache-asp).

Принимая во внимание особенности серверов и специфику задачи, для реализации дипломного проектирования был выбран **Wampserver** – это программный пакет для создания полнофункционального локального веб-сервера, помогающего в разработке веб-сайтов, включающий в себя основные инструменты веб-разработчика (Apache, MySQL, интерпретатор скриптов PHP, phpMyAdmin) и другие дополнения. Также включает в себя панель управления, которая обеспечивает удобное управление серверами и предоставляет быстрый доступ к Phpmyadmin и SQLitemanager.

К преимуществам данного сервера можно отнести:

* свежие версии софта;
* высокая стабильность работы;
* небольшой размер установочного файла;
* легкая установка;
* возможность вынести базы данных, конфигурационные файлы на отдельный диск, или любую другую папку извне;
* включение модулей, настроек и расширений php, apache одним кликом мышки.

Базовый комплект:

* Apache 2.2.21
* Php 5.3.10
* Mysql 5.5.20
* XDebug 2.1.2
* XDC 1.5
* PhpMyadmin 3.4.10.1
* SQLBuddy 1.3.3
* webGrind 1.0

## Архитектура модуля адаптивного тестирования

Программная система построена на основе архитектуры клиент-сервер. Как правило, компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам.

Компьютер, управляющий ресурсом, называют сервером этого ресурса.

Клиентом называется программа, выполняющаяся на конечной системе (компьютере), основной функцией которой является формирование запросов и получение результатов обслуживания от программы сервера, расположенной на другой оконечной системе.

Клиент и сервер какого-либо ресурса могут находиться как в рамках одной вычислительной системы, так и на различных компьютерах, связанных сетью. Взаимодействие клиентской и серверной частей приложения осуществляется через сеть - локальную или глобальную. Доступ к сервису осуществляется с помощью браузера.

Клиентская часть системы обращается по сети к серверной части. Клиентские задачи выполняет браузер, а серверные – web-сервер. Запросы пользователя направляются на сервер, по полученным результатам web-сервер формирует ответ и направляет его в браузер клиента.

Разрабатываемая программная система, согласно функциональным требованиям, имеет архитектуру клиент серверного приложения с доступом к серверной БД, и состоит из следующих подсистем.

* **Модуль управления информацией** – в зависимости от прав доступа, предоставляет различную информацию.
* **Модуль администратора** – помимо базовых функций, реализует возможности добавления новые вопросов в базу.
* **Модуль студента** – предоставление возможности прохождения тестирования.
* **Модуль оповещения пользователя** – модуль, отвечающий за вывод информации о результатах и ранее пройденных тестированиях.
* **Модуль формирования тестирования –** модуль, отвечающий за организацию нового тестирования.
* **Модуль обработки данных –** модуль, отвечающий, за обработку ответов на вопросы определенного уровня.
* **Модуль интерфейс –** модуль позволяет осуществлять «диалог» пользователя и программы.

Пользователь заходит на сайт через модуль интерфейса, где пользователю предоставляется возможность пройти авторизацию для дальнейшей работы.

После того как пользователь авторизовался на сайте, ему присваивается определенный статус, в соответствии с которым определяются его права на доступ к тому или иному разделу.

Администратор имеет полный доступ ко всему содержимому сайта.

На рис. 4.1 изображена общая архитектура сайта. Модуль работы с БД позволит администратору делать резервные копии базы данных и восстанавливать из них, осуществлять изменения базы данных вручную.

|  |  |
| --- | --- |
|  | Рис 4.1 Общая архитектура сайта. |

## Моделирование работы автоматизированной системы

### Модель размещения

Диаграммы размещения, или развертывания, – это один из двух видов диаграмм, используемых при моделировании физических аспектов объектно-ориентированной системы (рис. 4.2.)

Такая диаграмма показывает конфигурацию узлов, где производится обработка информации, и то, какие компоненты размещены на каждом узле.

Диаграммы размещения используются для моделирования статического вида системы с точки зрения развертывания. В основном под этим понимается моделирование топологии аппаратных средств, на которых выполняется система. По существу, диаграммы развертывания – это просто диаграммы классов, сосредоточенные на системных узлах.

Диаграммы развертывания важны не только для визуализации, специфицирования и документирования встроенных, клиент-серверных и распределенных систем, но и для управления исполняемыми системами с использованием прямого и обратного проектирования.



Рис. 4.2. Диаграмма размещения

### Модель хранения данных

В качестве способа моделирования данных были выбраны диаграммы «сущность-связь».

Логический уровень модели данных представлен на рис.3.2.

Модель данных разработана на основе технического задания, она включает в себя таблицы, необходимые для хранения информации и заемщике. Основными являются таблицы:

* Disciplines. В данной таблице содержится информация о дисциплинах, по которым будет проходить тестирование.
* Group. Содержит информацию о группе, ее полном и коротком названии. К какой кафедре принадлежит группа, какие дисциплины изучает.



Рис.3.2. Модель данных. Логический уровен

* Tems. Содержит информацию о темах, изучаемых в рамках какой-либо дисциплины.
* Teacher. Включает полную информацию о преподавателе кафедры, его должность. Его адрес почты и пароль для аутентификации и телефон.
* Work. В данной таблице представлена информация о работах, которые внесены в план дисциплины.
* Answer. Включает в себя ответы на вопросы все уровней сложности.
* Student. Таблица содержит полную информацию о студенте, номер в группе, его адрес почты и пароль для аутентификации.
* Task. Это таблица, содержащая задания, необходимые для тестирования.

Остальные представленные таблицы являются более подробно рассмотренными таблицами Work, Task и Teacher. Алгоритмическое конструирование модуля социальной сети.

### Модель потоков данных

При проектировании системы был применен структурный подход [8]. Сущность структурного подхода состоит в декомпозиции программного средства на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны.

В рамках структурного подхода к проектированию системы в качестве модели, отражающей функциональность системы, было выбрано построение диаграмм потоков данных (DFD) [9]. Диаграммы DFD позволяют описать процесс обмена информацией между элементами изучаемой системы. DFD отображает источники и адресаты данных, идентифицирует процессы и группы данных, связывающие в потоки одну функцию с другой, а также, что важно, определяет накопители (хранилища) данных, которые используются в исследуемом процессе.

Модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю.

Диаграммы верхних уровней иерархии определяют основные процессы или подсистемы с внешними входами и выходами. Они детализируются с помощью диаграмм нижнего уровня. Такая детализация продолжается, создавая многоуровневую иерархию диаграмм до тех пор, пока не будет достигнут уровень декомпозиции, на котором процессы становятся элементами и детализировать их не имеет смысла.

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям информации.

Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, в нашем случае – это преподаватель, оператор, которые инициируют процесс создания тестовых заданий.

Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. В нашем случае процессами являются: подготовка материалов, разбиение материалов на вопросы, ответы и уровни сложности, а так же работа с программой.

Накопитель данных представляет собой базу данных, где хранятся созданные вопросы, ответы к ним и уровень сложности, выставляемый к ответам и результаты ранее пройденных тестов.

Контекстная модель потоков данных представлена на рис. 4.3. Из диаграммы видно, что у системы имеются три вида пользователей: студент, преподаватель и преподаватель-администратор. Преподаватель-администратор отвечает за регистрацию пользователей. Пользователь Преподаватель участвует в процессе занесения в систему необходимых материалов, составляет и редактирует задачи и ответы к ним, а так же получает от системы информацию о результатах теста всех студентов в различных учебных группах.

На этой модели видно, что для того, чтобы попасть в систему студент должен авторизоваться. После прохождения авторизации студент может посмотреть результаты прошлых тестов или пройти новое тестирование. Вопросы для тестирования подготавливает и редактирует преподаватель, после чего сохраняет их в базе данных.



Рис. 4.3. Контекстная модель потоков данных



Рис. 4.4. Второй уровень модели

## Проектирование пользовательского интерфейса

Вопрос пользовательских интерфейсов стал играть важную роль при разработке программного обеспечения только тогда, когда прямой доступ к компьютерам получил кто-то еще, помимо программистов и операторов. Современная концепция пользовательских интерфейсов появилась только тогда, когда появились терминалы, напрямую или косвенно соединенные с компьютерами. По мере роста числа людей, которые могли напрямую взаимодействовать с компьютерами, человеко-машинный интерфейс становился все более значимой задачей для разработчиков и проектировщиков программных средств. Если вначале решались в основном проблемы технологий (пользовательские интерфейсы), то потом стали решаться проблемы людей (пользователей).

