

АГЕНТСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ УЗБЕКИСТАНА

**ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

КАМАЛОВ ШУХРАТ КАМАЛОВИЧ

**РАЗРАБОТКА АЛГОРИТМА И ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ СЕМАНТИЧЕСКОЙ МОДЕЛИ
ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

5A330601 - “Программный инжиниринг”

ДИССЕРТАЦИЯ

на получение степени магистра

**Научный руководитель: д.т.н.,
проф. Раджабов Б.Ш.**

Ташкент – 2015

ВВЕДЕНИЕ	4
I ГЛАВА. ОРГАНИЗАЦИЯ УЧЕБНОГО ПРОЦЕССА НА ОСНОВЕ WEB–ТЕХНОЛОГИЙ	17
1.1. Применение компьютерных и информационных технологий в организации учебного процесса.....	17
1.2. Интеллектуальная система управления учебного процесса на основе Web-технологий.....	19
1.3. Общее определение программной среды, используемой для организации учебного процесса на основе Web-технологий.....	25
1.4. Цель исследования и его основные задачи.....	34
Выводы по I главе.....	36
II ГЛАВА. АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СЕМАНТИЧЕСКОЙ МОДЕЛИ ОБРАЗОВАТЕЛЬНЫХ ПРОЦЕССОВ	37
2.1. Общее определение электронных учебных ресурсов.....	37
2.2. Дискретные математические модели виртуализации учебных процессов.....	42
2.3. Семантическая модель электронных учебных ресурсов.....	52
2.4. Разработка многоступенчатой математической модели учебного процесса.....	58
2.5. Программное обеспечение размещения электронных учебных ресурсов в образовательных порталах.....	66
Выводы по II главе.....	77
III ГЛАВА. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВИРТУАЛИЗАЦИИ УЧЕБНОГО ПРОЦЕССА НА ОСНОВЕ WEB-ТЕХНОЛОГИЙ И ЕГО ПРИЛОЖЕНИЯ	78
3.1. Пользовательский интерфейс программы виртуализации учебных процессов.....	78
3.2. Программное обеспечение проведения дистанционно-тестового	

контроля на основе Web-технологий.....	81
3.3. Программное обеспечение автоматизированной системы для тестирования результатов программы, созданной на основе Web-технологий.....	90
3.4. Виртуальная модель обучения предмета “Информатика” на основе Web-технологий.....	97
Выводы по III главе.....	113
ЗАКЛЮЧЕНИЕ	114
СПИСОК ЛИТЕРАТУРЫ	116
ПРИЛОЖЕНИЯ	125

ВВЕДЕНИЕ

В связи с достижением независимости Республики Узбекистан во всех областях социально-экономической системы произошли положительные изменения. Известно, что информационно-коммуникационные технологии являются одним из основных секторов обеспечения социально-экономического развития нашего государства. Для дальнейшего развития этого сектора Президентом Республики Узбекистан и Правительством приняты ряд Указов и Постановлений. В частности, были приняты и внедрены Закон «Об информатизации» (Закон Республики Узбекистан №560-П, 11.12.2003 г.), Указ «О дальнейшем развитии компьютеризации и внедрения информационно-коммуникационных технологий» (Указ Президента Республики Узбекистан №УП-3080, 30.05.2002 г.), Постановление «О дополнительных мерах по дальнейшему развитию информационно-коммуникационных технологий» (Постановление Президента Республики Узбекистан №ПП-117, 08.07.2005 г.), Постановление «Об организации общественной образовательной информационной сети Республики Узбекистан» (Постановление Президента Республики Узбекистан №ПП-191, 28.09.2005 г.), «О мерах по дальнейшему развитию компьютеризации и внедрения информационно-коммуникационных технологий» (Постановление Кабинета Министров Республики Узбекистан №200, 06.06.2002 г.), Постановление «О дальнейшем развитии сети “ZiyoNet”» (Постановление Кабинета Министров Республики Узбекистан №282, 28.12.2005г.). На основании данных Законов, Указов и Постановлений во всех системах образования учебных заведений нашей Республики широко используются информационно-коммуникационные технологии. Такая ситуация дала возможность создания в республике мировых стандартов системы образования. Для этого имеются все правовые, материально-технические, научные и организационные основы.

Актуальность темы. В основе применения в учебном процессе информационно-коммуникационных технологий (ИКТ) лежит виртуализация учебного процесса в разрезе предметов, организация УП с помощью мультимедиа и сетевых систем. В настоящее время во многих учебных заведениях в большом объеме созданы электронные учебные ресурсы. В масштабах республики работает сеть “ZiyoNet”, обеспечивающая эффективное использование этих ресурсов. Однако взаимообмен электронными учебными ресурсами, созданными в учебных заведениях в разрезе направлений и предметов, находится еще на недостаточном уровне.

При разработке эффективных электронно-учебных ресурсов важное значение имеет наличие семантической модели учебного процесса, которая обеспечивает взаимосвязь элементов изучаемого материала. Более того при разработке других видов модели УП, раскрывающие закономерность УП, необходимо знать контенты учебного материала и их взаимосвязи. Эту задачу можно решить путем разработки семантической модели УП и его алгоритмического и программного обеспечения. В связи с этим виртуализация, на основе семантической модели эффективного обучения каждого предмета в разрезе направлений, их размещение в соответствующей локальной сети приобретает важное значение. Для эффективного решения этих задач создание программного обеспечения виртуализации УП, их стандартизация и инициализация на программных платформах на уровне мировых стандартов приобретает также немаловажное значение. Научные, технико-организационные исследования в этом направлении требуют разработки программного обеспечения УП в современных программных средах, и они вместе призваны формировать систему медиаобразования. Помимо того, что создание программного обеспечения на основе Web-технологий является сложным процессом, они требуют привлечения множества интеллектуальных и технических ресурсов, а система требований, предъявляемая к такому программному обеспечению, создает возможность поднятия УП на новый уровень. Структура программного обеспечения, среда его создания,

модули управления, возможности компонентов – все это направлено к единой цели – виртуализации учебного процесса по всем предметам. В настоящей магистерской диссертации предусмотрена разработка алгоритма и программного обеспечения семантической модели образовательного процесса, а также виртуальных моделей на основе Web-технологий в разрезе предметов УП, разработка их программного обеспечения на основе сценариев обучения, рассмотрены вопросы внедрения дистанционного обучения на базе современных мультимедийных систем.

Использование в системе образования виртуализации УП и мультимедийных средств оказывает положительное влияние на повышение качества обучения. Особенно, в техническом направлении, в частности, в обучении предметов направления ИКТ внедрение мультимедийных средств и Web-технологий, оперативное внедрение в УП новых данных по каждому предмету обеспечивает высокую эффективность. Кроме того, виртуализация УП создает широкие возможности для обучающегося, в частности, обеспечивает целостность, оперативность получаемых знаний. Кроме того, создается возможность оперативного получения дополнительных сведений в рамках виртуально изучаемой темы. Значит, методы введения Web-технологий в учебный процесс и внедрение методов их эффективного использования, создание их математического и программного обеспечения входит в число актуальных научно-технических задач.

На основании вышеизложенного тема диссертации считается актуальной научной темой в направлении внедрения ИКТ в УП.

Степень изученности темы. До настоящего момента научно-исследовательские работы по получению решения задачи внедрения информационно-коммуникационных технологий в учебный процесс ведутся с 90- годов 20 века. В этом направлении опубликованы работы следующих ученых и исследователей: Роналд Г.Рагсдале, Б.Ф.Скиннер, Л.Зайцева, Жон Д.Закис, О.Р.Воронцова, С.Ф.Катержина, С.В.Коршунова, В.Н.Гузненкова, А.М.Мазурина,

А.Н.Савин, А.В.Соловов. В нашей республике в этом направлении имеются научные работы академика М.М.Комилова, профессоров А.Х.Нишонова, Р.Х.Хамдамова, Ш.А.Назирова, Ф.М.Зокировой, Б.Ш.Раджабова, Б.Бегалова и других. Эти исследователи в своих работах определили концептуальные подходы к решению задач о роли компьютерных технологий в организации учебного процесса, разработке семантической модели УП, моделированию взаимоотношений между обучаемым и преподавателем, которое необходимо для использования этих компьютерных технологий, организации взаимного диалога с помощью программных средств, интерфейса, автоматизации систем тестового контроля.

Вначале, в 90-годы прошлого века возникли электронные лекции, электронные формы практических и лабораторных занятий, а также учебной литературы. На следующих этапах были разработаны электронные учебники, учебные пособия и системы обучения, основанные на корпоративном принципе по ряду общеобразовательных и профильных предметов. На их основе в большинстве направлений системы обучения, в частности, математике, физике, технике, экономике, биологии, медицине и по гуманитарным предметам были созданы специальные обучающие системы и соответствующее им программное обеспечение.

При дальнейшем развитии ИКТ возникли задачи полной виртуализации УП, задачи их объединения с информационно-ресурсными центрами высших учебных заведений. При решении этих задач родилась потребность широкого использования сетевых решений, в частности, возможностей сети Интернет. Виртуальный урок и его программное обеспечение сформировало возможность установления он-лайн-отношений между учеником и учителем. В результате в сети Интернет появилось множество обучающих автоматизированных систем.

Исследования следующего этапа были посвящены изучению педагогическо-психологических особенностей учеников во время виртуальных занятий при предположении математического, семантического и других видов моделирования

в процессе виртуализации УП. Однако каждая создаваемая виртуальная система по каждому предмету, исходя из свойств предмета, будет иметь определенное содержание и вид. Поэтому программное обеспечение виртуализации УП должно предоставить возможность наблюдения всех вышеуказанных свойств. В последнее время во многих ведущих высших учебных заведениях мира качественные показатели процесса образования резко изменились. В основе этого лежат учебные системы, основанные на мощных электронных учебных базах ресурсов и Web-технологии. На основе создания и использования таких систем лежит создание математического и программного обеспечения виртуальных учебных систем.

Связь диссертационной работы с планом научно-исследовательских работ. Диссертационная работа выполнена в рамках научно-практических исследований, проводимых в Ташкентском университете информационных технологий за номером АТД 17-014 на тему “Разработка программного комплекса виртуализации УП в системе высшего образования”, выполненных в период 2012-2014 годов.

Цель исследования обусловлена разработкой и внедрением математических моделей, в частности, семантической модели УП с соответствующими программными обеспечениями виртуализации учебных процессов на основе Веб-технологий.

Задачи исследования. В рамках реализации цели диссертации предусмотрено решение следующих основных задач:

- на основе электронно-учебных ресурсов сформировать его в качестве единого объекта, исследовать его основную структуру и параметры;
- разработка дискретных, семантических и многоэтапных математических моделей учебных процессов, определение принципов их устойчивости и адекватности;

- разработка и внедрение программного обеспечения размещения электронно-учебных ресурсов в систему обучения, основанного на Web-технологий;

- разработка пользовательского интерфейса виртуализации учебных процессов с использованием семантической модели УП и программного обеспечения технологии проведения дистанционного тестового контроля;

- создание программного обеспечения автоматизированной системы тестирования результатов программы, созданной на основе веб-технологий, и создание виртуальной модели обучения предмета «Информатика»;

- создание системы организации внедрения в учебные процессы в разрезе предметов программного обеспечения, основанного на Web-технологиях.

Объект исследования. В диссертационной работе в качестве объекта выступают образовательные процессы ВУЗов, имеющиеся математические модели, основанные на семантике учебных процессов, и методы их совершенствования, а также программные средства, основанные на Web-технологиях, которые приспособляют электронные учебные ресурсы для использования в УП на основе Web-технологий.

Предметом исследования являются методы разработки и использования программных средств, основанных на Web-технологиях, позволяющие реализовывать модели, в частности, семантические модели в УП, а также методы разработки соответствующего программного обеспечения в процессе подготовки специалистов в системе высшего образования, в частности, в области информационно-коммуникационных технологий.

Методы исследования. В магистерской диссертации используются методы системного анализа, методы разработки математических моделей УП, в частности, семантической модели УП. Исследуется методика разработки соответствующих программных модулей, обеспечивающих реализацию вышеизложенных моделей на базе Web-технологий. Используются методы

разработки программного обеспечения интеллектуальной системы управления УП, на основе Web-технологий.

Гипотеза исследования. В качестве гипотезы исследования предусматриваются процессы оптимального использования семантической модели УП, которая повышает качество и эффективность при виртуализации образовательного процесса на основе Web-технологий.

Основные положения, выносимые на защиту: В магистерской диссертации на защиту выносятся следующие положения:

- семантическая модель образовательных процессов и методы их использования в проектировании электронно-учебных ресурсов;
- многоэтапная математическая модель учебного процесса;
- программное обеспечение виртуализации учебного процесса, основанное на Web-технологиях;
- программное обеспечение для размещения электронно-учебных ресурсов в системе обучения, основанной на Web-технологиях;
- программное обеспечение проведения дистанционного тестового контроля;
- программное обеспечение системы управления учебным процессом основанное на Web-технологиях.

Научная новизна исследования. Одним из основных способов использования Web-технологий в УП является создание интерактивных Web-узлов и Web-страниц, работающих в локальной сети, разработка их математического и программного обеспечения. В диссертации научно обоснована возможность использования семантической модели УП при виртуализации образовательных процессов на основе Web-технологий. Разработаны другие методы моделирования, позволяющие идентифицировать образовательные процессы. Обоснованы стратегия и тактика разработки соответствующего программного обеспечения. При этом при проектировании программного обеспечения обоснована эффективность использования таких алгоритмических языков как Delphi, C++, Java, HTML, PHP, Apache-Сервер.

Научное и практическое значение результатов магистерской диссертации является то, что на основе оптимальной комбинации семантической и других типов моделей предлагается использовать полностью автоматизированные виртуальные модели образовательных процессов, которые могут функционировать в сети (на основе системы клиент-сервер). Важная сторона таких систем состоит в том, что преподаватели, ученики и другие пользователи, не являющиеся специалистами в области компьютерных технологий, имеют возможность создания Web-технологичных вариантов своих электронных учебных ресурсов. На основе задач, решаемых в рамках магистерской диссертации, предусмотрено создание моделей виртуализации процессов обмена информацией на базе компьютерных сетей, электронных библиотек и электронных учебных ресурсов. Научная суть работы заключается в обеспечении адекватности создаваемых моделей и соответствующего программного обеспечения.

Внедрение результатов исследования. Математическое (на базе семантики) и программное обеспечение виртуализации учебных процессов, использующие предложенные модели, позволяют создать единое информационное образовательное пространство на основе Web-технологий. Было сформировано в качестве единого программного комплекса в конкретной программной среде, реализуемые в корпоративных компьютерных сетях, в частности, в серверах учебных заведений и через сеть Интернет. Результаты исследования использованы в лекционных и практических занятиях для студентов 3 курсов бакалавриата ТУИТ, а также предлагается внедрить в УП системы высших учебных заведений, специального профессионального образования, в результате чего ожидается резкое повышение качества и эффективности процесса образования.

Апробация работы. Научно-практические результаты диссертации были изложены и обсуждены на Республиканской научно-практической ***** своё

Публикация результатов исследования. Научно-практические результаты магистерской диссертации были опубликованы в двух научных статьях, в тезисах конференции молодых преподавателей и магистрантов ТУИТ.

Структура и объем диссертации. Диссертация состоит из введения, 3 глав, заключений по главам, заключительного вывода и приложений, которые изложены на 125 страницах. Список использованной литературы состоит из 97 наименований.

Во введении кратко изложены актуальность проблемы, степень разработки, определение имеющихся противоречий и цель, задачи, объект, предмет, гипотеза, новизна, теоретическое и практическое значение, методологические основы исследования, использованные методы, масштаб применения и этапы его осуществления.

В первой главе изложены подходы применения компьютерных и информационных технологий в организации процесса образования, принципы построения интеллектуальной системы управления УП на основе Web-технологий, а также общая характеристика программной среды, используемой для организации процесса обучения на основе Web-технологий. В частности, полностью проанализирована роль компьютерных и информационных технологий в процессе образования, их возможность повышения качества и эффективности учебного процесса.

На основе анализа имеющиеся электронные учебные ресурсы: лекции, практические и лабораторные занятия рассматриваются в разрезе самостоятельных работ в качестве основного этапа. Поставлена задача сопоставления таких ресурсов к общим сетевым технологиям и серверным технологиям использования в УП. Приводится сравнительный анализ традиционной системы обучения и процессов образования на основе Web-технологий.

При этом проанализирована сущность и хронология научно-исследовательских работ, проведенных на ранних этапах виртуализации учебных

процессов, и обеспечено наличие дополнения их результатов на основе современных достижений ИКТ.

На основе полного изучения возможностей Web-технологий были определены закономерности разработки семантических, статических и динамических моделей УП, логической связи между структурой и параметрами этих моделей. На основе этих закономерностей будет разрабатываться интеллектуальная система управления учебным процессом. То есть логическая связь между компонентами, составляющими УП, будет формировываться в виде “учебный процесс – реализация информационной системы и системы знаний – система оценки полученных знаний – практические указания”.

А также в рамках проведенного анализа были сформированы понятия виртуальной системы обучения, модули проведения занятий, модули осуществления контроля и модули передачи в узел управления результатов в разрезе предмета, выбранного в узле локальной сети.

Были изучены возможности и общее определение математического и программного обеспечения, дающего возможность безусловного применения в УП созданной интеллектуальной системы. На основе проведенных теоретических и практических анализов в конце главы изложены цель диссертационной работы и решаемые в нем основные задачи.

Вторая глава диссертации посвящена общему определению электронно-учебных ресурсов, семантической модели электронно-учебных ресурсов, многоэтапной математической модели процесса обучения. **Сформулированы** задачи по созданию программного обеспечения размещения электронно-учебных ресурсов в систему обучения, основанного на Web–технологиях, и предложены их решения.

Содержание компонентов учебно-методического комплекса по предмету разрабатывается согласно учебной программе, разработанной на основе Государственного стандарта образования, а также на основе личностно ориентированных, развивающих технологий, принципов и требований

самостоятельного образования. Согласно этим требованиям излагается технология создания электронных учебных ресурсов. В частности, были разработаны модели электронных учебников, учебных пособий. Предложено соответствующее программное обеспечение создания раздаточных материалов и систем электронного тестового контроля. Были разработаны технологии использования в УП электронных форм, гипертекстов, виртуальных объектов и процессов с помощью средств мультимедиа.