Чтобы интерфейс программы считался правильным, должны выполняться следующие условия:

* Доступность. Интерфейс программы подготовки адаптивного тестирования можно назвать доступным и интуитивно понятным, так как все поля для заполнения подписаны и расположены в порядке, в котором мы чаще всего задаем вопрос (вопрос, ответ к нему)
* Эффективность. Наряду с доступностью программа не перегружена дополнительными формами, которые не будут мешать опытному пользователю постоянно использовать программу
* Контекст. Система должна удовлетворяет существующим реалиям и текущей среде эксплуатационного контекста, внутри которого она будет разворачиваться и применяться.

Классы для представления модуля адаптивного тестирования. На рис.3.3. представлена диаграмма классов для представления модуля адаптивного тестирования. Ниже представлена их спецификация.



Рис. 3.2. Диаграмма классов модуля адаптивного тестирования

Класс Группа – необходим для хранения информации о группах, имеющих возможность пройти тестирование. В качестве член-данных выступают: номер группы. Методами являются: выбор группы.

Класс Студент – необходим для хранения информации о студентах конкретной группы, имеющих возможность пройти тестирование. В качестве член-данных выступают: ФИО, группа, логин, пароль, результат. Методами являются: регистрация, авторизация, восстановление пароля, выбор ФИО, выбор группы, показать результаты.

Класс Дисциплина – необходим для выбора тестирования по конкретной дисциплине, изучаемой в рамках указанной ранее группы. В качестве член-данных выступают: наименование. Методами являются: выбор дисциплины.

Класс Оценка – необходим для хранения информации о результатах пройденного тестирования. В качестве член-данных выступают: результат. Методами являются: показать результат.

Класс Обработчик – служит для анализа поступающих данных и формировании дальнейшего перечня вопросов. В качестве член-данных выступают: количество ответов, количество правильных ответов, время max. Методами являются: обработка ответов.

Класс Тест – содержит набор предлагаемых тем тестов. В качестве член-данных выступают: тема, вопросы max, время max. Методами являются: выбор темы.

Класс Информация о тесте – содержит информацию, о пройденных ранее тестированиях. В качестве член-данных выступают: ФИО тестируемого, дата, результат. Методами являются: вывод информации.

Класс Задания – служит для доступа к вопросам различных уровней сложности. В качестве член-данных выступают: вопросы нижнего уровня сложности, вопросы среднего уровня сложности, вопросы высокого уровня сложности, сложность вопроса. Методами являются: выбор вопросов нижнего уровня сложности, выбор вопросов среднего уровня сложности, выбор вопросов высокого уровня сложности.

Класс Администратор – служит для обеспечения бесперебойной работы администратора сайта. В качестве член-данных выступают: логин, пароль. Методами являются: регистрация, авторизация, восстановление пароля.

Класс Интерфейс – класс, хранит форму, посредством которой осуществляется визуализация программы. В качестве член-данных выступают: форма. Методами являются: передача данных, вывод сообщения.

## Алгоритм работы проектируемой системы

На рис. 3.4 представлена алгоритмическая модель модуля адаптивного тестирования.



Рис.4.5. Алгоритмическая модель программной системы

В этой модели видно, что работа сайта состоит из нескольких этапов. Количество процессов может изменяться, что позволяет легко модифицировать модульную структуру без вмешательств в алгоритм работы основного сайта.

При авторизации пользователю предоставляется выбор: прохождение тестирования или просмотр результатов, ранее пройденных тестов.

На рисунке видно, что для начала тестирования необходимо выбрать раздел тестирования, дисциплину и тему, по которой будет проходить тестирование. Необходимо выбрать тип тестирования – адаптивное. После окончания тестирования, необходимая информация заносится в БД, после чего на экран выводится результат данного тестирования.

При выборе просмотра результатов, пользователю предоставляется вся информация о ранее пройденных им тестах: дата прохождения и результат. Кроме этого, пользователь может увидеть свою среднюю оценку. После просмотра, студент может либо выйти из системы, либо пройти новое тестирование или пересдать старое.

# Экспериментальная часть

## Назначение тестирования

Тестирование – это процесс, который позволяет оценить качество производимого продукта. Качественный программный продукт должен отвечать предъявленным к нему требованиям. Тестирование представляет собой процесс проверки разрабатываемого продукта на соответствие этим требованиям.

Оно направлено на выявление дефектов и на оценку свойств разрабатываемой системы. Целью тестирования в нашем случае является оценка качества разрабатываемой автоматизированной системы.

## Тестирование в нормальных условиях

Тестирование программного комплекса в нормальных условиях проводилось с целью определения его соответствия требованиям технического задания и подтверждения корректности работы программы в характерных ситуациях. Проверка на соответствие техническому заданию проводилась с целью определения функциональной полноты разрабатываемого комплекса и соответствия требованиям качества результатов работы системы. Основными направлениями экспериментальных испытаний были исследования работы системы с входными данными, корректная работа базы данных.

Для проверки надежности и функциональности системы предлагаются следующие типы тестов.

1. Работа функции авторизации, проверка прав доступа разным группам пользователей: студент.
2. Работа функции выбора дисциплины, для прохождения тестирования.
3. Работа функции выбора темы тестирования по выбранной дисциплине.
4. Работа функции тестирования.
5. Работа функции просмотра результатов и даты прохождения предыдущих тестов (рис. 5.1.).

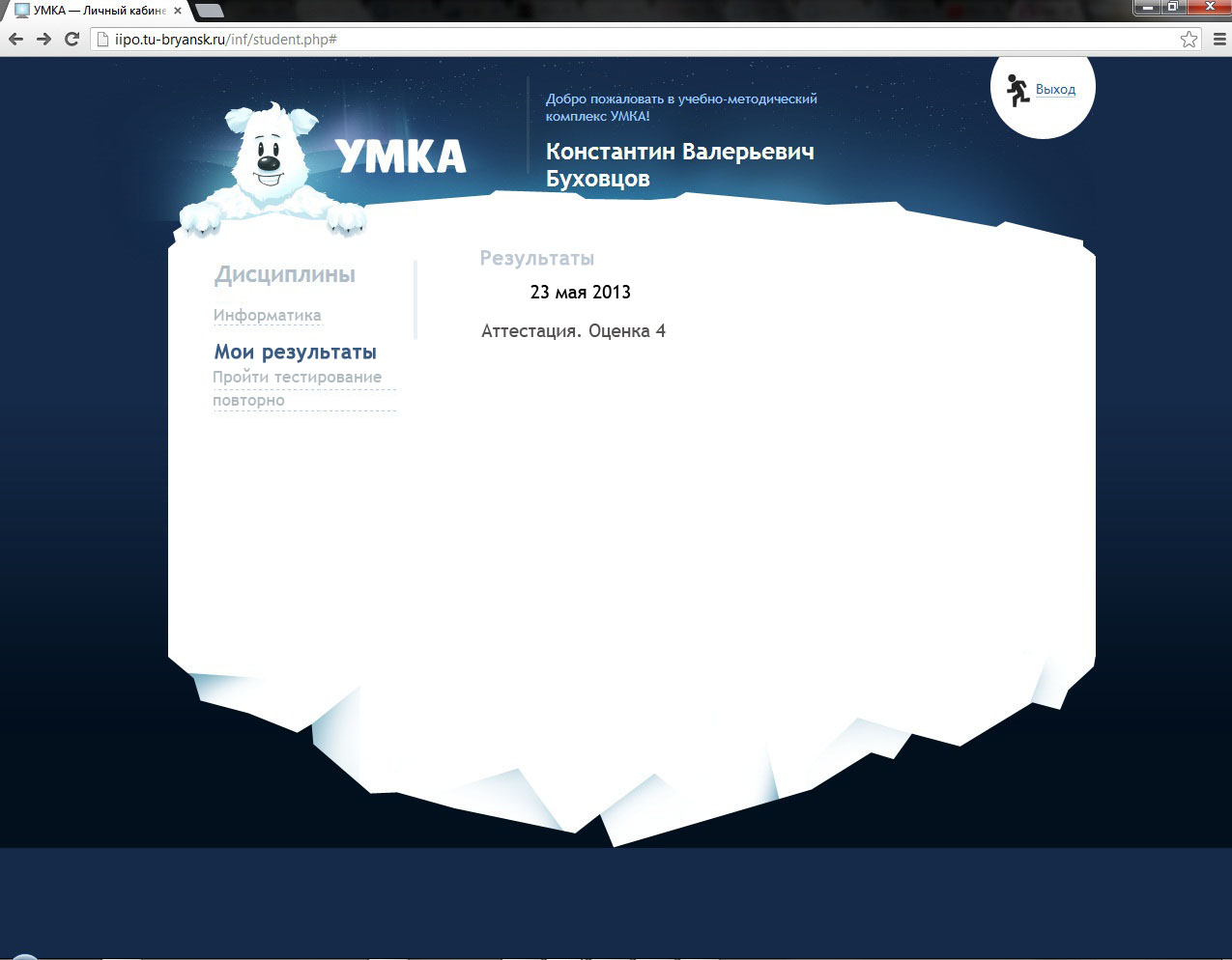


Рис.5.1. Вывод результатов ранее пройденных тестов

Проверка функции тестирования.