В рамках диссертации разработана образцовая структура электронного учебника, составляющего электронно-образовательный ресурс, и даны рекомендации по их использованию.

При полной виртуализации учебных процессов для раскрытия закономерностей, происходящих в них в реальности, дало высокий эффект использование методов математического моделирования и оптимизации. Поэтому в пунктах 2.2, 2.3 и 2.4 данной главы созданы дискретные, семантические и многоэтапные математические модели учебных процессов.

Если созданные математические модели выражают закономерности, идентифицирующие функциональные зависимости между сформированными в УП информационных потоков, находящихся в движении, и дискретные математические модели выражают закономерности между компонентами УП, то семантические модели формируют логическую последовательность знаний и сведений, которые преподносит УП ученику. А многоэтапные математические модели помогают в нахождении закономерностей единой структуры УП, структурных и параметрических связей внутри УП.

В этой связи следует подчеркнуть, что разработанные семантические и математические модели полностью отвечают условиям однозначности, устойчивости и адекватности. На основе этих моделей можно создать эффективные алгоритмы и программное обеспечение процесса виртуализации УП.

В частности, при внедрении электронных учебных ресурсов, были разработаны основные принципы применения Web-технологий. В целях систематизации этих принципов и создания технологии их применения в реальном учебном процессе в этой главе поставлена задача разработки семантической модели электронных учебных ресурсов, и эта задача была обоснованным образом решена. При этом изучена спецификация и определение контентной структуры, составляющей основу семантической модели. На основе определений семантических связей, анализа их иерархической структуры разработано программное обеспечение внедрения семантической модели.

Разработано программное обеспечение размещения в системе обучения электронных учебных ресурсов, основанных на Web-технологиях. То есть в качестве заключения этой главы предложено программное средство, используемое при создании электронных учебных ресурсов. Сформированы практические рекомендации по использованию этого программного средства.

В третьей главе диссертации разработано программное обеспечение специального интерфейса пользователя, которое виртуализирует УП. Это программное обеспечение сформировано в виде интерфейса пользователя, которое обладает следующими возможностями:

- обращение пользователя к виртуальному УП осуществляется в режимах непосредственного диалога и дистанционного режима;
- возможность введения необходимых информационных ресурсов и пользование ими в виртуальной системе для организации информационных ресурсов, имеющих теоретическое значение, и организации экспериментальных и практических процессов;
- обобщение результатов виртуальных учебных процессов, их передача в локальную сеть и предоставление этих результатов в необходимой форме.

Виртуализированная учебная система на основе Web-технологий была в качестве тестового испытания опробована в высших учебных заведениях на примере преподавания предмета “Информатика”, и были проанализированы

результаты. При этом было создано программное обеспечение виртуализации процессов электронной лекции, виртуальной лаборатории и практических занятий, а также процессов выполнения студентами самостоятельных занятий. Было создано программное обеспечение организации системы тестового контроля, основанного на Web-технологиях, которая считается важным компонентом учебного процесса, и оно было внедрено на примере высших учебных заведений.

Была разработана полная виртуальная модель по изучению темы «Основы алгоритмизации», обучаемой в рамках предмета «Информатика», и были даны рекомендации по его непосредственному использованию в УП. Это программное обеспечение было полностью изложено вместе с компонентами.

Было разработано программное обеспечение автоматизированной системы, которое контролирует (тестирует) компьютерные программы студентов, созданных в процессе выполнения лабораторных, практических и самостоятельных занятий по предмету «Дастурлаш таъмийноти».

На основе вышеизложенных научных и практических результатов было создано программное обеспечение внедрения на основе Web-технологий виртуальной системы, взаимно связывающей все компоненты учебного процесса. Были разработаны соответствующие рекомендации по непосредственному использованию этого программного обеспечения в учебном процессе.

В конце каждой главы изложены заключения и выводы по научно-практическим результатам, полученным в этих главах.

В **заключении** данной магистерской диссертационной работы были обобщены научные и практические результаты, полученные по каждой главе, и все они были систематизированы.

В приложениях приведены блок-схемы, программное обеспечение и результаты проведенных экспериментов, касающихся виртуальной системы УП, основанного на Web-технологиях.

ГЛАВА I. ОСНОВЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА, ОСНОВАННОГО НА WEB – ТЕХНОЛОГИЯХ

1.1. Применение компьютерных и информационных технологий в организации учебного процесса

В последнее время большое внимание обращается на применение в образовании информационных технологий, которые стали неотъемлемой частью современных педагогических технологий. Компьютеры, телекоммуникации и сетевые средства изменили в серьезной степени методы освоения информации и играют большую роль в овладении целью обучения и его достижении.

Информация, усвоенная через компьютер, может быть явной или неявной. Во первых, знания определяются простейшим образом и могут быть переданы студенту по частям (текстовому, графическому, видео и т.д.). Во вторых, знания являются важнейшим компонентом, и они основываются на практике, интуиции и других. Эта часть знаний не передается непосредственно студенту, то есть они могут быть получены при решении практических задач в процессе самостоятельного образования [5].

Основываясь на эти мысли, можно определить образовательные программные аппаратные комплексы. Технологии, поддерживающие данный процесс, обычно являются декларативными, среди которых можно выделить:

- компьютерные программы;
- учебные базы данных;
- текстовые и контролирующие программы, а также другие компьютерные средства, которые дают возможность контролировать полное усвоение студентами информации, являющейся целью обучения, передачи и хранения такой информации.

Технологии, осуществляющие процесс усвоения скрытой части знаний, являются процедурными, компьютерные информационные технологии такой

категории не располагают частично разделенной информацией и не проверяют знания. В компьютерные информационные технологии такого порядка можно отнести:

- пакеты прикладных программ;
- компьютерные тренажеры;
- лабораторные работы.

Системы экспертного обучения дают возможность получения других компьютерных средств, необходимых знаний по предмету, которые получает студент в процессе обучения.

Вышеизложенное определение является условным, то есть один образовательный программный аппаратный комплекс может использоваться в связи с методикой, применяемой и в первой, и во второй технологии. Данные технологии предполагают получение студентом в определенной последовательности частичной информации (текстовой, графической, видео и так далее – все это зависит от технического уровня) и обеспечивают контроль со стороны учителя (учебной программы) усвоения учебного курса в определенных точках.

В качестве одного из программных средств осуществления программного аппаратного комплекса можно привести инструментальную авторскую систему.

Традиционные инструментальные системы можно разделить в зависимости от наличия тех или иных функциональных возможностей на универсальные и специализированные.

Универсальная инструментальная система обеспечивает следующие функциональные возможности:

- ввод и анализ ответов;
- формирование логической структуры компьютерных учебных программ;
- формирование и поддержка текстовых и графических материалов;
- организация гипертекстовых структур;
- сбор и обработка статистической информации;

- формирование рейтинговой оценки уровня знаний;
- возможность работы в локальной вычислительной сети.

При изучении профилирующего предмета использование инструментальной системы приводит к повышению заинтересованности студента к предмету, его способности к самостоятельной работе, достижения усвоения знаний и самое главное к повышению качества обучения.

1.2. Интеллектуальная система управления учебными процессами на основе Web – технологий

Известна организация УП (учебного процесса) (в том числе в высшем образовании), повышение качества и эффективности УП на основании современных информационных технологий [2]. Для этого в республике во многих высших учебных созданы технические и организационные возможности.

Основной целью использования современных информационных технологий в учебном процессе являются:

- 1) совершенствование качества подготовки кадров на основе современных информационных технологий;
- 2) повышение творческого интеллекта, составляющего учебную деятельность, с применением передовых методов обучения;
- 3) достижение интеграции разных форм учебной деятельности;
- 4) достижение обучающимися привыкания к учебе в индивидуальном порядке с помощью информационных технологий;
- 5) разработка новых информационных технологий обучения для эффективного использования в учебном процессе. При этом учитываются имеющиеся средства информационных технологий и принципы опоры на методы их использования и активации интересов обучающихся.
- 6) обеспечение непрерывности и последовательности обучения;

7) разработка или совершенствование информационных технологий дистанционного обучения;

8) совершенствование программно-методического обеспечения учебного процесса.

Вообще говоря, если поднять на высокий уровень значимость информации и если превратить информационную культуру человека в основной фактор, определяющий его профессиональную деятельность, то изменятся требования учебной системы в информационном обществе и повысится его статус.

Однако наряду с тем, что требования на программное обеспечение, адаптированное к виртуализации УП, создание такого программного обеспечения на основе Web-технологий предполагает специальный научный и методический подход. Потому что корпоративное использование имеющимися электронными учебными ресурсами дает возможность осуществления учебного процесса на уровне мировых стандартов [1, 3].

Одним из эффективных решений этой задачи является использование Web – технологий в учебном процессе.

Исходя из основных вопросов, рассмотренных выше, для применения Web-технологий в учебном процессе прежде всего следует сформировать полный и обновляющийся информационный массив учебно-методических материалов – знаний, навыков и квалификаций, необходимый для обучения и полностью его охватывающий, а также разработать программные средства для их использования в учебном процессе с помощью сетевого метода.

Вместе с тем, создание специального программного средства, объединяющего в себе информационные массивы, необходимые для УП (по предметам и учебным формам), и программные средства, предназначенные для их использования в сетевом методе – интеллектуальной системы, управляющей учебным процессом, является одним из самых актуальных задач.

При создании данной системы принимаются во внимание следующие основные принципы:

- модуль определения текущего состояния начальных знаний обучающегося по выбранному предмету (с помощью специальной тестовой программы);
- модуль анализа теоретических основ начальных знаний обучающегося (на основе результатов тестового контроля);
- модуль предоставления сведений на основе результатов анализа и предложения в виртуальном виде теоретических знаний, которые должны быть изучены (в режиме реального времени);
- модуль проведения тестового контроля уровня изучения предложенных теоретических знаний;
- модуль допуска в новые темы (знания), указанные в учебной программе, в случае удовлетворительного результата проведенного тестового контроля (не менее 75%);
- модуль проведения контроля на основе специальных заданий по теоретическим и практическим знаниям по каждой изученной теме и анализа результатов.

Предлагается на основе данного алгоритма поэтапно (по предмету в каждом блоке знаний) предоставлять обучающемуся теоретические и практические знания (на основе виртуальной технологии обучения).

Функциональная схема интеллектуальной системы управления учебным процессом на основе Web-технологий представлена на рис. 1.1. Здесь *Предметный сегмент* – массив информации по учебным предметам; S_i – обучающийся (студент, учение); T_i – модуль, передающий на узел управления результаты по каждой теме.

Элементы, составляющие массив информации по предметам, состоят из следующих:

1. тест, осуществляющий контроль начальных знаний обучающегося по предметам, база практических заданий и сборник теоретических сведений, охватывающий данную базу;

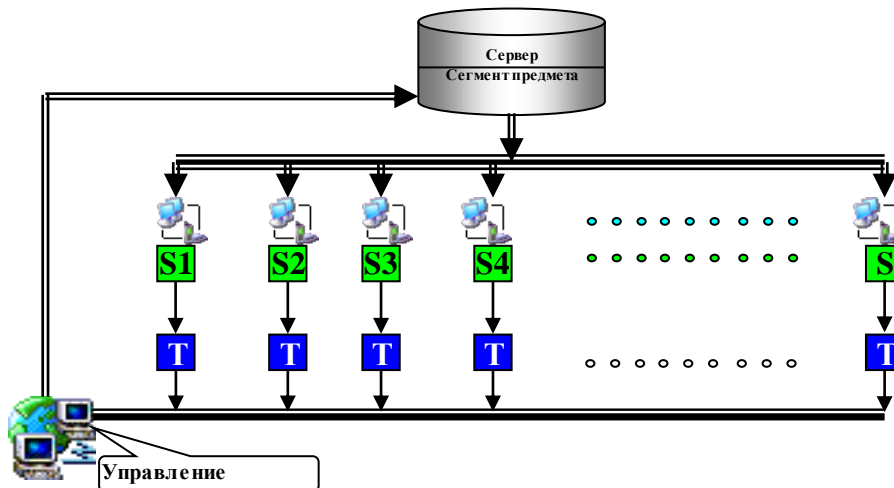


Рис. 1.1. Интеллектуальная система управления учебным процессом на основе Web-технологий

2. базы тем и практических заданий, соответствующих лекциям (N), практическим (A) и лабораторным (семинарам) занятиям (L) по блокам знаний, выделенных для учебного семестра предмета (отдельно для каждого блока);
3. база элементов минимальных знаний, навыков и квалификаций, которые должны быть изучены по видам занятий (опорные выражения по темам);
4. база тестово-контрольных вопросов для теоретических знаний, сборников задач для практических занятий и виртуальных лабораторий для лабораторных занятий по изучаемому занятию (в разрезе тем);
5. база данных об учителях предметов, создающих массивы информации;
6. база данных о зарегистрированных обучающихся;
7. база данных по результатам усвоения обучающихся;
8. комплекс программ, используемый при создании вышеуказанного массива информации и базы данных.

При входе обучающегося в систему система предоставляет обучающемуся данные по сегменту предмета на основе схемы, изображенной на рис 1.2.

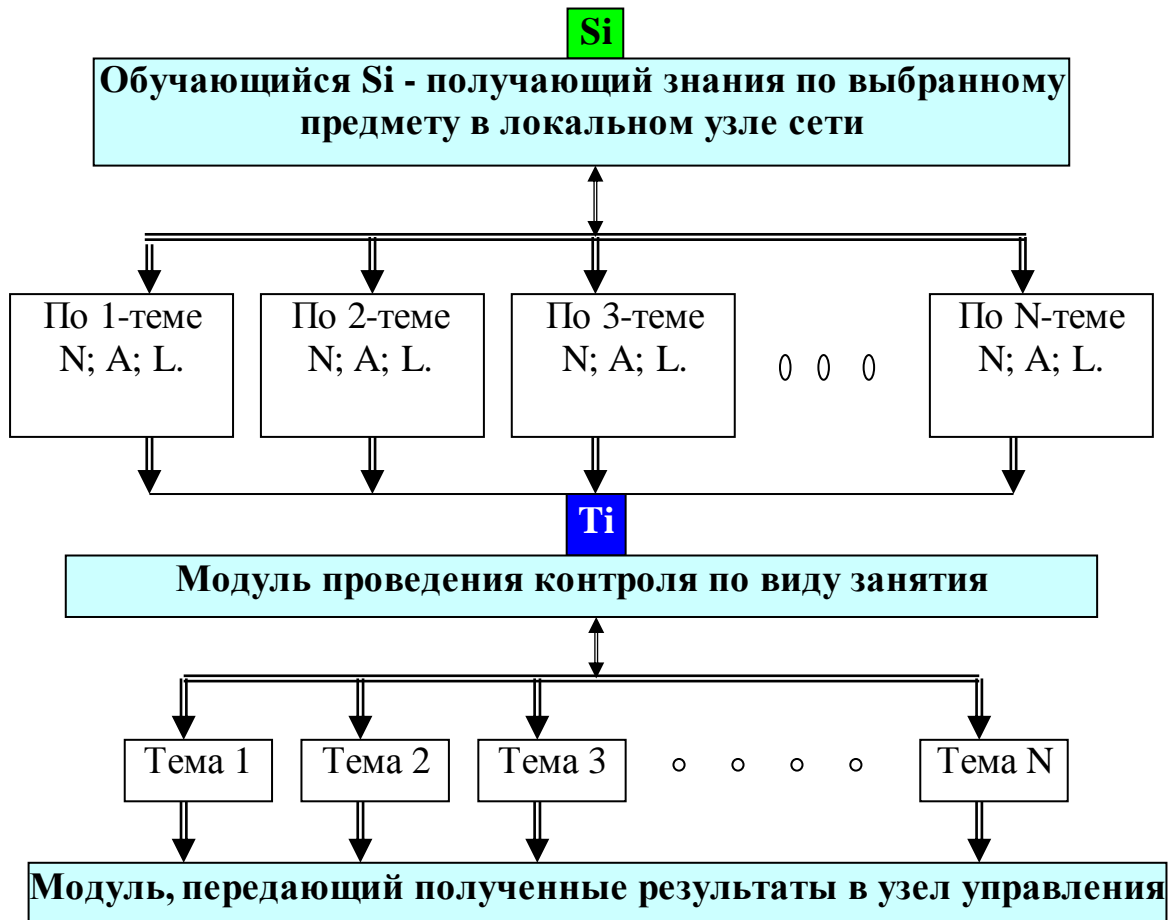


Рис. 1.2. Схема предоставления системой учебного материала обучающемуся и контроля результатов.

Программа, работающая на основе схемы 1.2, выполняет следующие действия:

1. объявление обучающемуся результатов контроля теоретических данных (N), практических занятий (A), лабораторных занятий (L) по каждому виду занятий, то есть вычисление и предоставление значения $T1 = \frac{N_{nat} + A_{nat} + L_{nat}}{3}$;

2. предоставление информации о комплексе знаний, оставшихся не изученными в соответствии с уровнем усвоения обучающегося (по темам и видам занятий);

3. передача в узел управления для записи результатов вместе с заключениями по анализу сектора, выделенного для обучающегося - учителя в сегменте предметов.

Специальное программное обеспечение на узле управления предоставляет обучающемуся следующий график усвоения по результатам данного обучающегося (рис 1.3).

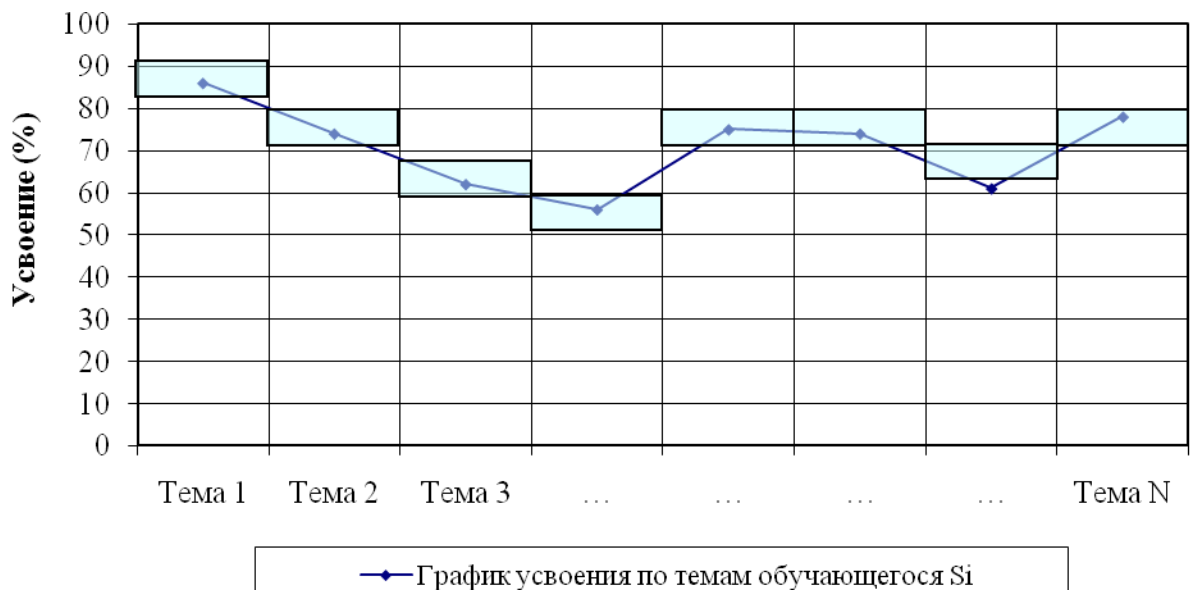


Рис 1.3. График, предоставленный системой для анализа степени усвоения обучающегося.

Вместе с тем система передает в сектор обучения результаты, объявленные обучающемуся, и сведения по неувоенным им знаниям вместе со сравнительным анализом имеющейся в массиве информации ситуаций в разрезе тем, видов занятий, поставленных вопросов тестового контроля и практических заданий.

Таким образом, на основе данной системы между преподавателем и обучающимся создается интерактивное общение, основанное на Web-технологиях, и в режиме реального времени решается вопрос осуществления теоретической и практической помощи, объективности определения уровня индивидуальных знаний, навыков и квалификаций ученика (студента).

Кроме того, обучающемуся предоставляются результаты анализа по занимаемому этапу обучения, а преподавателю – полная аналитическая информация о результатах обучающегося и его неусвоенных знаниях.

Учитель, использующий данную систему в процессе обучения, в первую очередь должен обратить внимание, во-первых, на метод и форму предоставления точных теоретических и практических знаний в области обучения, во-вторых, обладать знаниями об ученике (студенте), пользующимся системой.

В результате этого преподаватель планирует необходимость применения простых методов передачи сложных особенностей предлагаемых теоретических знаний или необходимость отдельного индивидуального подхода в передаче углубленных теоретических знаний, соответствующих интересам или устремленности обучающегося, то есть определяет учебную стратегию и предлагает его обучающемуся.

Этапы разработки программных средств, применяемых в общей деятельности предлагаемой интеллектуальной системы, и рекомендации по его использованию приведены в следующих главах диссертации.

1.3. Общее определение программной среды, используемой для организации учебного процесса на основе Web - технологий

Основным методом эффективного использования Web-технологий в УП считается создание интерактивных Web–узлов и Web–страниц, работающих в локальных сетях, и разработка их математического и специального программного обеспечения. В диссертации на основе создания технологий Web-узлов использованы алгоритмические языки высокого уровня. К подобным средам программирования входят Delphi, C++, Java, HTML, PHP, Apache-Server, и на основе их оптимальной комбинации предложены полностью автоматизированные виртуальные модели УП, которые могут функционировать в сети (на основе

системы клиент-сервер). В этой связи предлагаем общее определение вышеуказанных сред программирования. **Среда программирования Delphi (общие определения).** Программа на языке Delphi составляется в виде изображения алгоритмов, которые следует выполнять, если произойдет конкретное событие, связанное с формой.

Для каждого события, разрабатываемого в форме с помощью страницы инспектора Events, организуется процедура, и программист между ключевыми словами Begin и end вписывает требуемый алгоритм, составленный на языке Object Pascal. В первую очередь среда **Delphi** предназначена для профессиональных программистов, разрабатывающих корпоративные информационные системы. Однако среда понятна не только для профессиональных программистов, но и для пользователей, знакомых с любым языком программирования, и считается средой, удобной для изучения. Разница этой среды от компилятора Паскаля заключается в том, что в языке Delphi текст программы непосредственно транслируется в машинный код, в результате чего программное средство, созданное в Delphi, выполняется в 10-20 раз быстрее, чем средства, созданные на других языках.

Приложения с открытой архитектурой, созданные с помощью **Delphi**, работают устойчиво и надежно. Программы, написанные на языках Delphi, C и C++ и входящие в состав DLL, могут использовать имеющиеся объекты, а также сервер OLE, VBX и объекты, созданные на Delphi. Использование в рабочих приложениях готовых компонентов может осуществляться очень быстро и удобно. Эта библиотека состоит из стандартных объектов, составляющих интерфейс пользователя, объектов управления данными, графических объектов, мультимедийных объектов, объектов управления файлами, а также объектов управления DDE и OLE.

Основной интерфейс среды программирования Delphi всегда стоит на экране в рабочем состоянии (рис. 1.4) и используется для управления процессом создания программы.

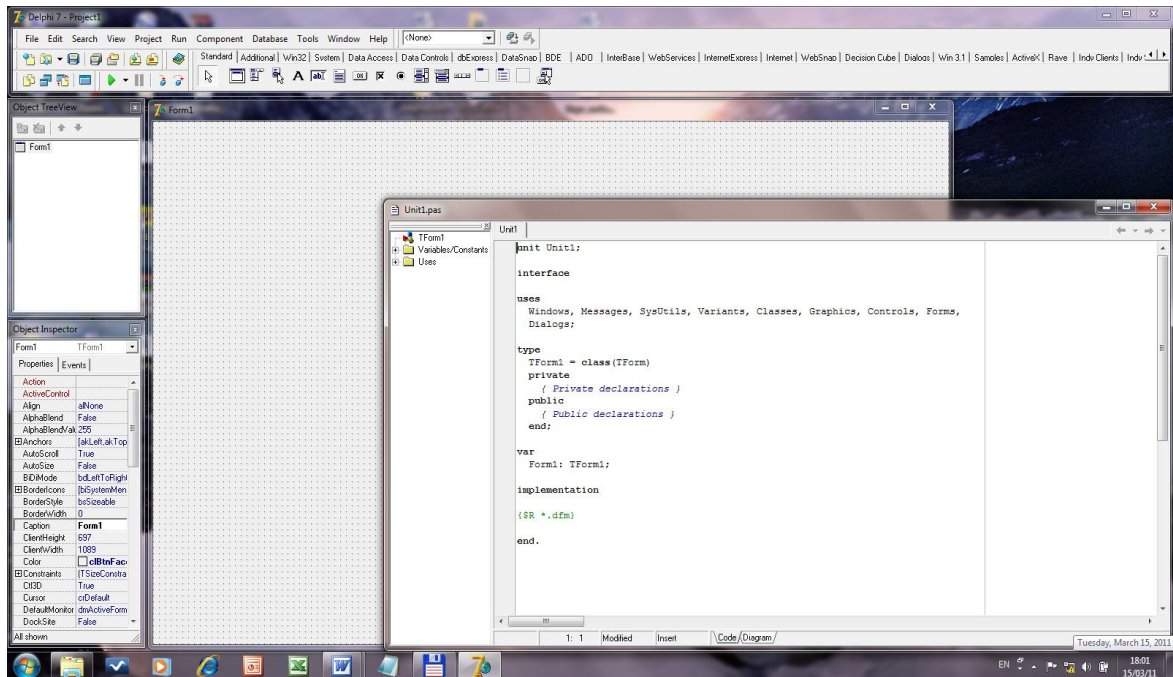


Рис. 1.4. Основной интерфейс среды программирования Delphi 7.

Пиктограммы облегчают доступ к часто используемым командам основного меню. С помощью меню компонентов осуществляется вход в стандартный набор сервисного приложения среды DELPHI, они изображают некоторые визуальные элементы (компоненты), размещенные программистом в окне формы. Каждый компонент имеет конкретный набор свойств. Например, цвет, заголовок окна, надпись внутри кнопки, размер шрифта и так далее. С помощью объектов библиотеки можно составить систему управления базами данных на основе архитектуры клиент-сервер. Эти объекты на низком уровне инкапсулируют компоненты Borland Database Engine в самих себя.

Язык C++. Язык Си был разработан в 1972 году Деннисом Ритчи для операционной системы Unix. При разработке данного языка программирования оно было спроектировано в виде инструмента системного программирования, и основное внимание было уделено структурным программам. Если с одной стороны, были сохранены все возможности программирования высокого уровня: определение типа данных, операторы for, while, if и другие операторы, то с

другой стороны, была выдвинута идея сохранения элементов языка Ассемблер: регистровые переменные, адресная арифметика, возможность работы с битовыми полями и другое.

Структура программы на языке Си следующая:

#директива препроцессора

.....

директива препроцессора

функция a()

операторы

функция b()

операторы

void main () // функция начала исполнения программы

операторы

определения

присваивание

функция

пустой оператор

составные операторы

выбор

повторение

переход

Директива препроцессора управляет отображением текста программы в его компиляцию. Текстовый файл программы вхождения, подготовленного на языке Си, проходит через следующие 3 этапа:

- 1) препроцессорное отображение текста;
- 2) компиляция;
- 3) компоновка (редактирование или сбор связей).

После этих 3 этапов оформляется исполняемый машинный код этой программы.

Задача препроцессора состоит в отображении текста программы в его компиляцию. Правила обработки препроцессора определяются программистом с помощью директив препроцессора. Директивы начинаются со знака #, например:

1) `#define` – директива препроцессора, определяющая правило замены в тексте.

`#define ZERO 0.0` – в случае такой директивы все имена `ZERO`, использованные в программе, заменяются на `0.0`.

2) `#include <имя файла заголовка >` выполняет роль вставки в текст программы текста из каталога “Заголовочных файлов”. При этом можно будет использовать стандартную библиотеку языка. Каждая функция библиотеки соответственно определяется в одном заголовочном файле. Список заголовочных файлов определяется в стандартах языка.

Связка с функциями в библиотеке осуществляется после компиляции на стадии компоновки. В заголовочном файле может быть приведено полное определение стандартных функций, однако с кодом программы могут быть связаны только с функцией, использованной в этой программе.

Обычно каждая функция, определенная в тексте программы, имеет свой заголовок. После заголовка функции размещается его основное тело. Тело функции начинается с фигурных скобок, которое состоит из различных определений, описаний и исполняемых операторов, и заканчивается фигурной скобкой. Каждое определение или оператор отделяется точкой с запятой.

Описание – используется для сообщения компилятору имен и свойств функций, объектов, встречающихся в других частях программы.

Операторы – определяют действия, выполняемые на каждом шаге программы.

Вообще язык программирования Си используется как объектно-ориентированный язык высокого уровня. На базе данного языка были

разработаны несколько современных языков и сред программирования. Например, на базе вышеприведенного языка были разработаны и внедрены в практику такие среды программирования как C++, Visual C++, Borland C++ Builder. Результаты, полученные в данной диссертации, были разработаны в среде программирования Borland C++ Builder. Основной интерфейс данной среды приведен на рис.1.5.

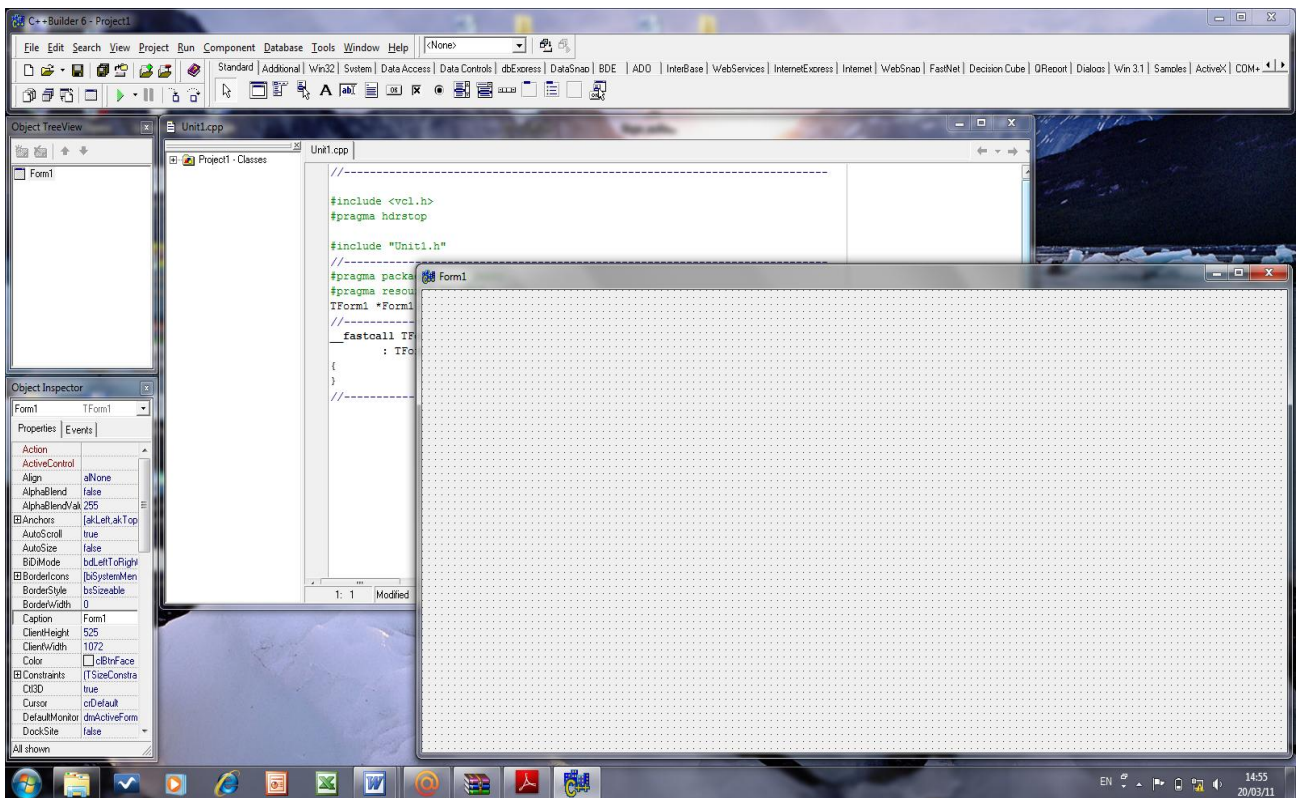


Рис 1.5. Основной интерфейс среды программирования C++ Builder 6.

Язык Java. Компьютерные программы или программное обеспечение представляют собой комплекс указаний, которые используются для решения определенной задачи. Программисты разрабатывают программное обеспечение с помощью специальных языков программирования Basic, C, C++, Java и так далее. В некоторых случаях при разработке разных программ приходится использовать несколько языков программирования. В настоящий момент в качестве «горячей темы» в мире компьютерных технологий признается Интернет и World Wide Web

(“Всемирная паутина”). В системе Интернет для обмена и обработки данными используются различные программные средства. Например, для управления данными в системе Интернет и их качественной доставки пользователю обычно используются специальные языки сценариев. Возникновение таких языков в свою очередь связано с языком Java, которое не связано с техническим и программным обеспечением компьютера. С помощью этого языка можно разработать программные продукты, работающие в сети в режиме распределения. В этой связи приведем основные свойства языка программирования Java, разработанного и внедренного в практику фирмой Sun Microsystems.

- Java – это язык программирования, предназначенный для создания апплетов, работающих с автономными программами и интернет браузерами;

- Апплеты, созданные в Java, работают в независимости от типа компьютера (технической платформы). Это свидетельствует о том, что один апплет работает в разных системах Windows, Macintosh или UNIX.

- Отличие языка Java от других языков программирования в том, что он не компилирует для конкретного процессора, а создает виртуальный машинный код. Этот код браузер превращает в бинарный код для конкретного процессора.

Значительной стороной этого языка является то, что наряду с развитием Веб-технологий управление информационными потоками приобретает сложный вид. В этом процессе для разработки самых удобных языков управления будет необходим единственный для всех язык. В качестве такого языка и предлагается Java.

Язык Java также является объектно-ориентированным языком программирования, на котором можно создавать автономные программы, похожие на язык C++ и апплеты, работающие под управлением браузера.

Возможности языка JavaScript. Язык JavaScript, разработанный фирмой Netscape, считается языком программирования для создания интерактивных Web документов. Этот язык направлен на создание пользователями специальных приложений для сервера и клиентов. С помощью этого языка можно создавать

приложения, исполняемые на сервер-компьютере, клиентские приложения, исполняемые на рабочих станциях.

Для создания обеих видов приложений используется ядро языка. Ядро считается множеством стандартных объектов, переменных, функций и основных объектов. Клиентские приложения разрабатываются непосредственно пользователем на рабочих станциях и интерпретируются напрямую на этом компьютере через Интернет браузеры. А приложения, разработанные для сервера, исполняются на этом сервере и служат для предоставления информации клиентам на основе запросов.

Составляющие языка JavaScript приведены на рис. 1.6.