1. Проверка функции начальной точки тестирования;
   1. Создается пользователь, у которого нет пройденных тестов, т.е. средняя оценка отсутствует. При выборе прохождения тестирования, система выдает вопросы среднего уровня сложности.
   2. Создается пользователь, у которого средняя оценка равна 3. При выборе прохождения тестирования, система выдает вопросы низкого уровня сложности.
   3. Создается пользователь, у которого средняя оценка равна 4. При выборе прохождения тестирования, система выдает вопросы среднего уровня сложности.
   4. Создается пользователь, у которого средняя оценка равна 5. При выборе прохождения тестирования, система выдает вопросы среднего уровня сложности.
2. Проверка функции перехода с низкого уровня сложности на средний;
3. Проверка функции перехода с низкого уровня сложности на высокий;
4. Проверка функции перехода со среднего уровня заданий на высокий;
5. Проверка функции перехода с высокого уровня заданий на средний;
6. Проверка функции вывода результатов тестирования (рис.5.2.)

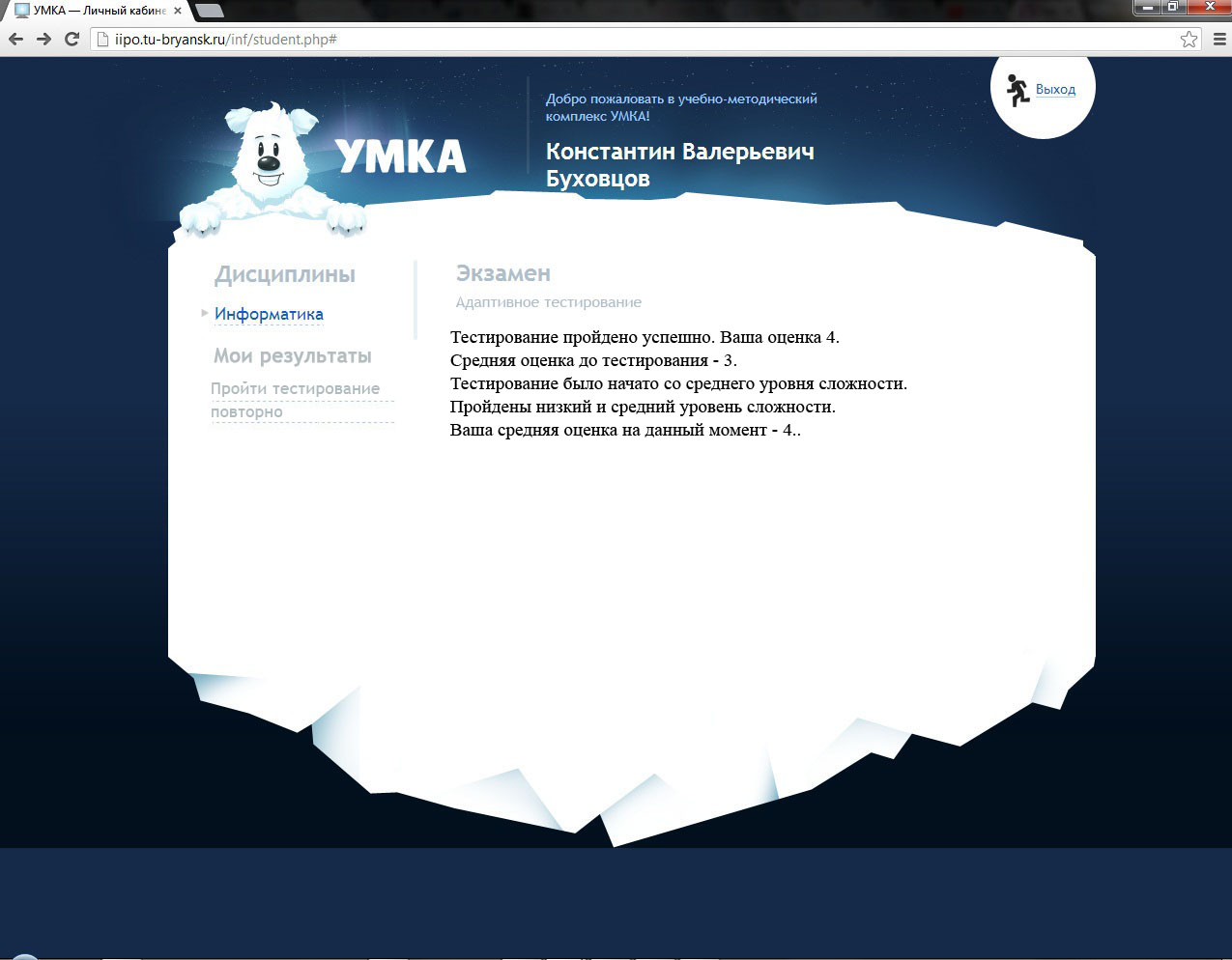


Рис.4.5. Вывод результатов тестирования

## Проверка программной системы в экстремальных условиях

Проверка в экстремальных условиях предполагала оценку работоспособности программы для граничных ситуаций.

К граничным ситуациям относится проверка возможности добавления данных в БД в случае не полного заполнения визуальных форм.

описать ситуацию не хватки вопросов в теме с нужным уровнем сложности

## Проверка системы в исключительных ситуациях

Проверка в исключительных ситуациях выполняется при вводе данных и отсутствующим или ограниченным подключением к интернету.

Однако при работе модуля преподавателя нарушения работы связанные с техническими проблемами не является критичным.

описать ситуацию, когда студент не закончил тестирование м вышел из браузер, тогда…

## Основные выводы

что все прошло успешно!!!

# Техническая документация

## Руководство пользователя

После ввода в адресную строку адреса размещения комплекса «УМК-А» iipo.tu-bryansk.ru/inf появляется страница авторизации единая для администратора и студентов (рис. 6.1).