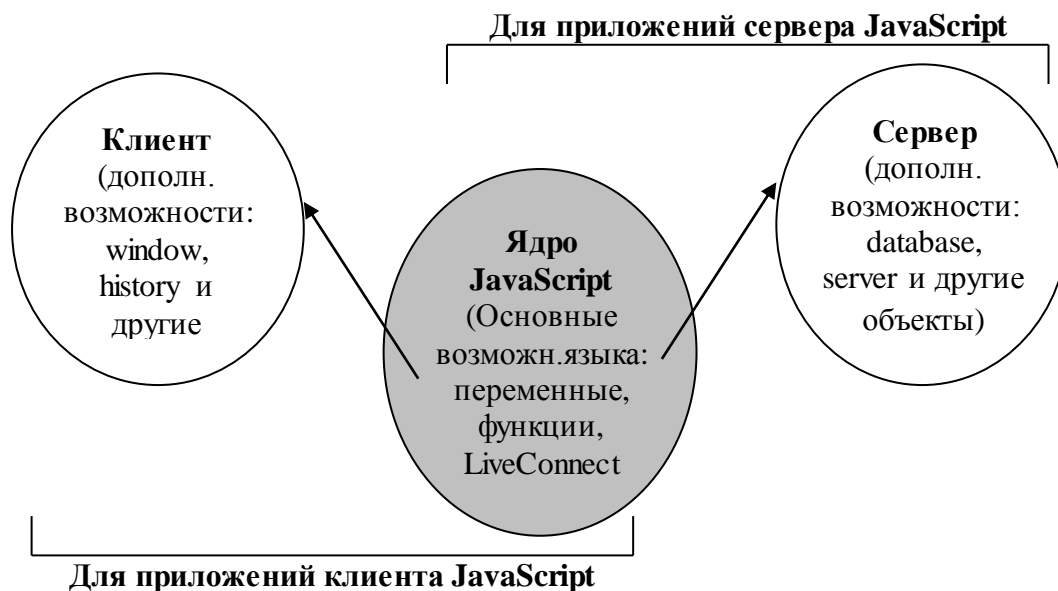


Рис. 1.6. Основные составляющие языка JavaScript и связь между ними.

Язык программирования JavaScript также как и другие языки имеет свой отдельный синтаксис, согласно которому каждый оператор заканчивается специальным разделяющим знаком “;”. Например, alert (“примечание”);

Для применения сценариев в составе HTML-документа используются специальные теги специального языка HTML. То есть они вписываются между специальными парами тегов <SCRIPT></SCRIPT>.

Если сценарии расположены в отдельных файлах, тогда в составе документов между вышеуказанными тегами вводится атрибут SCR.

Этот язык программирования, как и все другие языки программирования, поддерживает несколько типов данных. Эти данные следующих типов:

1. Целые;
2. Действительные;
3. Строковые (строчные, символьные);
4. Логические.

Примеры для целых типов: 123, -123, +123.

Данные в действительном формате предоставляются в виде десятичной дроби (плавающий тип данных): 1.25, 0.125e01, 12.5e-1 ва т.д.

Строковые данные состоят из последовательности алфавитных символов, для обработки которых используются кавычки.

Данные логического типа считаются данными, принимающими два значения: истина или ложь.

Возможности PHP. В 1994 году знаменитый программист Расмус Лерддорф (Rasmus Lerdorf) объявил о выходе нового языка программирования PHP (Personal Home Page). На сегодняшний день PHP (Processor Hyper Text) считается средством программирования, который используется во множествах Интернет-серверах и стремительно развивается. В настоящий момент также развитыми считаются ASP, FrontPage и mod_perl. С помощью этого языка можно быстро и легко создать динамическую страницу. Файлы, созданные таким образом, хранятся на серверах и обрабатываются. Когда пользователь запрашивает PHP документ, такие скрипты как JavaScript исполняются не на браузерах, а на сервере и результат их работы отправляется пользователю. А это выполняет программа CGI, написанная на C или Perl. Отличие языка PHP от программ CGI заключается в том, что код PHP можно написать в любом месте HTML страницы. Сценарии (коды) PHP используются в составе HTML-документа и выполняются сервером.

Основной синтаксис написания PHP-сценария следующий.

1-синтаксис. Синтаксис, использующий стиль XML:

```
<?php Здесь пишется код на языке PHP?>
```

2-синтаксис. Синтаксис в стиле HTML:

```
<scriptlanguage="php">
```

Здесь пишется код на языке PHP

```
</script>
```

3-синтаксис. Синтаксис в стиле ASP:

```
<% Здесь пишется код на языке PHP%>
```

4-синтаксис. Кратко выраженный синтаксис:

```
<? Здесь пишется код на языке PHP?>
```

Кроме того разрабатываются другие языки и программные среды, такие как Perl, C/C++, Фортран, TSL, UnixShell, VisualBasic, AppleScript, которые работают на основе CGI (CommonGatewayInterfase – общий шлюзовый интерфейс, считающийся стандартом написания HTTP приложений)

1.4. Цель диссертации и его основные задачи

В результате вышеуказанных сведений, их анализа, а также обработки научно-практической информации, изложенной в литературе, была определена цель и основные задачи диссертации. Целью диссертации является разработка математических моделей, в частности, семантической модели УП и программного обеспечения виртуализации учебных процессов на основе Веб-технологий и их внедрение. В рамках реализации цели диссертации предусмотрено решение следующих основных задач:

- на основе общего определения электронно-учебных ресурсов формирование их в качестве целого объекта;

- разработка дискретных, семантических и многоэтапных математических моделей учебного процесса, определение принципов их устойчивости и адекватности;

- разработка соответствующего программного обеспечения и внедрение в образовательный процесс электронно-учебных ресурсов на основе Web-технологий;

- разработка специального программного интерфейса пользователя по виртуализации учебных процессов, программного обеспечения проведения дистанционного тестового контроля;

- создание программного обеспечения автоматизированной системы, тестирующей программы, созданные на основе Web-технологий, и создание виртуальной модели преподавания предмета “Информатика”;

- создание в разрезе предметов системы организации внедрения в учебный процесс программного обеспечения, основанного на Web-технологиях.

Выводы по I главе

1. На основе системного анализа проанализированы возможности информационно-коммуникационных технологий в организации виртуализации учебных процессов, на их базе была разработана базовая концепция по их использованию в учебных процессах.

2. На основе Web-технологий была создана интеллектуальная система управления учебными процессами и были определены принципы их внедрения.

3. Определены программные среды, используемые для виртуализации учебных процессов на базе Web-технологий, с использованием семантических и математических моделей.

4. Была сформирована цель магистерской диссертации и основные задачи, решаемые в его рамках.

II ГЛАВА. АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СЕМАНТИЧЕСКОЙ МОДЕЛИ ОБРАЗОВАТЕЛЬНЫХ ПРОЦЕССОВ

2.1. Общее определение электронных учебных ресурсов

Использование информационно-коммуникационных технологий в управлении учебного процесса расширяет возможности повышения эффективности обучения. Одной из основных причин этого является то, что при этом создается возможность полного контроля учебного процесса и его обновления. Например, сведения, передаваемые по локальной сети образовательного учреждения, ведут к сокращению объема документов, что в свою очередь создает возможность экономии времени при принятии управленческих решений в учебном процессе.

То, что на сегодняшний день одним из приоритетных направлений на пути развития нашего общества является инвестирование системы образования в целях подготовки знающих национальных кадров и обеспечение развития информационно-коммуникационных технологий, подчеркнуто в «Национальной программе подготовки кадров». Одним из важных направлений внедрения информационных технологий в учебный процесс является создание электронных учебников, пособий и курсов.

Несомненно, что основу системы обучения составляет компьютер, информационные и педагогические технологии. Следует особо подчеркнуть, что использование информационно-коммуникационных технологий в учебном процессе требует создания базы данных в виде электронной формы знаний, навыков и квалификаций и разработки автоматизированной системы контроля знаний, полученные на основании этой базы данных. В данной главе диссертационной работы рассмотрены алгоритм и программное обеспечение получения решения задачи о разработке электронных ресурсов, предназначенных для использования в учебном процессе.

При организации процесса образования на основе информационно-коммуникационных технологий особое значение имеет представление на должном уровне его составной части и его проектирование (рис. 2.1).

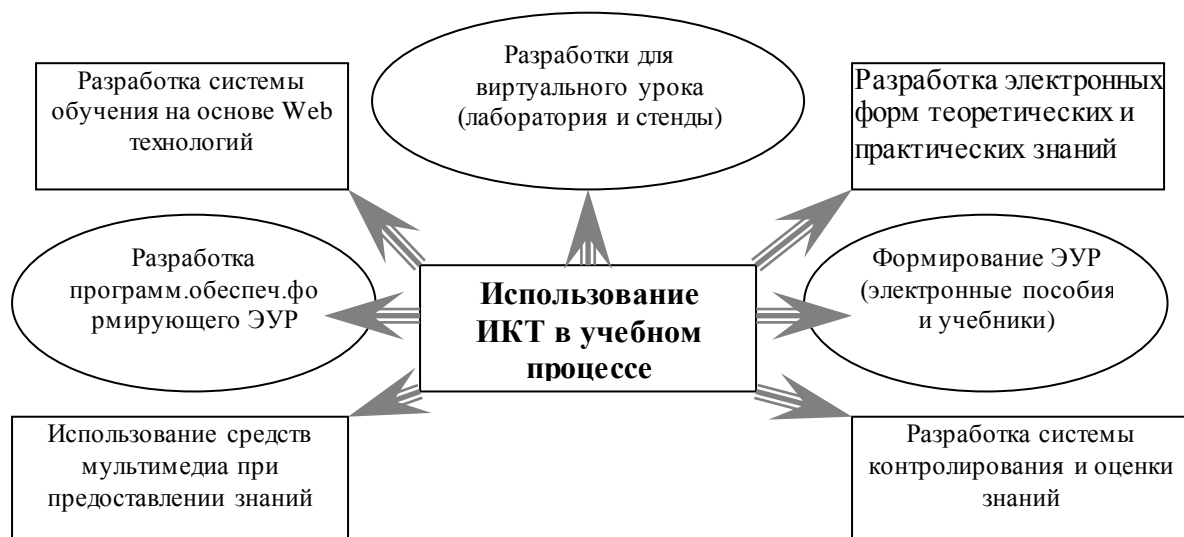


Рис. 2.1. Применение в учебном процессе ИКТ (составляющие)

Таким образом, при применении в учебном процессе ИКТ основное внимание уделяется разработке электронных форм знаний. При подготовке электронно–учебных ресурсов следует иметь полное представление о сведениях, предоставляемых обучаемым (СПО), или принимать во внимание требования, предоставляемые к их проектированию. В этой связи дадим определение для учебно-нормативных документов, являющихся основой для формирования знаний.

Учебно-методический комплекс (УМК) включает в себя учебно-методические источники, дидактические средства и материалы, электронные образовательные ресурсы, технологии обучения, методы и нормы оценки, направленные на формирование знаний, навыков и квалификаций, которые должны быть приобретены на основании требований, определенных в учебной программе, и государственных стандартов образования, на получение гарантированных результатов на основании комплексного проектирования

учебного процесса, на обеспечение осуществления самостоятельного образования и контроля, на развитие творческих способностей студентов.

Содержание компонентов учебно-методического комплекса предмета разрабатывается согласно учебной программе, созданной на основе Государственного стандарта образования, а также на основе технологий, принципов и требований, направленных на развитие личности и получение самостоятельного образования [3].

Исходя из этих требований общую структуру разрабатываемого учебно-методического комплекса можно представить на рис. 2.2.



Рис. 2.2. Общая структура учебно-методического комплекса.

Если отдельно провести структурный анализ компонентов, составляющих данную структуру, необходимо будет сформировать следующие блоки, как указано на рис. 2.3, 2.4, 2.5.

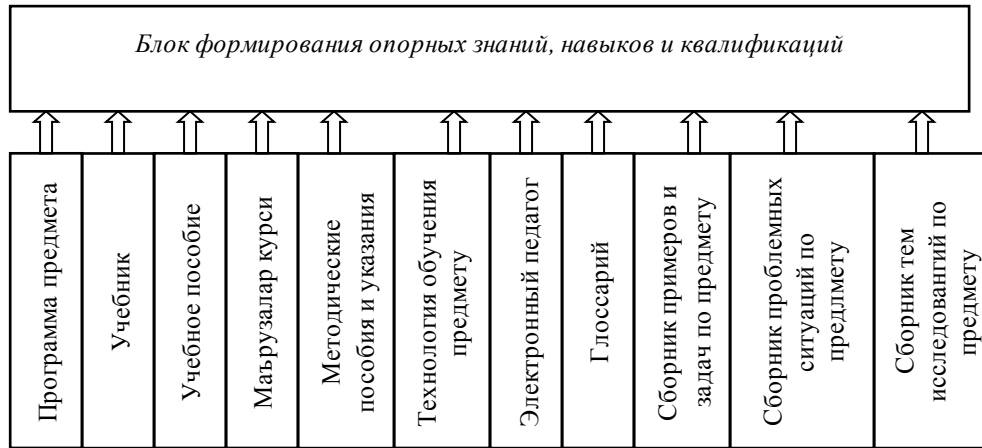


Рис 2.3. Составляющие блока формирования опорных знаний, навыков и квалификаций

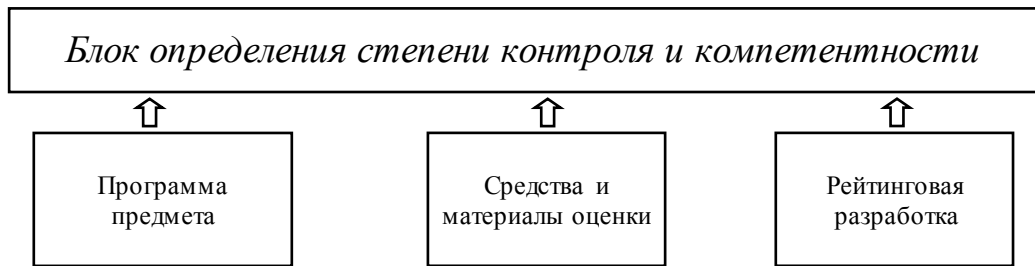


Рис. 2.4. Составляющие блока определения степени контроля и компетентности.

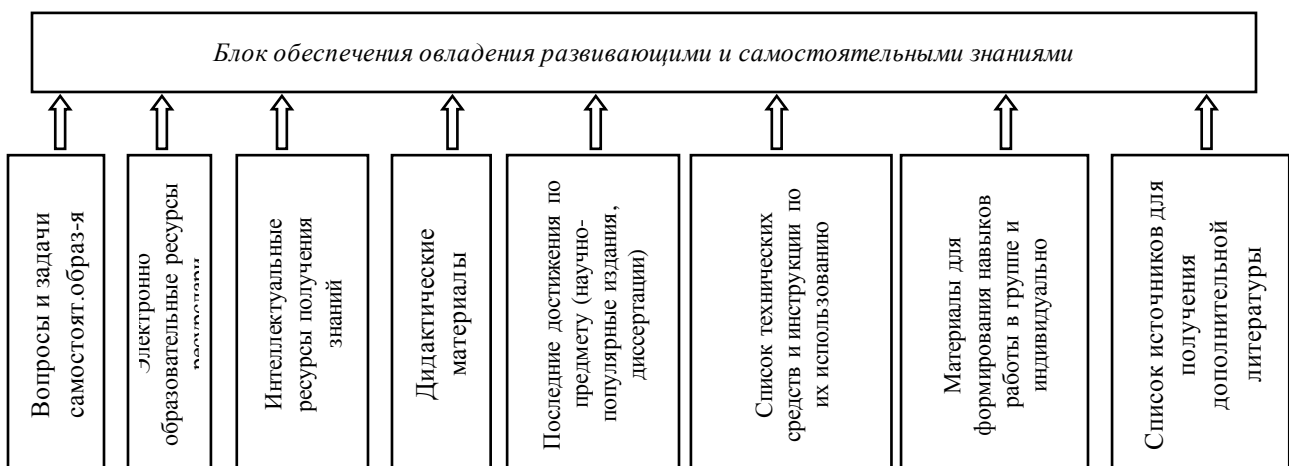


Рис. 2.5. Составляющие блока обеспечения овладения развивающими самостоятельными знаниями.

С помощью вышеуказанных рисунков был разработан полный проект составляющих учебных материалов и будут созданы электронные учебные ресурсы. А эти электронные учебные ресурсы должны иметь следующий вид (Рис. 2.6).



Рис. 2.6. Общий проект электронно-учебного ресурса.

Рекомендации по разработке электронно-учебного ресурса:

Обеспечить соответствие с Государственным образовательным стандартом по направлению (специальности);

Разработка состава создаваемого учебно-методического комплекса. Каждый компонент должен быть разработан на основе личностно ориентированных,

развивающих и самостоятельных принципов и требований, исходя из специальных требований, предъявляемых к предмету;

Определение сроков создания УМК и его реализация;

Определение формы и структуры компонентов для сохранения единого подхода.

В целях автоматизации формирования учебно-методического комплекса, разработанного (созданного) на основе данных рекомендаций, была разработана специальная программная система. Общая характеристика такой программной системы приведена в следующих параграфах.

2.2. Дискретные математические модели виртуализации учебных процессов

Характерной стороной современного развивающегося общества является то, что оно требует постоянной модернизации. Потому что, если постоянно развивается одна ячейка общества, другая ячейка остается без развития. Отсюда следует, что наряду с процессом производства следует также анализировать и моделировать другие социальные процессы и на основе полученных результатов определять новую стратегию развития. [1].

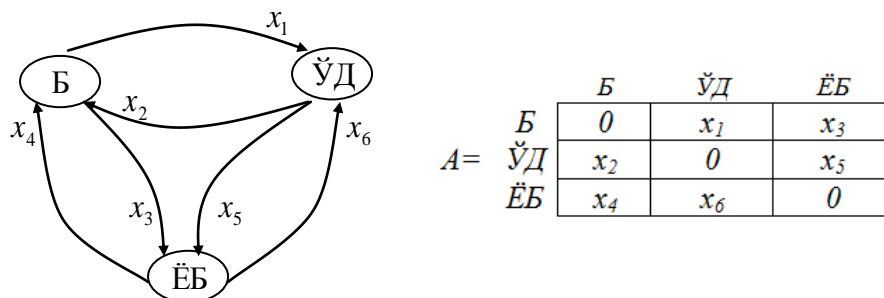
Виртуализация учебных процессов выполняется на основе их математических моделей, алгоритмического и программного обеспечения. Получение знаний учеником (группой учеников) в виртуальном процессе обучения может вестись по следующим направлениям:

- учебный процесс на основе жесткого сценария;
- выбор учеником определенного сценария и процесс получения знаний на основе адаптированного алгоритма.

Основной целью, предусмотренной этими направлениями виртуализации учебного процесса, является оптимизация процессов получения знаний. При этом

оптимизация обеспечивается по двум направлениям, то есть минимизация времени, расходуемого на учебный процесс, или максимализация степени усвоения в течение фиксированного времени.

При виртуализации учебного процесса интеллектуализация диалога со студентами связана с особенностями изучаемого предмета. Оптимизация, определенная выше, полностью зависит от содержания и формы математической модели учебных процессов. В данном исследовании рассмотрена задача составления дискретной модели виртуального учебного процесса на основе составления направленных графов. Для этого введем следующие определения, касающиеся направленных графов. Самый простой график виртуализированного учебного процесса можно определить следующим образом.



$$A = \begin{matrix} & \begin{matrix} Б & ЁД & ЁБ \end{matrix} \\ \begin{matrix} Б \\ ЁД \\ ЁБ \end{matrix} & \begin{matrix} 0 & x_1 & x_3 \\ x_2 & 0 & x_5 \\ x_4 & x_6 & 0 \end{matrix} \end{matrix}$$

Рис. 2.7. Направленная графа виртуализированного учебного процесса и матрица связанности.

При этом Б - комплекс знаний, данных, задач и вопросов; ЁД - степень усвоения учеником полученных знаний и данных; ЁБ - комплекс данных, помогающих усвоению; А – матрица связанности направленной графа; $x_1, x_2, x_3, x_4, x_5, x_6$ – переменные связанности концов направленной графа.

Теперь определим количественные показатели концов направленной графа. СУ – степень усвоения определяется следующим образом: $СУ = ПО/Б$, здесь ПО - количество правильных ответов ученика. Этот показатель может меняться в пределах (0,1). ВС – вспомогательная степень, которое также меняется в пределах (0,1).

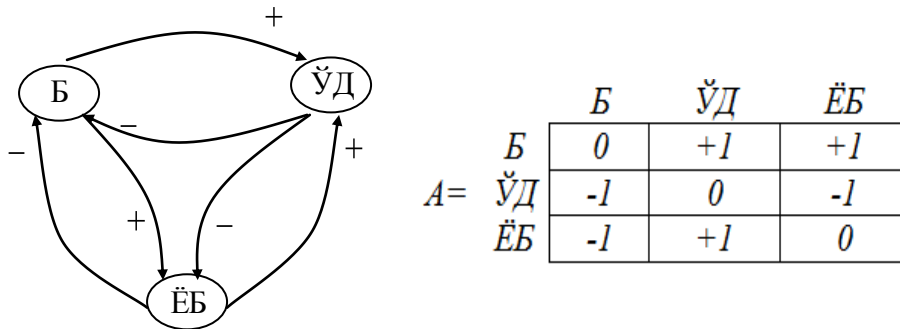


Рис. 2.8. Пример знаконаправленной графа и ее матрица связанности

Края (ребра) направленной графов определяются численными параметрами x_k и согласно вышеприведенному примеру обозначаются как $X=(x_1, x_2, x_3, x_4, x_5, x_6)$. При этом x_k могут быть определены согласно знакоопределенному примеру, дуги, соединяющие концы, обозначаются знаком +. В результате вышеуказанная направленная графа может быть выражена с помощью введенных знаков следующим образом.

Значит при составлении моделей виртуализированного учебного процесса на начальном этапе можно использовать направленные и знаковые графы. Графы для сложных структур виртуальных учебных процессов можно определить с помощью добавления следующих факторов (концов графов) в вышеуказанную структуру: ОМ – требуемый объем усвоения учебного материала; СМ – степень сложности учебного материала; УС - возможности усвоения студентов.

При этом методе создание модели эксперта или группы экспертов можно начинать на начальном этапе. Можно хотя бы реализовать связь между различными характеристиками изучаемого учебного процесса на качественном уровне. Сравнительно сложная функциональная модель виртуализированной системы обучения приведена на рис. 2.9. Здесь добавлены такие концы как объем изученного материала (ОМ), степень сложности материала (СМ), степень одаренности ученика (УС). Эти значения, характеризующие входящие показатели моделируемого процесса, можно также разместить в пределах интервала [0,1]. В

указанной модели эти параметры оказывают влияние только на концы В, УО и П, однако в них отсутствует обратная связь. Также нет связи влияющей на концы ОМ, СМ, УС. С помощью обыкновенного анализа можно определить, что ребра x_7, x_8, \dots, x_{15} будут иметь знаки $x_7^+, x_8^+, x_9^-, x_{10}^-, x_{11}^-, x_{12}^+, x_{13}^+, x_{14}^+, x_{15}^-$.

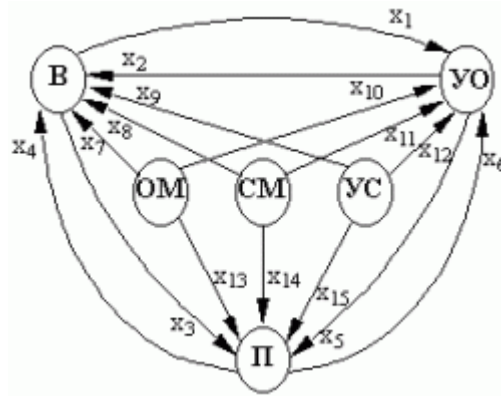


Рис. 2.9. Пример сложной модели виртуального учебного процесса.

Для анализа виртуализированной модели обучения, представленной с помощью направленных графов, необходимо будет составить алгоритм, влияющий на изменение значения одного конца при его замене на значение другого конца. Суть этого в том, что с помощью влияния на анализируемый конец графа изменяется его значение, то есть его значение увеличивается или уменьшается. Например, в параметр на конце В графа можно ввести данные или вопросы, касающиеся изучаемой темы.

В дальнейшем, при изучении влияния этого добавленного значения на значения на других концах графа создается импульсный процесс. Эту закономерность можно определить с помощью следующей дискретной модели. Для этого введем следующие общие обозначения. Каждый конец u_i направленного графа с концами u_1, u_2, \dots, u_n в результате импульсного процесса в течение дискретного момента времени $t = 0, 1, 2, \dots$ получит значение $v_i(t+1)$, которое будет содержать сведения об импульсных изменениях u_j , связанных с u_i . Для определения значений, образованных на концах направленного графа в течении импульсного процесса, рекомендуется использовать следующую формулу:

$$v_i(t+1) = v_{i(t)} + \sum_{j=1}^n w(u_j, u_i) p_j(t) \quad (2.2.1)$$

Здесь $w(u_j, u_i)$ – дуги, начинающиеся с конца u_j и заканчивающиеся на конце u_i . В частном случае, если не существует дуги (u_j, u_i) , то $w(u_j, u_i) = 0$.

$p_j(t)$ – изменение значения на конце u_j в течении времени t .

Согласно вышеприведенному соотношению, то есть согласно формуле (2.2.1) если взять в качестве доли дуги, направленной из конца u_j в конец u_i как w , тогда в течении времени t увеличению значения u_j на z будет соответствовать повышение в течении времени $t+1$ значения на конце u_i на $z \cdot w$.

В дальнейшем, для каждого конца u_i рекомендуется различать значения состояния $v_i(x)$ и начального значения $v_i(0)$ и на основании этого предлагается следующая закономерность:

$$v_i(0) = v_i(x) + p_i(0), \quad (2.2.2)$$

здесь $p_i(0)$ – начальный импульс конца u_i (в момент $t=0$).

На основании вышесказанного введем следующие векторные выражения:

$V(x) = (v_1(x), v_2(x), \dots, v_n(x))$ – векторное значение концов графа относительно состояния;

$P(0) = (p_1(0), p_2(0), \dots, p_n(0))$ – начальный вектор импульса;

$V(t) = (v_1(t), v_2(t), \dots, v_n(t))$ – векторное значение концов графа в момент t .

На основании введенных обозначений и выражений, а также учитывая соотношение (2.2.2) алгоритм развития импульсного процесса можно выразить следующим образом:

$$V(t) = V(x) + (I + A + A^2 + \dots + A^t)TP(0) \quad (2.2.3)$$

здесь I – $(n \times n)$ -мерная единичная матрица;

A – $(n \times n)$ -мерная матрица связанности направленной графа;

T – знак транспонирования, t – степень.

Значит, согласно введенным обозначениям и соотношению (2.2.3), в качестве дискретной математической модели учебного процесса на основании отношений (2.2.1)-(2.2.2) можно выбрать процесс импульсного изменения выбранного графа.

Предложенные импульсные процессы могут быть устойчивыми и неустойчивыми. При этом неустойчивость импульсных процессов может привести к бесконечному увеличению или бесконечному уменьшению значений взаимосвязанных концов графа. То есть, если несмотря на повышение базы знаний, данных и вопросов размещенных на конце В, не обеспечить потенциал усвоения студентами значения всех концов направленной графа резко уменьшаться. Устойчивость импульсного процесса объясняется асимптотическим приближением значений концов направленной графа к известному постоянному значению. Различие в устойчивости или неустойчивости определяется знаком параметра X и численными значениями дуг направленной графа.

В качестве примера (Рис. 2.10) рассмотрим случай, когда развитие неустойчивого импульсного процесса относительно концов УО и П изображено согласно параметрам $X=(0.25, -1, 0.3, 0.1, -1, 1)$, $V(x)=(0,0,1)$, $P(0)=(5,0,0)$.

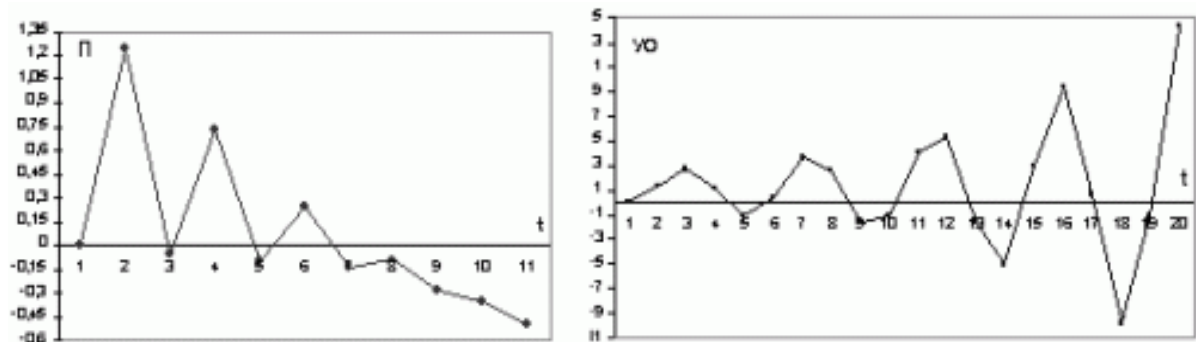


Рис. 2.10. График развития неустойчивого импульсного процесса.

Определение оптимальных значений концов и дуг направленных графов виртуализированного учебного процесса на основании моделей устойчивых импульсных процессов имеет большое научное и практическое значение. В этом направлении доказано, что для устойчивости импульсного процесса графа каждое абсолютное значение в его матрице связности не должно превышать 1. Поэтому во многих случаях при оптимизации виртуального учебного процесса целесообразным считается использовать вышеприведенные алгоритмы, создавать

компьютерные модели на базе условий устойчивости соответствующих им импульсных процессов.

Принимая во внимание данные соображения при разработке нового методического средства для моделирования учебного процесса необходимо учесть следующие аксиомы:

1. Учебный процесс считается недетерминированным случайным процессом;
2. На основании рекомендаций, выданных согласно изменениям параметров курса на базе материалов, изученных во время обучения, проводятся промежуточные контроли;
3. Учебный процесс считается управляемой системой на основе обратной связи. [3].

Рассматриваемый учебный процесс считается случайным процессом, основные характеристики которого можно оценить, и можно дать рекомендации по изменению параметров курса. В качестве основной характеристики процесса можно взять количество определений, формул и других дидактических единиц (ДЕ), а также время изучения темы, последовательность обучения и вероятность усвоения темы.

Состояние процесса обучения определяется как неопределенные параметры $ws(n)$ плотности распределения вероятности (ПРВ). Отбираются некоторые гарантированные определения процесса обучения (это могут быть минимальные требования, предъявляемые к усвоению ученика), соответственно в качестве самого плохого ПРВ берутся значения $ws(n)^*$, принадлежащие i -му множеству $\{ws(n)\}$. Принято такой метод считать устойчивым или прочным [4]. Несомненно, для обеспечения выполнения минимальных требований Государственной программы по предмету необходимо включить эти методы в начальный этап проектирования учебного курса.

Одной из возможностью формализации учебного процесса считается непрерывность времени и связь между дискретными событиями и случайными марковскими процессами.

В таких процессах в качестве плотности распределения вероятности λ (плотность вероятности при переходе процесса из одного состояния в другое) играет последовательную роль. После прохождения каждой темы проводится текущий контроль усвоения учениками пройденного материала, по результатам контроля принимается решение о переходе на следующую тему или проведение дополнительного занятия (в частности самостоятельного занятия) по предыдущей теме. Результаты текущего контроля, проведенного по успешно усвоенным темам курса, состоят из элемента, характеризующего случайность вероятности P . Статистическая оценка этой вероятности будет служить основой для успешного прохождения учеником очередного промежуточного контроля.

Для отображения процесса составляется дуговая графа состояния (рис. 2.11). Это состояние рассматривается как вектор, вобравший в себя долю ученика, успешно прошедшего промежуточный контроль. Оно зависит от модели сложности усвоенных тем, количества дидактических единиц темы, степени усвоения тем учениками и других возможных характеристик.

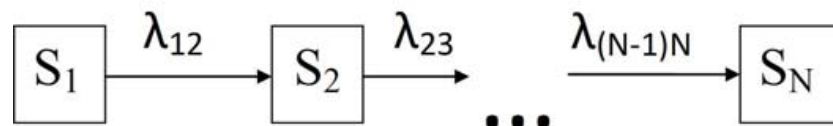


Рис. 2.11.

В зависимости от типа управления выбранных педагогических процессов, множество начальных условий выражает вектор законченных состояний или после каждого промежуточного контроля выражает начальные значения анализируемой последовательности обучения.

Вид процессов обучения в качестве системы управления приведено на рис. 2.12.

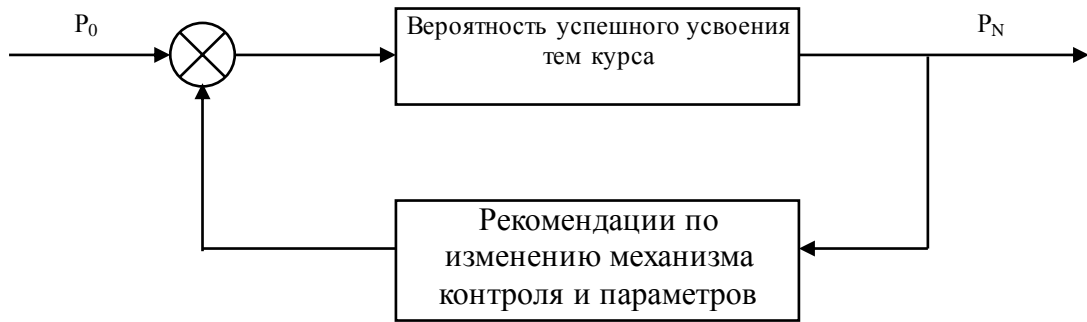


Рис. 2.12.

Данная модель обучения работает по следующему образцу: результаты (решения) заданий, выполненных учениками, передаются в блок контроля, и даются рекомендации по изменению параметров курса. Блок контроля на основании анализа ответов выделяет ученика или группу учеников, которым следует заново сдать контроль, определяет для них параметры обучения дополнительных учебных занятий. Ученикам, имеющим успешные результаты контроля, предлагается перейти к следующей теме.

Точно также для управления педагогическим процессом необходимо определить следующее:

1. Определить модели знаний, навыков и квалификаций по результатам выполнения контрольных заданий и для рекомендаций;
2. Создание базы данных предмета, которую требуется изучить.

База данных предмета оформляется на основании ДБ и специальных заданий, которые содержат их в себе. Модель определения знаний формируется в виде вопросов, выбранных случайным образом из базы данных. Точно также в эту модель добавляется и множество взаимосвязанных заданий, предназначенных для определения ДБ, которые еще не изучены и рекомендуются для повторного обучения.

Для выхода из положения повторения неудовлетворительных результатов требуется уменьшить объем ДБ и оптимизировать количество заданий [5]. Поэтому рассматривается множество Ω , мощность ДБ которого равно N .

Предположим, что задано m заданий, состоящих из w дидактических единиц. Через w_{ij} обозначим i -ую дидактическую единицу, которая была использована при составлении j -той задачи или вопроса. Тогда $n_j = \sum_{i=1}^N w_{ij}$ ($j=1..m$) примем в качестве отношения, характеризующего количество ДБ и выражающего j -ую задачу. А относительное количество используемых ДБ выразим через отношение $\rho_j = \frac{n_j}{N}$.

Время, отводимое для выполнения каждой задачи, обозначим через C_j , а время, выделенное для выполнения всех заданий, выразим через $C_{общ}$.

В качестве целевого показателя выберем относительное количество ДБ, используемых в задачах учебной программы, а в качестве границы – общее время, отводимое для выполнения задачи.

Через $X_j = \begin{cases} 1 \\ 0 \end{cases}$ обозначим индикатор добавления задач в учебную программу, а через $X = [X_1, X_2, \dots, X_m]$ - вектор индикаторов конкурирующих задач, которые также добавляются в учебную программу. Тогда $C_{II} = \sum_{j=1}^m C_j X_j$ будет выражать общее время, расходуемое на нахождение решения задачи, связанной с задачами, которые были дополнительно добавлены в учебную программу. Количество $\rho(X) = 1 - \prod_{j=1}^L (1 - \rho_j X_j)$ будет выражать относительное количество ДБ, использованных в данной учебной программе.

Для введения в учебную программу требуется составить алгоритм выбора конкурирующих задач (или вопросов), которые применяются в рамках времени, выделенной для решения задач, и который вмещает в себя максимальное относительное количество ДБ, предназначенных для ввода в учебную программу.

$$\rho(X) = 1 - \prod_{j=1}^L (1 - \rho_j X_j) \rightarrow \max_{X_j \in X};$$

$$C_{II} = \sum_{j=1}^m C_j X_j \leq C_{умзм, X_j \in \{1,0\}}.$$

Оформляемая задача выражает задачу программирования нелинейных булевых переменных. Для решения этой задачи необходимо использовать специальные математические методы и средства программирования.