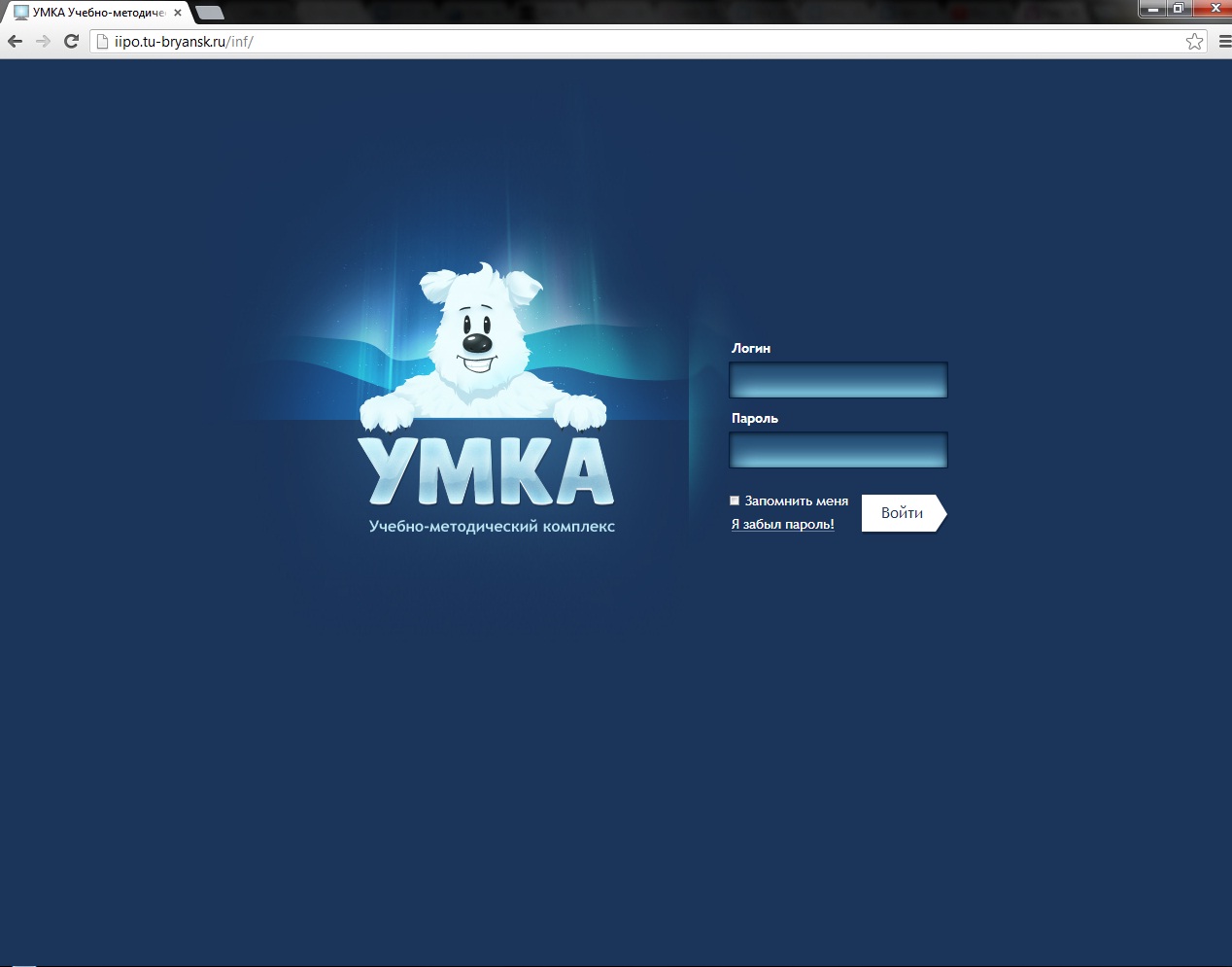


Рис. 6.1. Страница авторизации

Для входа в подсистему необходимо ввести свою фамилию и номер зачетной книжки. При этом зарегистрировать в подсистеме нового студента может только администратор – самостоятельная регистрация не возможна.

После входа в систему студенту доступны следующие группы меню (рис. 6.2):

* **Дисциплины** – здесь имеется возможность выбрать необходимую дисциплину и соответствующую тему тестирования.
* **Мои результаты** – данный пункт меню позволяет узнать результаты ранее пройденных тестов.

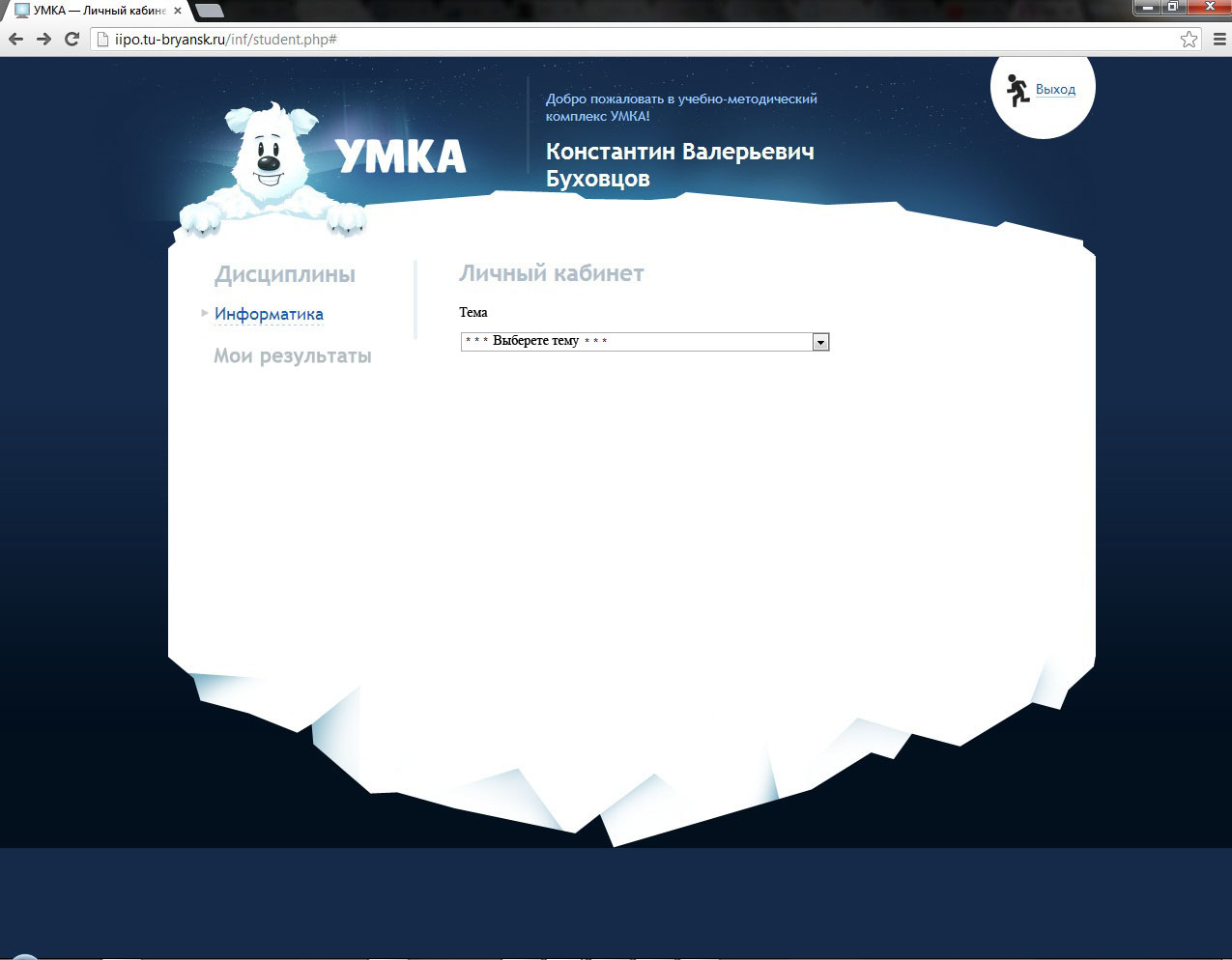
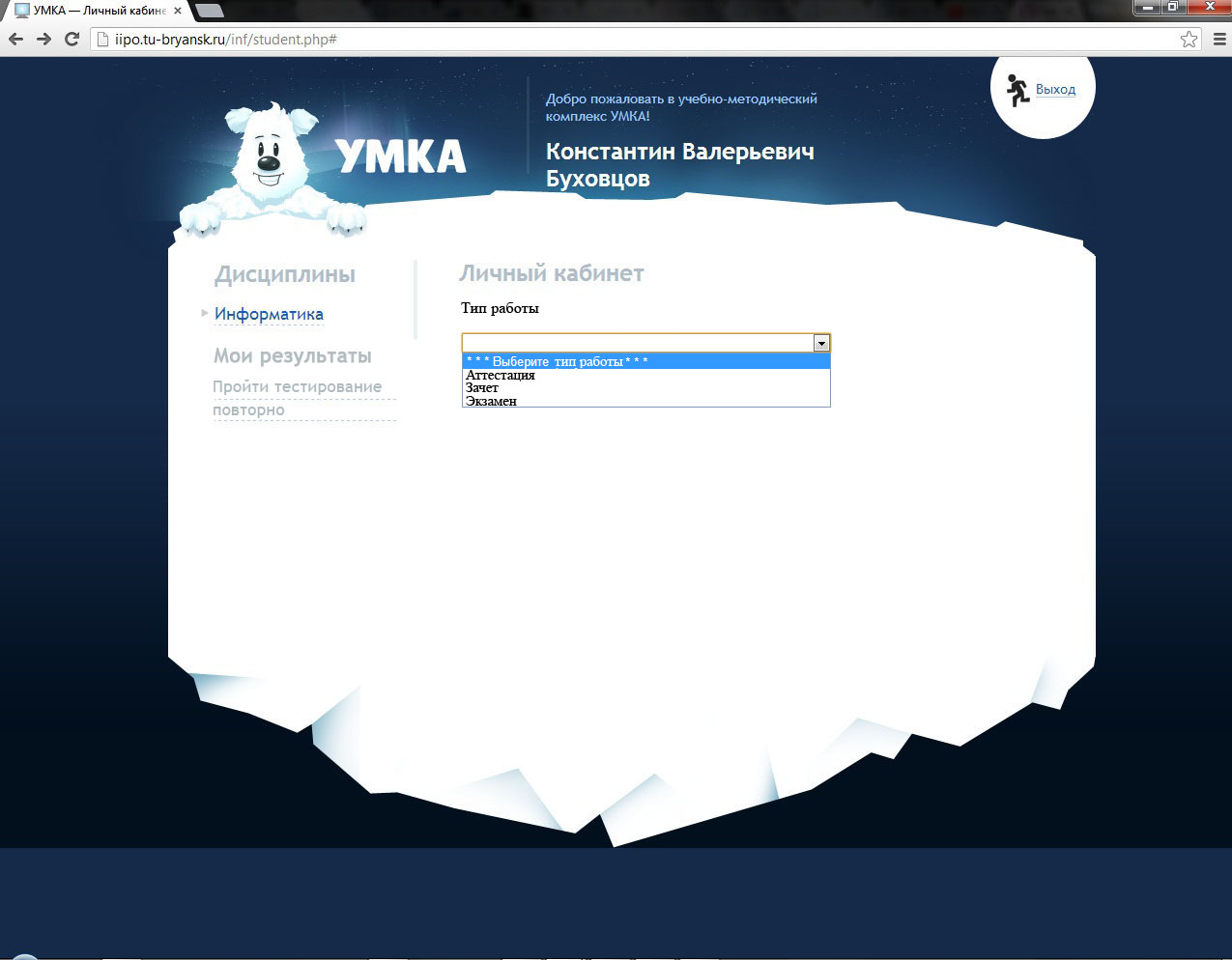