В этой связи возникает необходимость построения с помощью предоставленных математических моделей оптимального курса, в котором можно будет использовать априорные данные учебной программы и данные требования, а также необходимость анализа по объему ДБ и результатам последовательности промежуточного контроля. А это обеспечит качество и эффективность процесса обучения.

2.3. Семантическая модель электронных учебных ресурсов

При разработке электронных учебных ресурсов требуется в первую очередь определить их общую структуру. То есть при проектировании электронных учебных ресурсов должны решаться задачи размещения материалов, составляющих их сущность, и определения порядка их взаимосвязи. Через разработку семантической модели электронных учебных ресурсов создается последовательность конкретных указаний по их использованию.

При разработке семантической модели электронных учебных ресурсов рекомендуется применить следующий принцип:

1. Выделение содержания (контента) данных, составляющих электронные учебные ресурсы в множество структурных элементов, то есть $E = \{e_1, \dots, e_n\}$;

2. Типизация структурных элементов в виде $G: E \rightarrow T$. Здесь $T = S \cup C \cup L$ - множество видов, S - множество видов семантических элементов, соответствующих дидактической единице (ДЕ) контента, например, “лекция”, “определение”, “теорема”, “термин”, C - множество видов элементов мультимедиа, например, “текст”, “графика”, “анимация”, “видео”, L - множество

видов элементов семантической связи, используемых для установления логической связи, например, “исходя”, “определяется”, “выражает”.

3. Иерархическое упорядочивание контента, определяющего внутри структурных элементов отношение «часть одного целого». Иерархия определяется через отображение $F: E \rightarrow 2^E$, которое каждому структурному элементу e_i сопоставляет множество его производных элементов (рис. 2.13).

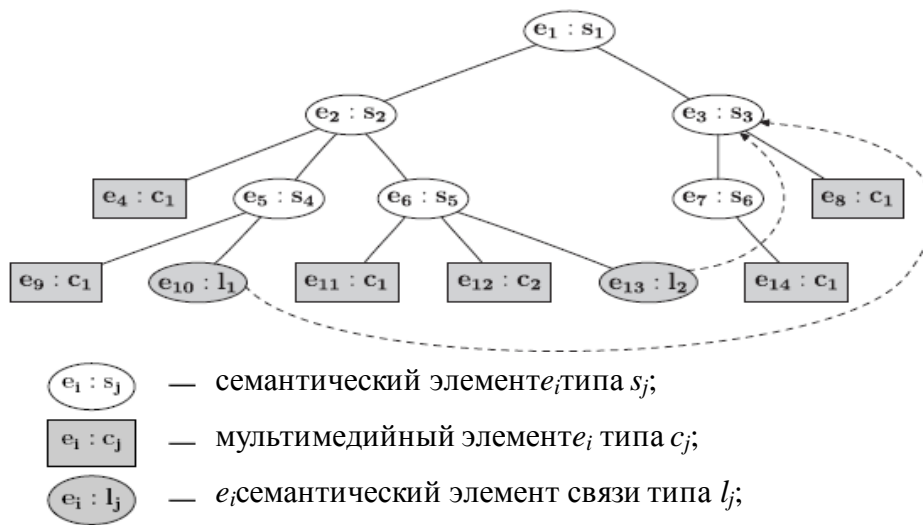


Рис. 2.13. Иерархическое упорядочивание контента.

4. Определение семантических связей, изображающих логическую связь между структурными элементами. Семантическая связь определяется как отображение множества элементов семантической связи в множество семантических элементов $H : \{e : G(e) \in L\} \rightarrow \{e : G(e) \in S\}$.

При разработке электронных учебных ресурсов в качестве предметной области выбирается целевая аудитория и элементы, в которых используются подобные множества факторов и ограничения, выявленные в возможных между ними отношениях. Для контроля правильности структуры контента будет необходим механизм оформления подобных ограничений. Метод, предлагаемый в этой исследовательской работе, требует принимать во внимание следующие ограничения.

1. Определение множества видов структурных элементов.

$S = \{s_1, s_2, \dots, s_n\}$ – множество используемых семантических элементов;

$L = \{l_1, l_2, \dots, l_n\}$ – множество используемых элементов семантических связей;

$C = \{c_1, c_2, \dots, c_n\}$ – множество используемых мультимедийных элементов.

2. Спецификация контентной иерархической структуры в виде $s \rightarrow exp$, где exp – шаблон, определяющий виды производных элементов для s и состоящий из следующих выражений:

- t – требуемый элемент типа t ;
- $t?$ – не требуемый элемент типа t ;
- t^* – произвольное количество элементов типа t ;
- t_1/t_2 – элемент типа t_1 или t_2 ;
- t_1, t_2 – элемент типа t_1 , стоящий после элемента типа t_2 ;
- (...) – группа элементов.

3. Для каждого вида связи, для которого организуется обращение и состоящего из выражений $l_i \rightarrow \{s_{j1}, \dots, s_{jk}\}$, выявляется спецификация семантических связей, определяющая множество видов элементов для этих видов связи.

В настоящей работе предлагается ряд численных характеристик, предназначенных для оценки семантической структуры контента. Основным требованием для предложенного контента считается адекватное изображение семантики дидактических материалов и постановка широких возможностей для обработки контента. Предложенные характеристики предназначены для оценки данных требований структуры контента. Исходя из семантической модели, использование механизмов характеристики структурирования подразделяются на три вида: *разделение по видам, семантическая связь, иерархическая структура*.

Основной характеристикой структуры контента является количество структурных элементов. По правилам, насколько много структурных элементов, используемых для изображения контента, настолько будут высоки возможности их использования в программе.

Следующие характеристики типизации дают возможность разделить структурные элементы по видам и классам:

$|TP_{t_i}(E)|$ – количество элементов в виде t_i ;

$|\bigcup_{t_i \in S} TP_{t_i}(E)|$ – количество семантических элементов;

$|\bigcup_{t_i \in C} TP_{t_i}(E)|$ – количество мультимедийных элементов;

$|\bigcup_{t_i \in L} TP_{t_i}(E)|$ – количество элементов семантических связей.

Для исследования структуры семантических связей определим отношение *указатель* во множестве семантических элементов. Если семантический элемент s связан (обращается) с производным элементом s' с помощью указателя, это считается прямой непосредственной связью. Однако в некоторых случаях указатель требует принимать во внимание и связь через третий элемент (посредник) или «отцовский» элемент (рис. 2.14).

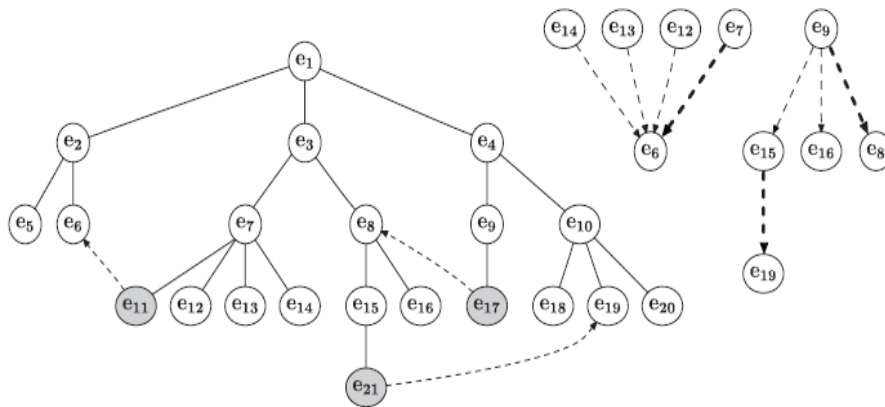


Рис. 2.14. Структура контента и ситуации для отношения *указатель*

На рис. 2.14 жирными линиями выделены прямые указатели. В некоторых случаях отношение *указатель* не вбирает в себя все семантические связи между элементами, в частности, не принимаются во внимание связи между производными элементами. Например: если параграф одной лекции предоставляется другому параграфу естественным образом можно считать эти

лекции связанными. Для анализа подобных связей введем семантически связанное отношение. Если выполняется одно из следующих условий, будем считать элементы семантически связанными: элементы совпадают; элементы или их наследники связаны отношением указатель; элементы связаны рекурсивно. В связи со связанностью каждого элемента со своими наследниками, предлагается рассматривать данное отношение на множествах, не состоящих из пары таких элементов.

Для оценки иерархической структуры контента необходимо обратить его влияние на возможность обработки данных, в первую очередь, на поиск информации. Для этого рекомендуется с помощью операций, посредственно определяемых иерархической структурой, оценить выделение большого количества различных частичных множеств множества структурных элементов.

Основными требованиями, предъявляемыми к структуре контента, считаются: адекватное изображение семантики дидактических материалов и создание широких возможностей для обработки контента. Предлагаемые характеристики служат для оценки соответствия структуры контента данным требованиям. Типизация и характеристика семантических связей разрешает специалистам по предмету и персоналу, занимающемуся подготовкой к изданию контента, исследовать его семантическую структуру и оценить его соответствие с помощью семантики дидактических материалов и закономерностей предметной области. Характеристики иерархических структур предназначены для оценки закономерностей контента по созданию программ и считаются не имеющими отношения к его какой-либо структуризации.

В рамках модели непосредственно выделяются структурные элементы, совпадающие с дидактическими единицами учебной информации, на основе разработанной технологии можно будет отделить семантическую структуру контента от механизма его визуализации и организации диалога с пользователем. Предложенные характеристики контента разрешают создавать алгоритмы и программы количественной оценки его параметров.

Основной особенностью предлагаемых технологий является то, что здесь требуется учитывать семантику контента, позволяющего еще более расширить функциональные возможности создаваемых в процессе его работы электронных учебных ресурсов.

В работе для создания электронных учебных ресурсов создан инструментальный программный комплекс, структура которого приведена на рис. 2.15 (на схеме стандартное программное обеспечение обозначено серым цветом). Для сохранения контента используется язык расширения XML, а для его редактирования – стандартный XML редактор.

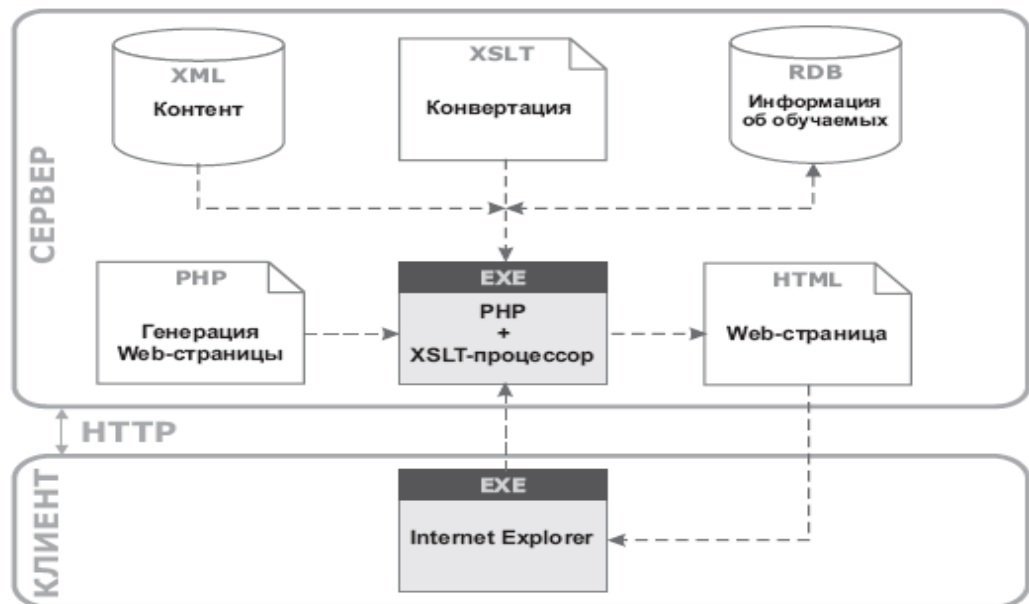


Рис. 2.15.

Контент может размещаться в различных специализированных форматах, для его визуализации и организации используются соответствующие браузеры. Позволяют создавать различные версии ЭУР. Инструментальный программный комплекс для распространения контента поддерживает сетевые версии, используемые в Интернете, создание локальных версий распространяемых на компакт дисках. А вследствие этого при учете семантики контента при

конвертации создается возможность использования одного из технологий распространения.

2.4. Разработка многоступенчатой математической модели учебного процесса

При виртуализации учебного процесса важное значение имеет разработка его математической модели. В то время как основной частью учебного процесса является обучение ученика, его математическая модель является основной составной частью модели учебного процесса. Основываясь на этом утверждении, рассмотрим основные этапы учебного процесса и процесс создания математической модели его управления. Сделаем это в логической последовательности:

1. Формализацию цели учебного процесса на основе предъявляемых к нему требований можно определить следующим образом:

$$Z^*: \begin{cases} \varphi_i(S) \geq a_i \quad (i = 1, \dots, k_1), \\ \psi_j(S) = b_j \quad (j = 1, \dots, k_2), \\ \eta_l(S) \rightarrow \text{extr} \quad (l = 1, \dots, k_3). \end{cases} \quad (2.4.1)$$

Здесь φ_i , ψ_j и η_l - состояния объекта S обучения и его функциональные критерии, определенные в среде $S = \langle X, Y \rangle$.

Также в (2.4.1) неравенство выражает комплекс минимальных знаний ученика при усвоении темы. Например, функция φ_i выражает критерий оценки знаний ученика по i -му предмету, в частности, в 5 балльной системе $a_i = 3$.

Равенство связано с базовыми знаниями и навыками ученика. Например, если ψ_j выражает степень знания какого-либо j -го закона природы, если он знает этот закон, то $\psi_j = 1$, а если не знает, то $\psi_j = 0$. Точно также $b_j = 1$.

Экстремальные отношения в целевой функции связаны с логическими целями учебного процесса, то есть эти отношения определяют качество учебного процесса. Например, если η_l - это средний балл по предмету l , тогда выполнение $\eta_l \rightarrow \max$ является естественным требованием. Если η_l рассматривать как время, необходимое для обучения предмета l , тогда обязательно следует добиваться чтобы $\eta_l \rightarrow \min$.

2. Этап отделения учебного процесса от среды. На этом этапе определяется граница знаний, необходимых для учебного процесса, и они формализуются.

Цель обучения в рамках знаний определяется с помощью Z^* в рамках выделенных учебных ресурсов R . Конечно такое положение связано с формой учебного процесса и выражает связь учебного процесса с внешней средой.

3. Этап создания структуры модели ученика. Создание такой модели очень важно для обеспечения эффективности учебного процесса и его управления. Рассмотрим структуру модели в виде оператора и определим его следующий обобщенный вид:

$$Y = F(X, U) \quad (2.4.2)$$

Здесь $X = (x_1, x_2, \dots, x_n)$ - начальное состояние ученика (или коллектива учеников), $U = (u_1, u_2, \dots, u_q)$ - ситуации, влияющие на учебный процесс, то есть комплекс знаний и информации, которых планируется довести до ученика, $Y = (y_1, y_2, \dots, y_m)$ - состояние ученика (коллектива учеников) во время учебного процесса.

Учитывая вышеизложенное, структуру модели оператора F в (2.4.2) и его параметры определим следующим образом:

$$F = \langle W, C \rangle \quad (2.4.3)$$

Здесь W - структура модели; $C = (c_1, c_2, \dots, c_k)$ - параметры модели.

Структура модели учебного процесса будет иметь вид:

$$Y = f_i(X, U, C) \quad (2.4.4)$$

Здесь f_i, C - операторы, выбранные с точностью параметров.

Необходимо в учебном процессе различать Y и Y' в выражениях (2.4.2), (2.4.4). Здесь Y' - количество ответов ученика на вопросы тестового контроля.

Тогда влияние на учебный процесс U можно понимать двояко. То есть,

$$U = (U_1, U_2) \quad (2.4.5)$$

U_1 - часть данных (знаний), предъявляемых для усвоения ученику. Когда ученик его усваивает, то учебный процесс оценивается оценкой Y .

U_2 - показатель усваиваемости ученика, то есть количество правильных ответов на заданные вопросы (или тесты). В результате этого учебный процесс оценивается оценкой Y' .

Значит, на рассматриваемом этапе в качестве структуры модели можно взять следующие:

$$Y = f_1(U_1, X) \text{ и } Y' = f_2(Y, U_2, X') \quad (2.4.6)$$

Здесь $X' = f_3(X, U_3)$, отсюда f_3 - количество правильных ответов ученика на вопросы тестового контроля на основе знаний ученика U_3 .

4. Синтез модели на уровне параметров. На этом этапе используем ведомость наблюдения за учебным процессом:

$$\langle X'_i, Y'_i \rangle, \quad (i = 1, 2, \dots, N) \quad (2.4.7)$$

При составлении модели на уровне параметров используем метод планирования опытов []. Для этого определяется поле опытов Ω , и вводится критерий эффективности учебного процесса K . Тогда при оценке состояния учебного процесса придется решить следующую задачу оптимизации.

$$K(U, W) \xrightarrow{U \in \Omega} \text{extr} \Rightarrow U^*. \quad (2.4.8)$$

Здесь W - структура модели; U^* - оптимальный план (состояние эффективности учебного процесса).

5. Синтез модели учебного процесса. На этом этапе в случае правильности структуры модели решается задача определения данных (знаний) U^* , необходимых для достижения цели Z^* .

Для этого цель учебного процесса, определенного в (2.4.1), формализуем следующим образом:

$$Z^*: \begin{cases} \varphi_i(X, Y) \geq a_i, & (i = \overline{1, k_1}), \\ \psi_j(X, Y) = b_j, & (j = \overline{1, k_2}), \\ \eta_l(X, Y) \rightarrow extr, & (l = \overline{1, k_3}). \end{cases} \quad (2.4.9)$$

Теперь поставим в (2.4.9) модель F , определенную в (2.4.2), и получим следующую задачу многоэкстремальной оптимизации

$$\eta_i(X', F(X', U)) \xrightarrow{U \in \theta} extr, \quad (i = \overline{1, k_3}). \quad (2.4.10)$$

Здесь θ - множество данных (знаний), определенных в учебной программе. Множество θ определяется следующим образом:

$$\theta: \begin{cases} \varphi_i(X', F(X', U)) \geq a_i, & (i = \overline{1, k_1}), \\ \psi_j(X', F(X', U)) = b_j, & (j = \overline{1, k_2}), \\ U \in R \end{cases} \quad (2.4.11)$$

R - ресурсы, выделенные для учебного процесса (учебная программа, база данных и так далее).