Рис. 6.2. Страница личного кабинета

После выбора темы студент должен выбрать необходимый ему тип задания. Это может быть аттестация, зачет или экзамен (рис.6.3.)



6.3. Страница личного кабинета

Система выводит на экран по одному вопросу с несколькими вариантами ответа. Тестируемый должен выбрать ответ, который считает верным и перейти к следующему вопросу, нажав кнопку «Ответить» (рис 6.4.).

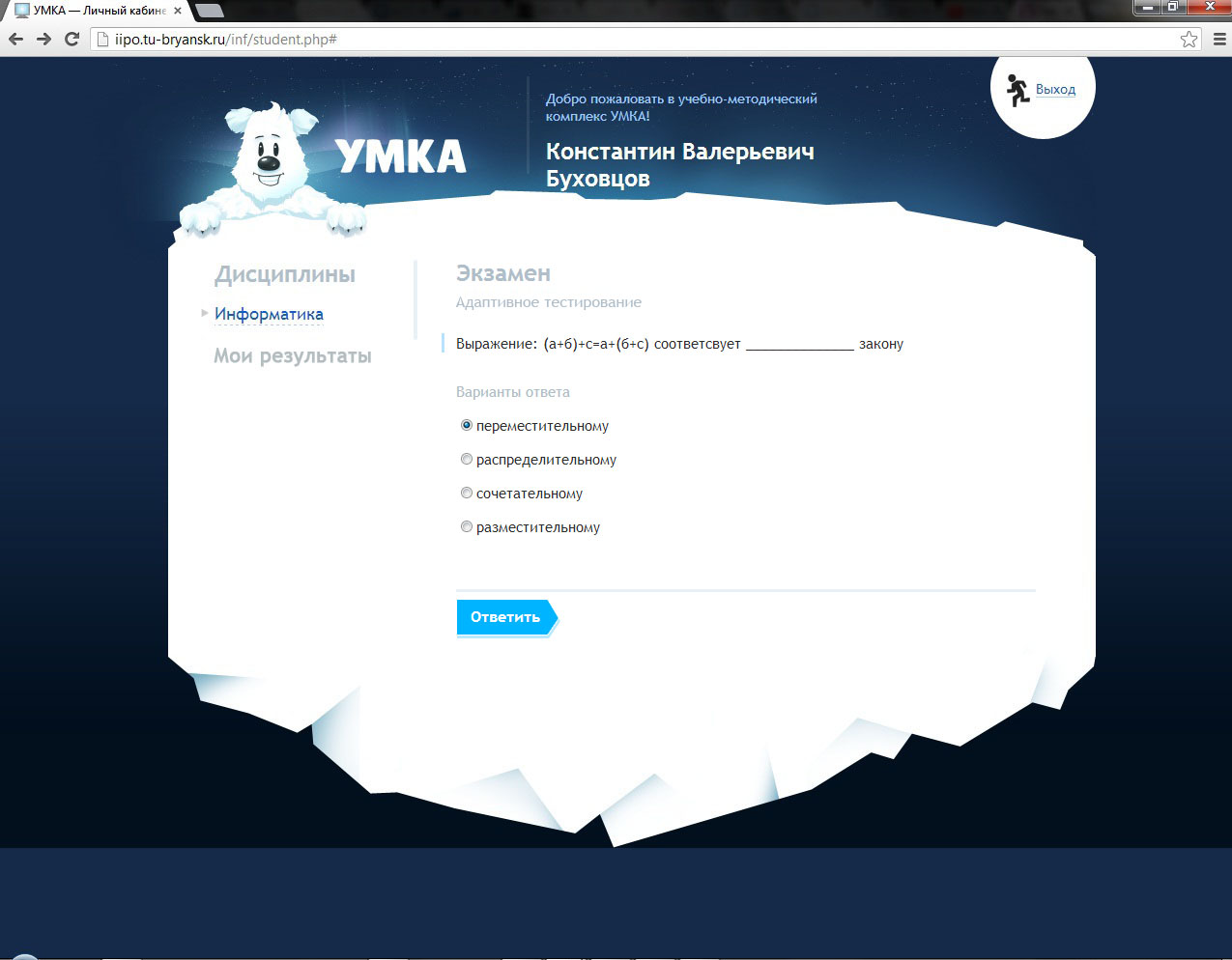


Рис. 6.4. Страница личного кабинета

## Руководство администратора

Установка и обслуживание программы может производиться только администратором.

Работа с программой осуществляется через браузер и не требует каких-нибудь особых навыков. На клиентской программе должен быть установлен один из следующих браузеров.

* MS Internet Explorer 8/9;
* Firefox 9;
* Safari 4;
* Opera 11.60;
* Google Chrome.

Минимальные требования к параметрам технических средств на сервере зависят, прежде всего, от установленной операционной системы. Минимальная конфигурация должна обеспечивать работу web-сервера (Apache) и СУБД (MySQL).

На стороне сервера необходимо обеспечить наличие операционной системы семействаWindows/Unix (или совместимые серверные системы) на которой будет корректно работать нижеперечисленные программные серверы:

* Web-сервер Apache 2.2.13;
* СУБД MySQL 5.1.37;
* PHP 5.3.0 с поддержкой MySQL;
* phpMyAdmin 3.2.1 – система управления MySQL через Web-интерфейс.

Для установки программной системы необходимо знать следующие данные:

* Сетевой или WWW-адрес компьютера, куда производиться установка.
* Имя корневой директории WWW-сервера.
* Адрес БД.
* Имя пользователя СУБД MySQL.
* Пароль пользователя СУБД MySQL.

До начала установки необходимо скопировать файлы системы в корневую директорию сервера. Также необходимо запустить веб-сервер и СУБД MySQL.

### Копирование файлов web-сайта на сервер

Для работы web-сайта необходимо скопировать соответствующие файлы на сервер компании предоставившей хостинг:

* настраиваем FTP соединение рис. 6.4;
* странички, файлы и скрипты загружаем в папку inf.

Рис. 6.4. Настройка FTP клиента

### Присоединение базы данных к MySQL

Для присоединения базы данных к MySQL нужно в панели управления сайтом зайти по ссылке в «Базы данных MySQL» и создать новую базу данных (рис. 6.5). Далее необходимо добавить нового пользователя перейдя по ссылке «Привилегии» и назначить права. В окне «Базы данных» (рис. 6.6), необходимо нажать на кнопку «Импорт» и выбрать базу данных входящую в поставку (bd.sql). После чего подтвердить свой выбор, нажав на кнопку «OK».

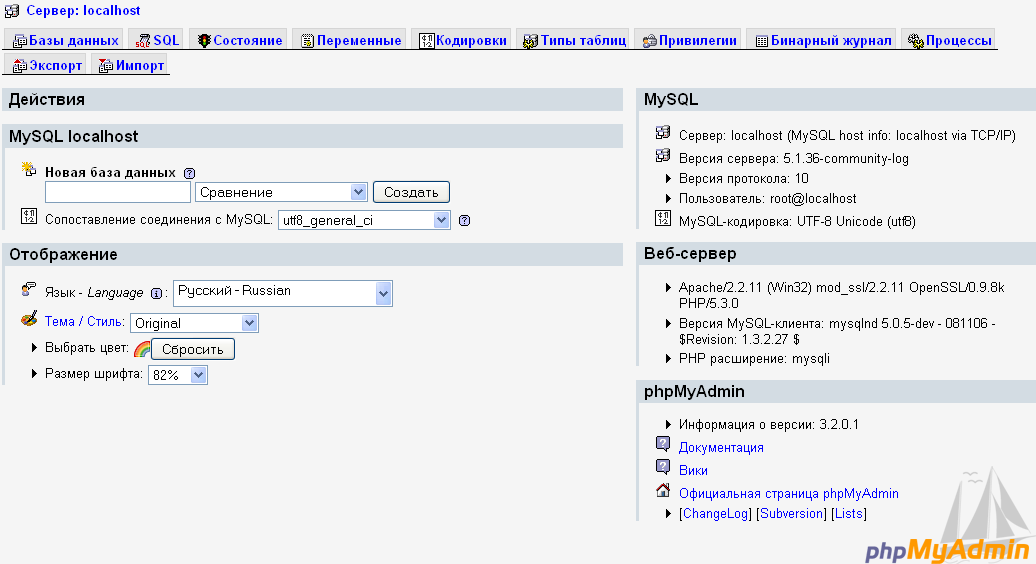


Рис. 6.5. Окно создания новой базы данных панели phpMyAdmin

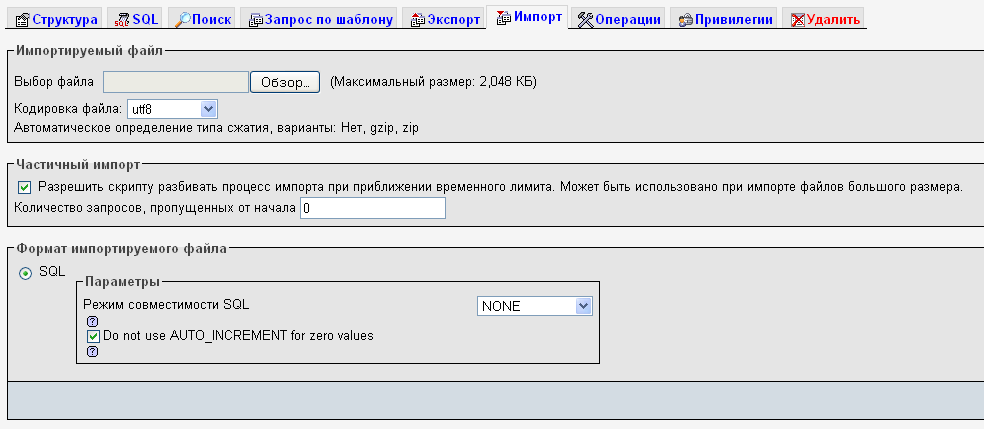


Рис. 6.6. Окно вкладки «Импорт» базы данных MySQL

# БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА

На сегодняшний день компьютеры используются повсеместно, и в некоторых случаях являются жизненно важными компонентами работы систем. Не смотря на то, что компьютерные технологии совершенствуются, на данный момент при работе на нем следует руководствоваться не только санитарными правилами, во избежание различных ЧС, но и заботой о собственном здоровье.

Данный раздел дипломного проекта посвящен рассмотрению следующих вопросов:

* определение требований к рабочему месту программиста;
* расчет освещенности;
* расчет устойчивости производства при чрезвычайной ситуации

## Требования к рабочему месту программиста

### Организация рабочего места

Организация рабочего места на каждой машине имеет свои специфические особенности, зависящие от модели машины, метода работы на ней, характера выполняемой работы, квалификации оператора и т.п. Учитывая специфику машины, рабочее место организуют так, чтобы использовать рациональные приемы работы и эксплуатации машины при наименьшем числе движений оператора и удобном обращении с обрабатываемым материалом.

Планировкой рабочего места называют пространственное расположение основного и вспомогательного оборудования, оснастки и предметов труда, а также самого работающего, обеспечивающее рациональное выполнение трудовых движений и приемов, благоприятные и безопасные условия труда.

Главными элементами рабочего места программиста являются письменный стол и кресло.

При выборе письменного стола следует учитывать следующие принципы:

* высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
* нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;
* поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;
* конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3-х для хранения документации, листингов, канцелярских принадлежностей, личных вещей).

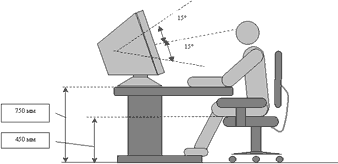
Оптимальными размерами стола являются:

* длина стола 1300 мм;
* ширина стола 650 мм;
* высота 710 мм.

При работе в положении сидя (рис. 8.1) рекомендуются следующие параметры рабочего пространства:

* ширина не менее 700 мм;
* глубина не менее 400 мм;
* высота рабочей поверхности стола над полом 700-750 мм.

Поверхность для письма должна иметь не менее 40 мм в глубину и не менее 600 мм в ширину.



*Рис. 8.1. Рабочее место пользователя*

Важным элементом рабочего места программиста является кресло. При проектировании кресла исходят из того, что при любом рабочем положении программиста его поза должна быть физиологически правильно обоснованной, т.е. положение частей тела должно быть оптимальным. Для удовлетворения требований физиологии конструкция рабочего сидения должна удовлетворять следующим основным требованиям:

* допускать возможность изменения положения тела, т.е. обеспечивать свободное перемещение корпуса и конечностей тела друг относительно друга;
* допускать регулирование высоты в зависимости от роста работающего человека (в пределах от 400 до 550 мм );
* иметь слегка вогнутую поверхность;
* иметь небольшой наклон назад.

Оптимальные размеры рабочего кресла:

* высота сидения от 400 до 500 мм;
* ширина не менее 400 мм;
* глубина не менее 380 мм;
* высота поверхности спинки не менее 300 мм;
* ширина – не менее 380 мм;
* угол наклона спинки от 90º до 110º к плоскости сидения.

Важным моментом является также рациональное размещение на рабочем месте документации, канцелярских принадлежностей.

### Режим труда и отдыха

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

В табл. 8.1 представлены сведения о регламентированных перерывах, которые необходимо делать при работе на компьютере, в зависимости от продолжительности рабочей смены, видов и категорий трудовой деятельности с ВДТ (видеодисплейный терминал) и ПЭВМ (в соответствии с СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ»).