При решении задачи (2.4.9) – (2.4.10) можно воспользоваться следующим выражением, оценивающим комплекс экстремальных целей.

$$Q(X', U) = \sum_{i=1}^{k_3} \beta_i \eta_i(X', F(X', U)) \quad (2.4.12)$$

Здесь $\beta_i \geq 0$ - считается долей i -ой цели. В результате получим следующую вариационную задачу:

$$Q(X', U) \xrightarrow{U \in \theta} \min \Rightarrow U^* \quad (2.4.13)$$

Если критерии цели в (2.4.9) и (2.4.11) состоят из функций, тогда задача (2.4.13) приводится к обычной задаче математического программирования [1], и в таком случае имеются методы для его решения [2].

6. Этап внедрения учебного процесса. На данном этапе для внедрения учебного процесса на уровне структуры модели и модели на уровне параметров предлагается использовать следующую задачу оптимизации:

$$\|U' - U^*\| \rightarrow \min \quad (2.4.14)$$

U' – степень знаний ученика на основе усвоения данных знаний, U^* – степень знаний ученика на основе усвоения данных. При этом $\|..\|$ – выбранная норма, например, среднее квадратичное отклонение.

7. Адаптация модели учебного процесса. Как известно, существует ряд факторов, влияющих на учебный процесс. В таких случаях требуется адаптировать модель, созданную в вышеупомянутых этапах. Адаптация модели во многих случаях выполняется на уровне его параметров, и она реализуется посредством решения следующей задачи.

$$\|f_1(X, U, C) - f(X', U')\| \xrightarrow{C \in D} \min \Rightarrow C^* \quad (2.4.15)$$

Здесь C^* – параметры адаптированной модели; D – область изменения параметров.

В результате реализации вышеизложенных этапов создается математическая модель учебного процесса. А это, создавая возможность формализации учебного процесса, также дает возможность вести его к поставленной цели. На каждом из изложенных этапов используется структура модели учебного процесса или практика адаптации на уровне параметров (рис. 2.16).

В учебном процессе большие объемы информации находятся в движении. Кроме того, на каждом из вышеизложенных этапов приходится решать сложные идентификационные и оптимизационные задачи. Для этого рекомендуется использовать известные методы и соответствующее программное обеспечение, которое служит для внедрения его в практику, в частности, технологии Web программирования для централизованного управления учебным процессом.



Рис. 2.16. Этапы виртуализации учебного процесса.

Учебная модель, определенная выше, имеет важное значение при оптимальной организации учебного процесса. Для уточнения этого введем понятие модели ученика. Как известно состояние ученика выражается с помощью его оператора следующим образом:

$$Y = F^0(X, U). \quad (2.4.16)$$

С помощью этого оператора учебный процесс выражается таким образом:

$$\mathcal{F} = F(X', U). \quad (2.4.17)$$

Использование модели ученика определяется посредством степени его соответствия к процессу обучения. На практике существует несколько видов оператора F . Из (2.4.17) и (2.4.16) определим алгоритм обучения ученика:

$$U = \varphi(X', Y, Z^*, R) \quad (2.4.18)$$

Здесь φ - функция алгоритма обучения; Y – оценка состояния обучения ученика на основе модели F ; Z^* - цель обучения; R - ресурсы обучения.

Кроме того, алгоритм обучения включает в себя V -тесты, оценивающие состояние ученика, и эти тесты определяются следующим образом:

$$V = \Psi(X', Y) \quad (2.4.19)$$

Здесь Ψ - алгоритм создания теста V .

База данных или знаний для обучения I состоит из данных и знаний, которые должны быть усвоены учеником во время учебного процесса.

На этом основании можно следующим образом выразить комплекс данных и знаний, предлагаемых ученику при обучении на определенном этапе:

$$U' = \Psi_1(U, I) \quad (2.4.20)$$

Здесь Ψ_1 - алгоритм создания преподаваемых данных и знаний. Тесты, используемые при оценке результатов усвоения, можно определить следующим образом через операторную модель:

$$V' = \Psi_2(V, I) \quad (2.4.21)$$

В таких случаях ученик в системе обучения играет роль “отображателя” и степень его обученности V' определяется с помощью тестов. А модель этого процесса можно выразить таким образом:

$$Y' = D_Y(Y, V') \quad (2.4.22)$$

Здесь D_Y - V' оператор отображения тестов. В частном случае, то, что

$$U = V \quad (2.4.23)$$

намного упрощает систему обучения.

На основании вышеизложенных заключений в качестве самых важных параметров в системе обучения примем модель ученика F и алгоритм обучения φ .

На основании вышеопределенных закономерностей и обозначений определим следующим образом информационную структуру организации учебного процесса на базе модели ученика.

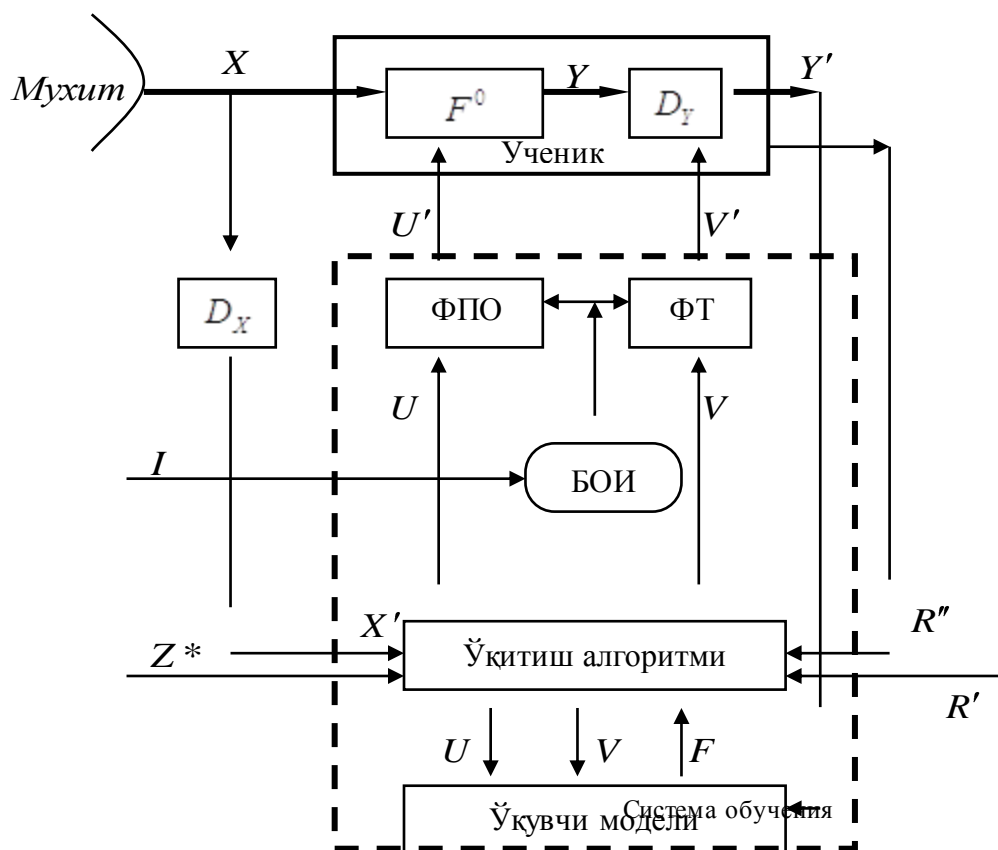


Рис. 2.17. Информационная структура организации учебного процесса на основе модели ученика.

В связи с тем, что поток информации, проходящей через среду, является очень важной для ученика, модель ученика F и способность восприятия и отображения ученика D рассматривается совместно в качестве объекта. Основным фигурантом в его составе является модель ученика и алгоритм обучения. Направления движения информационных потоков во время учебного процесса обозначены соответствующими стрелками. Функция цели обучения непосредственно входит в состав алгоритма обучения.

Формально видно, что оценить закономерность изменения состояния ученика в результате влияния системы обучения через его модель очень удобно. А это, в свою очередь, очень удобно при виртуализации учебного процесса и создает возможность для постоянной правильной оценки всех введенных показателей.

2.5. Программное обеспечение размещения электронных учебных ресурсов в образовательных порталах

В настоящее время использование информационно-коммуникационных технологий в УП ведет к повышению качества и эффективности системы образования. Этот процесс осуществляется в два этапа:

- учебные ресурсы, в частности, создание компьютерных моделей сценариев уроков и электронной базы учебных ресурсов;

- внедрение созданных электронных учебных ресурсов в реальный УП на основе мультимедиа и Web-технологий.

На обоих этапах требуется формализация учебных ресурсов объекта, сценариев уроков, то есть разработка их компьютерных моделей. Регенерация этих моделей (создание новых компьютерных форм) дает возможность эффективного использования мультимедиа и Web-технологий. При этом основной задачей является регенерация электронных учебных ресурсов и определение программного обеспечения, которое будет основным критерием его использования в корпоративной сети (через систему клиент-сервер).

Функциональная схема основного интерфейса программы, регенерирующей ресурсы разрабатываемого электронного учебно-методического комплекса, приведена на рис. 2.18.

Программное обеспечение выполнено на базе среды Delphi 7 и его модули имеют следующее содержание [7]:

- 1) В качестве **основного модуля** программного обеспечения приводится описание и внедрение его основной рабочей формы (интерфейс диалога между пользователем и педагогом, создающим электронный учебный ресурс). В этот программный модуль загружаются все задачи, начиная с этапа формирования структуры электронного учебного ресурса до этапа регенерации результата в виде

файла специального формата (текст программы приведен в приложении Unit Main).

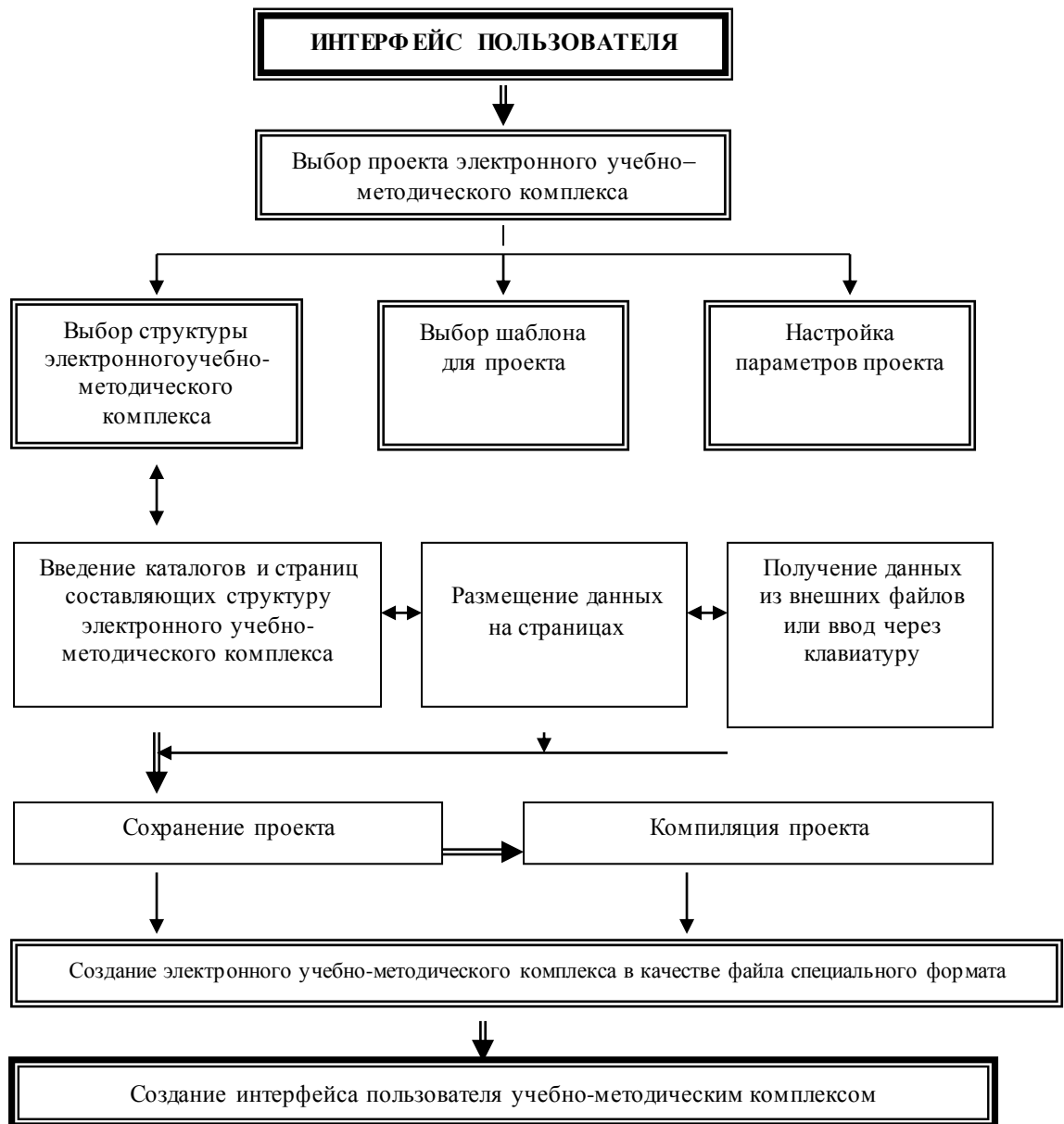


Рис. 2.18. Функциональная схема программы, формирующей электронную учебно-методическую среду

2) **Модуль** построения проекта электронного учебного ресурса и настройки параметров проекта. Этот модуль используется для введения имени, автора

проекта (электронного учебного ресурса), его внешнего вида (шаблона), таких параметров как место сохранения файла полученных результатов:

```
unit ProjectParamsUnit;
```

```
interface
```

```
uses { блок списка основных компонентов }
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Mask, ToolEdit, StdCtrls, JPEG, Buttons, Main, sComboBox, sButton, sEdit,
sLabel, sMaskEdit, sCustomComboEdit, sComboEdit, sDialogs, sSpeedButton,
sTooledit;
```

```
type
```

```
{ блок типов }
```

```
procedure
```

```
{ блок основных процедур }
```

```
end.
```

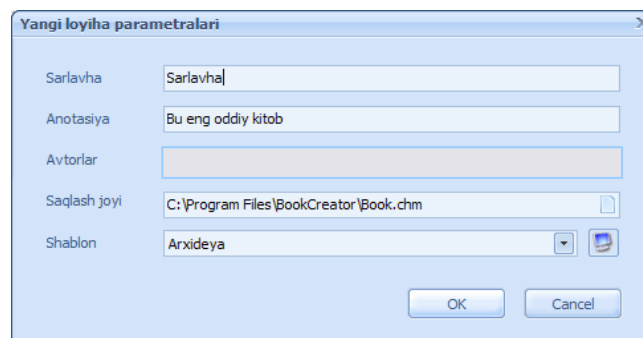


Рис. 2.19.

3) Модуль формы, показывающей процесс переноса электронного учебного ресурса, созданного с помощью программного средства, в файл специального формата.

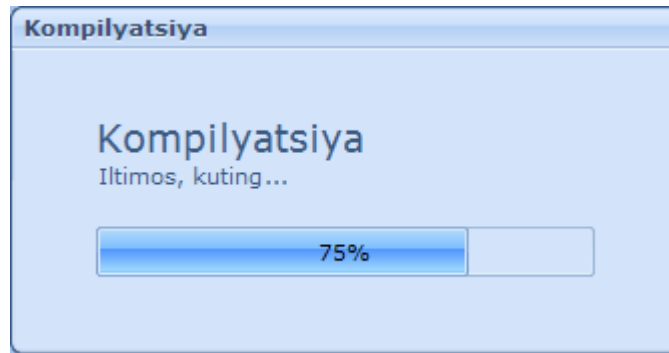


Рис. 2.20.

ПОЛНЫЙ ТЕКСТ ЭТОГО МОДУЛЯ:

```

unit Processing;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Animate, GIFCtrl, ExtCtrls, sGauge;

type

  TProcessingForm = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    sGauge1: TsGauge;
  private
    { Private declarations }
  public
  end;
var
  ProcessingForm: TProcessingForm;
implementation
  {$R *.dfm}
end.

```

4) **Модуль выбора шаблона** – внешний вид для пользователя электронного учебного ресурса. Этот модуль используется для процесса выбора шаблонов, введенных заранее в программное средство.

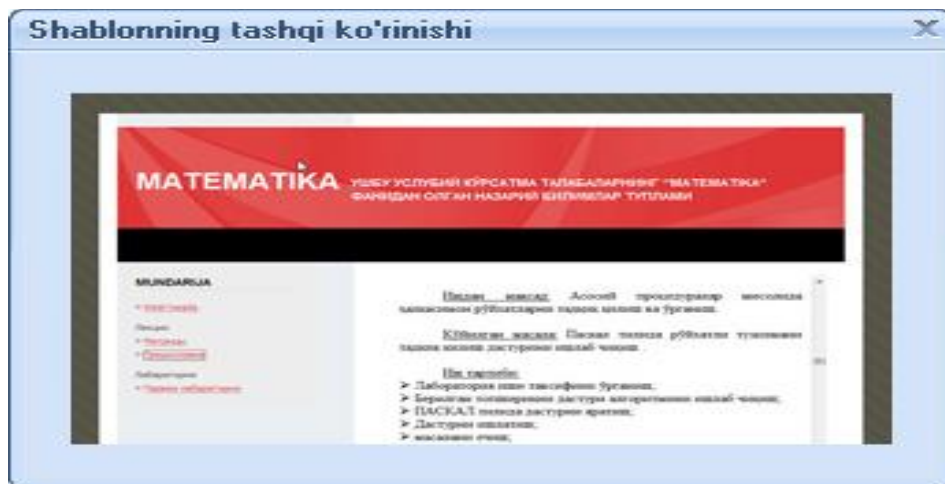


Рис. 2.21.

```

unit TemplatePreviewForm;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TPrevDialog = class(TForm)
    Image1: TImage;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  PrevDialog: TPrevDialog;
implementation

```

```
{ $R *.dfm }
```

```
end.
```

Для вышеуказанного программного средства с помощью основных программных модулей, используя программу, формирующую электронные учебные ресурсы, работающие на основе функциональной схемы, приведенной на рис. 2.18, создается следующий электронный учебно-методический комплекс (рис. 2.22).