Таблица 8.1

**Время регламентированных перерывов при работе на компьютере**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Категория работы с ВДТ или ПЭВМ** | **Уровень нагрузки за рабочую**  **смену при видах работы с ВДТ** | | | **Суммарное время**  **регламентированных перерывов, мин** | |
|  | **Группа А, количество знаков** | **Группа Б, количество**  **Знаков** | **Группа В, количество**  **Часов** | **При 8-часовой смене** | **При 12-часовой смене** |
| 1 | до 20000 | до 15000 | до 2,0 | 30 | 70 |
| 2 | до 40000 | до 30000 | до 4,0 | 50 | 90 |
| 3 | до 60000 | до 40000 | до 6,0 | 70 | 120 |

В соответствии со СанПиН 2.2.2/2.4.1340-03 все виды трудовой деятельности, связанные с использованием компьютера, разделяются на три группы:

* группа А: работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом;
* группа Б: работа по вводу информации;
* группа В: творческая работа в режиме диалога с ЭВМ - отсутствует.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

### Параметры микроклимата

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. Для комфортной работы сотрудников соблюдаются все основные требования к микроклимату учебных и дошкольных помещений, описанные в санитарных правилах и нормах ГОСТ (табл.8.2).

Таблица 8.2

***Оптимальные параметры микроклимата во всех типах учебных и дошкольных помещений с использованием ПЭВМ***

|  |  |  |  |
| --- | --- | --- | --- |
| Температура, С° | Относительная влажность, % | Абсолютная влажность,  г/м3 | Скорость движения воздуха, м/с |
| 19 | 62 | 10 | <0,1 |
| 20 | 58 | 10 | <0,1 |
| 21 | 55 | 10 | <0,1 |

Объем помещений, в которых размещены работники, не должен быть меньше 19,5 м3 на человека с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены в табл. 8.3.

Таблица 8.3

**Нормы подачи свежего воздуха в помещения**

|  |  |
| --- | --- |
| Характеристика помещения | Объемный расход подаваемого в помещение свежего воздуха, м3 на одного человека в час |
| Объем до 20м3 на человека | Не менее 30 |
| 20-40м3 на человека | Не менее 20 |
| Более 40м3 на человека | Естественная вентиляция |

Для повышения влажности воздуха в помещениях с ВДТ и ЭВМ применяются увлажнители воздуха. Для обогрева помещений в холодное время года используются обогреватели.

### Профессиональные заболевания при работе с ПК

Зрительный синдром. Зрение человека, сформированное в ходе длительной эволюции, оказалось мало приспособлено к зрительной работе с компьютерным изображением.

Экранное изображение от естественного отличается тем, что оно обладает следующими признаками:

* оно самосветящееся (а не отраженное);
* имеет значительно меньший контраст, который еще больше уменьшается за счет внешнего освещения;
* не непрерывное, а состоит из дискретных точек – пикселей;
* мерцающее (мелькающее), т.е. точки с определенной частотой зажигаются и гаснут; чем меньше частота мельканий, тем меньше точность установки аккомодации;
* не имеет четких границ (как на бумаге), потому что пиксель имеет не ступенчатый, а плавный перепад яркости с фоном.

Зрительная нагрузка возрастает из-за необходимости постоянного перемещения взора с экрана на клавиатуру и бумажный текст. Невозможность правильно и рационально организовать рабочее место оператора (блики на экране монитора от внешних источников, неправильное расстояние от глаз до экрана, неудачный выбор цветов, чрезмерно большая яркость экрана) усугубляют ситуацию.

Кроме того, статичная поза во время работы, повторяющиеся движения и нерациональная организация рабочего места могут приводить к возникновению расстройств скелетно-мышечной системы пользователя ПК, которые сопровождаются многочисленными симптомами. При шейном остеохондрозе возникают головные боли, чувство выпирания глазного яблока, пульсирующие боли в глазах, затуманивание зрения, «летающие мушки» и радужные круги.

### Упражнения для предупреждения профзаболеваний

Учитывая все профессиональные заболевания, связанные с работой с персональным компьютером, можно привести в пример 4 типа упражнения, которые помогут предупредить заболевания.

Рекомендуется выполнять следующее упражнение для глаз.

1. Закрыть глаза, не напрягая глазные мышцы, на счет 1-4, широко раскрыть глаза и посмотреть вдаль на счет 1-6. Повторить 4-5 раз.
2. Посмотреть на кончик носа на счет 1-4, а потом перевести взгляд вдаль на счет 1-6. Повторить 4-5 раз.
3. Не поворачивая головы (голова прямо), делать медленно круговые движения глазами вверх-вправо-вниз-влево и в обратную сторону: вверх-влево-вниз-вправо. Затем посмотреть вдаль на счет 1-6. Повторить 4-5 раз.
4. При неподвижной голове перевести взор с фиксацией его на счет 1-4 вверх, на счет 1-6 прямо; после чего аналогичным образом вниз-прямо, вправо-прямо, влево-прямо. Проделать движение по диагонали в одну и другую стороны с переводом глаз прямо на счет 1-6. Повторить 3-4 раза.

Для улучшения мозгового кровообращения рекомендуется выполнять следующее упражнение.

1. Изначальная поза – стоя или сидя, руки на поясе. 1-2 круг правой рукой назад с поворотом туловища и головы направо. 3-4 то же левой рукой. Повторить 4-6 раз. Темп медленный.
2. Исходное положение – стоя или сидя, руки в стороны, ладони вперед, пальцы разведены. 1 – обхватив себя за плечи руками возможно крепче и дальше. 2 – исходное положение. То же налево. Повторить 4-6 раз. Темп быстрый.
3. Исходное положение – сидя на стуле, руки на пояс. 1 – повернуть голову направо. 2 – исходное положение. То же налево. Повторить 6-8 раз. Темп медленный.

Существуют также упражнения общего воздействие. Примером может послужить следующее упражнение.

1. Исходное положение – руки в стороны. 1-4 – восьмеркообразные движения руками. 5-8 – то же, но в другую сторону. Руки не напрягать. Повторить 4-6 раз. Темп медленный. Дыхание произвольное.
2. Исходное положение – стойка: ноги врозь, руки на поясе. 1-3 – три пружинящих движения тазом вправо, сохраняя исходное положение плечевого пояса. 4 – исходное положение. Повторить 4-6 раз в каждую сторону. Темп средний. Дыхание не задерживать.
3. Исходное положение – стойка: ноги врозь, руки на поясе. 1 – руки в стороны, туловище и голову повернуть налево. 2 – руки вверх. 3 – руки за голову. 4 – исходное положение. Повторить 4-6 раз в каждую сторону. Темп медленный.
4. Исходное положение – основная стойка .1 – руки вперед, ладони книзу. 2 – руки в стороны, ладони кверху, 3 – встать на носки, руки вверх, прогнуться. 4 – исходное положение. Повторить 4-6 раз. Темп медленный.
5. Исходное положение – ноги врозь, немного шире плеч. 1-3 – наклон назад, руки за спину. 3-4 – исходное положение. Повторить 6-8 раз. Темп средний.
6. Исходное положение – ноги на ширине плеч. 1 – руки за голову, поворот туловища направо. 2 – туловище в исходное положение, руки в стороны, наклон вперед, голову назад. 3 – выпрямиться, руки за голову, поворот туловища налево. 4 – исходное положение. 5-8 то же в другую сторону. Повторить 6 раз. Темп средний.
7. Исходное положение – руки к плечам. 1 – выпад вправо, руки в стороны. 2 – исходное положение. 3 – присесть, руки вверх. 4 – исходное положение. 5-8 то же в другую сторону. Повторить 6 раз. Темп средний.

Для профилактики и лечения тоннельного синдрома следует позаботиться о том, чтобы рабочее место было удобным, как можно чаще прерывать работу и выполнять небольшой набор упражнений для рук.

* + При работе с клавиатурой, угол сгиба руки в локте должен быть прямым (90 градусов).
  + При работе с мышкой кисть должна быть прямой, и лежать на столе как можно дальше от края.
  + Стул или кресло должно быть с подлокотниками, также желательно наличие специальной опоры для запястья (коврик для мыши, специальной формы клавиатура или компьютерный стол со специальными силиконовыми подушечками и т.п.).
  + Если приходится много и однообразно работать запястьем, нужно приобрести и носить поддерживающий запястье бандаж.

## Расчет искусственного освещения

Согласно СанПиН 2.2.2/2.4.1340-03 освещенность на рабочей поверхности стола при работе с монитором должна быть в пределах 300–500лк. Примем освещенность равной 400лк. Проведем расчет освещения по методу светового потока. Параметрами производственного помещения являются:

* освещенность Е = 400 лк;
* длина помещения а = 10 м;
* ширина помещения b = 6 м;
* расчетная высота светильника hp = 3,6 м;
* тип применяемых ламп – ЛБ.

Определим приемлемое для данного типа светильников расстояние между светильниками. Коэффициент λ для светильников ЛБ равен 1,4.

**

Примем Lсв(мах) = 5 м.

Определяем расстояние между светильником и стеной при наличии рабочих мест у стены

**

Определяем общее количество рядов светильников



Примем Nш(min) = 2.



Примем Nд(min) = 3.

Определяем общее расчетное минимальное количество светильников

.

Определяем показатель помещения



Принимаем коэффициент отражения потолка ρn = 50 %, коэффициент отражения стен ρc = 30 %. Принимаем коэффициент использования светового потока ηи = 0,4. Принимаем коэффициент запаса и коэффициент, учитывающий неравномерность освещения, соответственно k = 1,5; z = 1,1.

По типу светильника, показателю помещения i и коэффициентам отражения потолка и стен определяем коэффициент использования светового потока, определяем по формуле



где х – число источников в светильнике.

По расчетному световому потоку одной лампы *Фрасч* выберем стандартную лампу необходимой мощности со световым потоком *Фтабл*, ближайшим к *Фрасч*. Выбираем лампу ЛБ65, для которой *Фтабл*. =4325 лм.

Рассчитаем необходимое количество светильников для трех типов ламп и занесем результаты в табл.8.1.

.

Таблица 8.1

Расчетное количество ламп разных типов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Тип лампы*** | ***Число светильников*** | | ***Принятое количество,n*** | ***Отклонение,%*** | ***Мощность лампы*** | ***Мощность системы, Вт*** |
| ***Световой поток,***  ***Фрасч лм*** | ***Необходимое количество, nрасч*** |
| ЛБ 40 | 2850 | 8,68 | 8 | -7,8 | 40 | 1280 |
| ЛБ 65 | 4325 | 5,73 | 6 | 4,8 | 65 | 1560 |
| ЛБ 80 | 4960 | 4,99 | 6 | 20,2 | 80 | 1920 |

Определяем отклонение Фрасч. от Фтабл., которое соответствует лампе ЛБ65:

Так как попадает в промежуток от -10% до 20%, следовательно выбор сделан верно.

Рассчитываем полную мощность системы освещения. Которая равна произведению мощности принятой лампы на принятое число ламп.

На основе полученных данных составим план размещения светильников в помещении (рис. 8.1).



## Расчет устойчивости производства при чрезвычайной ситуации

В качестве примера рассмотрим возможную ЧС: на расстоянии  от производственного помещения на железной дороге произошла авария, в результате которой оказались разгерметизированы цистерны с жидким хлором, масса хлора составляет . Направление ветра совпадает с расположением производственного помещения и его скорость равна . Авария случилась в дневное время суток, в пасмурную погоду, емкость не обвалована.

1. Определим степень вертикальной устойчивости воздуха. При указанных метеоусловиях степень вертикальной устойчивости – *изотермия*.
2. Находим глубину *Г* распространения зараженного воздуха при скорости .  для поражающей концентрации.
3. Определим ширину зоны химического заражения. Для изотермии:



Определим площадь зоны химического заражения:



1. Нанесем на план зону химического заражения (рис. 8.3).

На схеме: 1-место непосредственного разлива сильнодействующих ядовитых веществ, 2-производственное помещение, 3-зона химического заражения.

***Направление ветра***

***Г = 14,2 км***

***SЗ = 15,123 км2***

***Ш = 2,13 км***

***R = 5 км***

***3***

***1***

***2***

***Рис. 7.2. Схема зоны химического заражения***

1. Для изотермии и скорости ветра  находим среднюю скорость переноса облака зараженного воздуха  Время подхода зараженного воздуха к помещению:



1. Время поражающего действия СДЯВ tпор в очаге химического поражения определяется временем испарения СДЯВ с поверхности его выброса (разлива):

Для СДЯВ хлора при скорости V= 1 м/с и емкости необвалованной время испарения СДЯВ равно 1,3.

Из этого следует, что

1. Вывод.

Из расчетов можно сделать вывод, что разрушение необвалованной емкости, содержащей хлор 40 т, может повлечь за собой последствия, связанные с поражением людей.

Для защиты рабочих и служащих, а также людей, находящихся в жилых районах недалеко от химических предприятий, необходимо всем подняться на верхние этажи.

# ЗаключЕние

В ходе выполнения дипломной работы был разработан модуль адаптивного тестирования.

Разработанный модуль представляет собой результат проектирования системы, включающего в себя функциональное моделирование системы, построение диаграмм потоков данных и создание базы данных.

Исходя из анализа аналогов, их преимуществ и недостатков, были выработаны требования, позволяющие данной разработке быть конкурентоспособной и удовлетворять требованиям пользователей. Также, анализ программ аналогов позволил создать удобный интерфейс пользователя, что делает работу с модулем простой и не требует дополнительного обучения.

# Список литературы

1. Мосолков, А. Е. Электронные образовательные ресурсы нового поколения (ЭОР) [Электронный ресурс].- Режим доступа: http://www.metod-kopilka.ru/page-article-8.html
2. Булгагова, Е.Т. Использование информационных технологий в учебном процессе [Электронный ресурс].- Режим доступа:http://science.ncstu.ru/articles/hs/12/07.pdf/file\_download
3. GMAT [Электронный ресурс]. – Режим доступа: <http://www.bkc.ru/exams/gmat>
4. АСТ-тест Player 4 [Электронный ресурс]. – Режим доступа: http://www.ast-centre.ru/testirovanie/ast\_test/705/
5. АСТ-тест Administrator 4.0 [Электронный ресурс]. – Режим доступа: <http://www.ast-centre.ru/testirovanie/ast_test/706/>
6. ACT-Тест Player [Электронный ресурс]. – Режим доступа: <http://www.call.lunn.ru/pdf/pr_otchoyt_I.pdf>
7. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с
8. Рудаков, А.В. Технология разработки программных продуктов / А.В. Рудаков. – М.: Издательский центр "Академия", 2006.
9. Вальд А., Последовательный анализ, пер. с англ., М., 1960.
10. Применение модели Раша. Общее понятие модели – [Электронный ресурс]. – Режим доступа: http://www.wikiznanie.ru/ru-wz/index.php/Использование\_Теории\_тестовых\_заданий\_(Item\_Response\_Theory)\_в\_адаптивном\_тестировании.
11. Фаулер, М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер. – СПб.: Символ-Плюс, 2011.
12. Фленов, М. Web-сервер глазами хакера (+ CD) / М. Фленов. – СПб: БХВ-Петербург, 2007.
13. Челышкова, М.Б. Теория и практика конструирования педагогических тестов. Учебное пособие / М.Б. Челышкова. – М.: Логос, 2002.
14. Майоров, А.Н. Теория и практика создания тестов для системы образования: Как выбирать, создавать и использовать тесты для целей образования / А.Н. Майоров. – М.: Интеллект-центр, 2002.
15. Ломов, А.Ю. Apache, Perl, MySQL: практика создания динамических сайтов. Самоучитель / А.Ю. Ломов. – СПб: БХВ-Петербург, 2007.
16. Коггзолл, РНР 5. Полное руководство /Коггзолл. – М.: Издательский дом "Вильямс", 2006.
17. Веллинг, Л. MySql: Учебное пособие / Л. Веллинг, Л. Томсон. – М.: Вильямс, 2006. – 304с.
18. Гольцман, В. MySQL 5.0:/ В. Гольцман - СПб.:Питер, 2009 . - 256с.
19. ОРММ ИСЖТ 5.03–00. Процессы жизненного цикла ИС и программных средств – М. : ВНИИАС МПС России, 2000. – 48 с.
20. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М. : Финансы и статистика, 1998. – 176 с.
21. Марка, Д.А. Методология структурного анализа и проектирования SADT / Д.А. Марка, К. МакГоуэн. – М. : МетаТехнология, 1993. – 243 с.
22. Маклаков, С.В. BPwin и ERwin. CASE-средства разработки информационных систем / С.В. Маклаков. – М. : ДИАЛОГ-МИФИ, 2001. – 304 с.
23. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. – М. : ДИАЛОГ-МИФИ, 2005. – 432 с.
24. Адаптивное тестирование [Электронный ресурс]. – Режим доступа: <http://www.gumer.info/bibliotek_Buks/Pedagog/testing/>