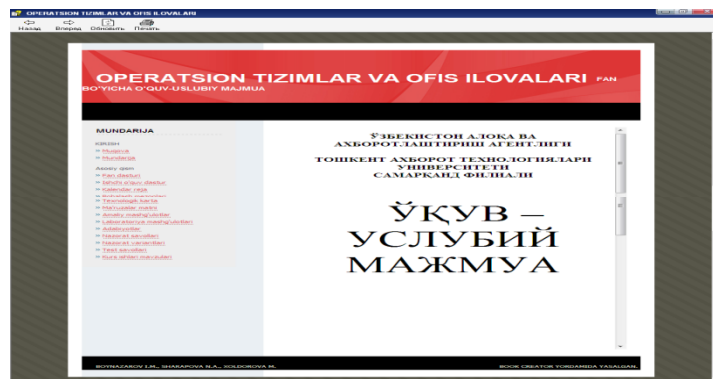


Рис. 2.22. Общий вид созданного электронного учебно-методического комплекса.

Проект созданного электронного учебно-методического комплекса сохраняется в файле специального формата *.XML, и в необходимых случаях для того, чтобы в электронный учебно-методический комплекс внести изменения и дополнения, данный файл заново загружается и редактируется.

Предлагаемое программное средство создано в среде Delphi 7.0 [7] и его целью является адаптация к использованию в виртуальном виде в процессе образования учебно-методического комплекса, подготовленного (запланированного) профессорами-преподавателями учебных заведений (в том числе высших). При этом важное значение имеет перевод учебных материалов в программное средство со специальным интерфейсом (перевод в файл формата *.chm на основе заранее выбранного шаблона) и адаптация к использованию электронного учебно-методического комплекса на основе сетевой технологии [4].

Программа призвана осуществлять следующие задачи:

- ввод электронных учебно-методических комплексов на основе специальной структуры;
- ввод учебных материалов и сохранение в специальном файле;
- редактирование введенных данных;
- размещение на страницах электронного учебно-методического комплекса таблиц, диаграмм, изображений, математических формул и других объектов и их редактирование;
- получение данных из файлов, созданных в программных приложениях Windows (*.doc, *.rtf, *.txt, *.ppt, *.xsl и так далее) и их редактирование;
- размещение на страницах данных из внешних файлов и их редактирование.

Общий интерфейс программы имеет следующий экранный вид (рис. 2.23):

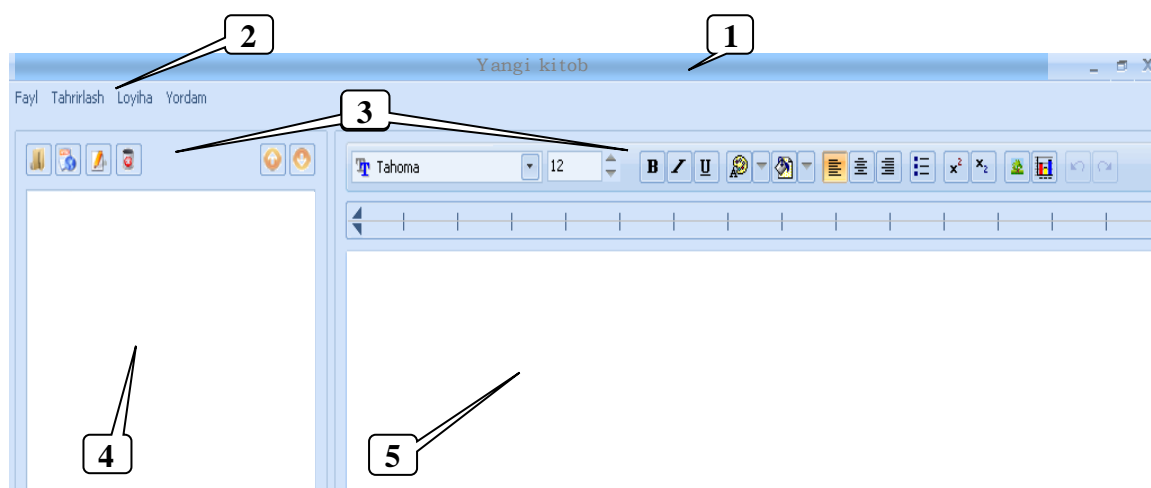


Рис 2.23. Основной интерфейс программы.

На рис. 2.23: 1 - строка заголовка, 2 - строка меню, 3 - панель инструментов, 4 - рабочая область для книжной структуры, 5 - рабочая область для размещения данных на страницах.

Строка меню программы состоит из таких пунктов как Файл, Редактирование, Проект, Помощь (рис. 2.24).

Пользователь, используя данные меню, может создать новый проект для электронного учебно-методического комплекса, для этого с помощью мышки

следует выбрать команду **Файл** → **Новый проект** или нажать клавишу на клавиатуре **Ctrl+N**.

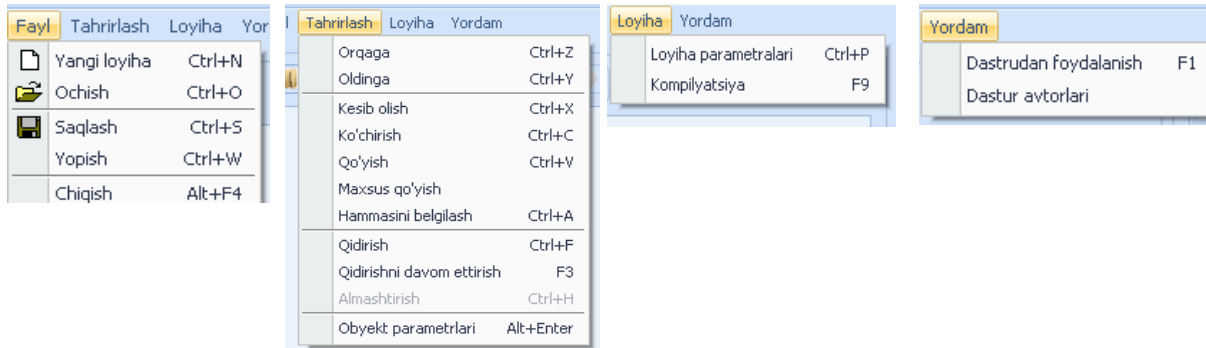


Рис. 2.24. Общий вид пунктов строки меню.

После этого, в поля специального окна программы по желанию пользователя вводятся параметры нового электронного учебно-методического комплекса (рис. 2.25).

Для подготавливаемого электронного учебно-методического комплекса вводятся заголовок, краткая аннотация, список авторов, место хранения электронного учебно-методического комплекса и соответствующее ему имя, а также выбирается шаблон для электронного учебно-методического комплекса.



Рис. 2.25. Диалоговое окно ввода параметров нового проекта.

В случае необходимости замены параметров проекта выбирается команда в строке меню **Проект**→**Параметры проекта** или нажимается клавиша **Ctrl+P**, и вносятся изменения в вышеуказанном диалоговом окне.

Панель инструментов программы состоит из двух частей (рис. 2.26), первая часть состоит из кнопок изменения каталога, страниц, параметров страницы, удаления страницы или каталога, места расположения страницы или каталога. Вторая часть состоит из кнопок редактирования данных, расположенных на страницах. При управлении этой панелью инструментов используется мышка. Во второй части можно изменять шрифт текста (имя, форму и размеры), цвет текста, цвет фона текста, место расположения текста на странице (выравнивание по горизонтали влево, по центру, вправо), вводить данные в виде списка, писать в нижнем и верхнем регистре, размещать графические объекты из внешних файлов, размещать на страницах объекты, возвращаться в предыдущее и последующее состояние.

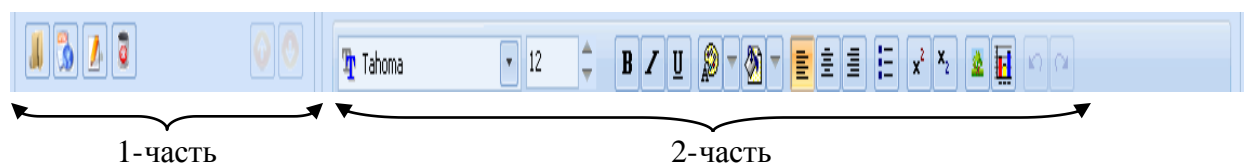


Рис. 2.26. Панель инструментов программы.

В процессе использования каждой кнопки на панели инструментов открываются специальные диалоговые окна и вводятся параметры, соответствующие подготавливаемому электронному учебно-методическому комплексу (Рис. 2.27, 2.28, 2.29). Например, для создания нового каталога используется диалоговое окно (Рис. 2.27а), здесь вводятся степень и имена каталогов в соответствии со структурой проекта. Диалоговое окно создания новой страницы (рис. 2.27б), здесь вводятся имя каталога, где расположена страница, выбирается его степень (основная страница или другое) и вводится его имя. Точно

также с помощью кнопки изменения параметров страницы можно отдельно изменить введенные параметры.

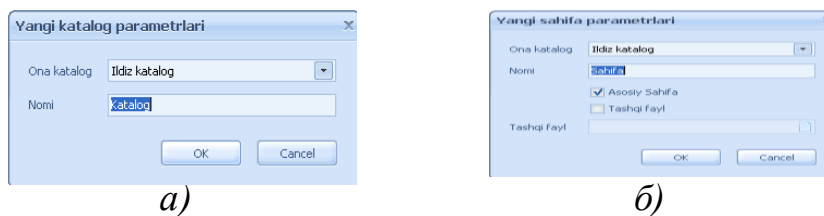


Рис. 2.27. Диалоговые поля ввода параметров нового каталога и страницы.

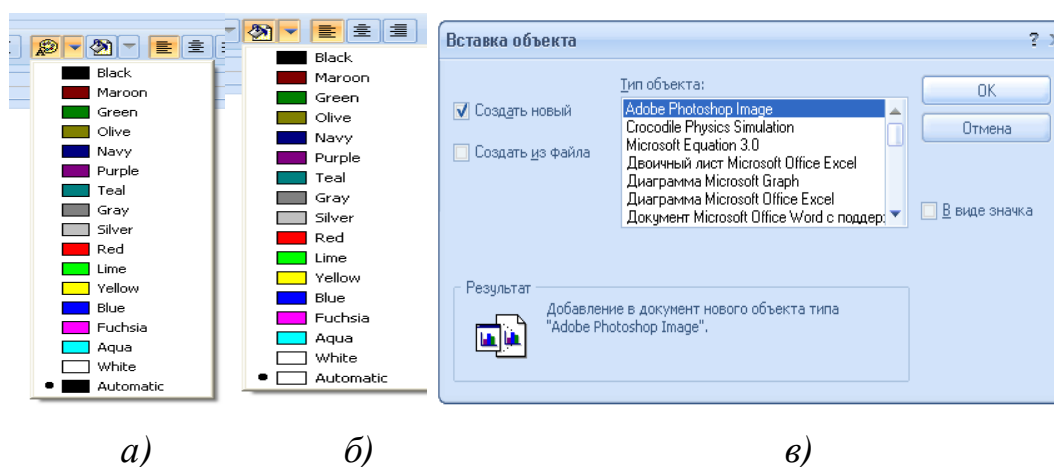


Рис. 2.28. Основные кнопки: 2-части панели инструментов: а - цвет текста, б - выбор фона страницы, в - размещение объектов на страницу.

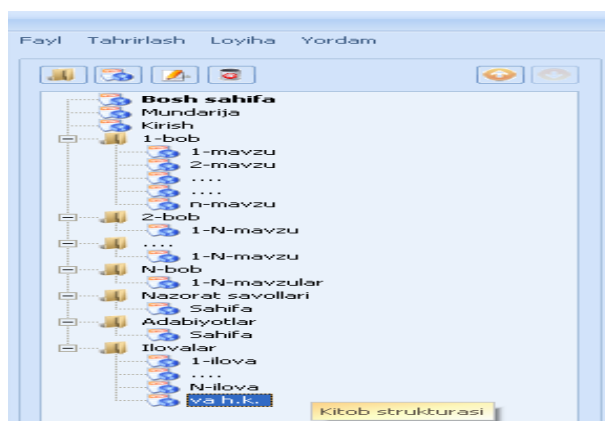


Рис. 2.29. Образцовая структура для разрабатываемого электронного учебно-методического комплекса (дерево страниц и каталогов).

Особенностью данной программы является то, что даже пользователи, не обладающие возможностью профессионального использования программным обеспечением компьютера, могут создавать виртуальные занятия. Если пользователь в процессе работы с программой столкнется с какой-либо неточностью, тогда с помощью пункта **Использование программы** в меню **Помощь** можно получить необходимую информацию.

Кроме того, в программе используются несколько дополнительных модулей и библиотек, не являющихся стандартными. Одной из них является известная большинству библиотека **RXLib**, состоящая из комплекса удобных компонентов. Эта библиотека состоит из набора компонентов совершенствованных вариантов стандартных компонентов среды **Delphi**, которые облегчают процесс программирования. В данной программе использована компонента **TRXRichEdit** библиотеки **RXLib**. Эта компонента похожа на стандартный компонент **Delphi TrichEdit** и вместе с тем, имеет ряд преимуществ и удобств. Размещение изображений различного формата в состав документа, создаваемого с помощью компонента **TRXRichEdit**, намного упрощается, а добавление в документ дополнительных объектов (диаграмм **Microsoft Word**, **Microsoft Excel**, математических формул и так далее) не вызывает у программиста никаких затруднений.

Еще одной из нестандартных библиотек, использованных в программе, является библиотека **Alpha Skins**, которая изменяет внешний вид программы. С помощью этой библиотеки программист может полностью изменить внешний вид программы. Для библиотеки **Alpha Skins** существует десяток различных файлов внешнего вида (скинов), которые построены со вкусом. Их выбор может осуществляться не только со стороны программиста, но и со стороны пользователя.

Выводы по II главе

1. На основе общего определения электронных учебных ресурсов разработаны дискретные математические модели виртуализации УП, и на их базе созданы соответствующие алгоритмы для реализации программного обеспечения.

2. Была разработана технология создания семантических моделей электронно-учебных ресурсов, и на их базе определены закономерности отношений между контентными электронно-учебного ресурса.

3. Разработана многоэтапная математическая модель УП, и были изучены свойства адекватности и устойчивости этих моделей. Была определена роль составленных моделей в виртуализации учебного процесса, и разработаны алгоритмы их внедрения.

4. Создано соответствующее программное обеспечение для размещения электронных учебных ресурсов в систему образования, основанного на Web-технологиях.

III ГЛАВА. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВИРТУАЛИЗАЦИИ УЧЕБНОГО ПРОЦЕССА НА ОСНОВЕ WEB-ТЕХНОЛОГИЙ И ЕГО ПРИЛОЖЕНИЯ

3.1. Пользовательский интерфейс программы виртуализации учебных процессов

При получении решения задачи виртуализации учебного процесса была разработана автоматизированная система, основанная на Web-технологиях. Пользователи этой системы разделены на две группы, в которой одной группой является обычный пользователь (студент, самостоятельно обучающиеся лица), а другой является системный администратор (администратор сервера, преподаватель) (Рис. 3.1).



Рис. 3.1. Интерфейс пользователя автоматизированной системы

При разработке системы, работающей на основе вышеуказанной функциональной схемы были разработаны требования к системе и были использованы следующие технологии с учетом его возможностей:

1. При создании части системы, проверяющей решение задачи, была выбрана среда программирования Borland Delphi. Как нам известно, язык Delphi считается развивающимся языком программирования. Основным из достоинств этого языка является то, что с помощью богатой библиотеки компонентов (VCL) в короткое время можно разработать программные средства.

2. При проектировании и разработке базы данных была выбрана база данных MySQL, потому что она имеет большую эффективность и можно добиться хороших результатов при его применении к крупным проектам.

3. Для связки программного средства, разработанного на Delphi, и базы данных MySQL вместо стандартной библиотеки Delphi dbExpress была использована быстрая и компактная библиотека Zeos-Lib. Потому что эта библиотека дает возможность связки и с отличными от MySQL базами данных (например, Oracle, MS SQL).

4. При разработке Web-интерфейса был выбран язык сценариев PHP. Следует особо подчеркнуть, что Web-интерфейс разработан на основе принципов программирования, ориентированных на полные объекты. Это создает большие удобства при расширении и совершенствовании системы. Вместе с тем в интерфейсе системы были полностью использованы элементы HTML, CSS и JavaScript. Это, в свою очередь, освобождает от установки пользователем дополнительных программных средств, то есть для пользования системой достаточно наличие только Web браузера.

5. В целях повышения удобства Web-интерфейса для пользователя и его интерактивности были применены MooTools Framework. Удобство фреймворка в том, что есть возможность использования технологии AJAX и пакет готовых инструментов для манипуляции с DOM.



Рис. 3.2. Web-интерфейс системы.

Разработанная система может работать в двух режимах:

- 1) режим использования в процессе практических и лабораторных занятий при преподавании предмета в локальной системе учебного заведения;
- 2) Режим, используемый для самостоятельной проверки знаний в режиме он-лайн сети Интернет.

В обоих режимах система работает на основе технологий клиент-сервер (Рис. 3.2).

3.2. Программное обеспечение проведения дистанционно-тестового контроля на основе Web- технологий

Одной из новых возможностей современных информационных технологий является организация дистанционного обучения. Под дистанционным обучением понимается обучение ученика на расстоянии (в разрезе учебных предметов) и дистанционный контроль усвоения каждого предмета. На сегодняшний день в развитых странах налажена организация вступительных экзаменов в высшие учебные заведения, проведение текущих и итоговых контролей по каждому этапу обучения на основе технологий дистанционного обучения. Основная цель этого в том, что ученикам на дистанции предоставляется комплекс знаний и навыков, а также их знания и навыки, приобретенные на всех этапах учебного процесса, оцениваются посредством виртуальных систем. С другой стороны, предусматривается самостоятельное обучение студентов - обучение изысканию. Дистанционное обучение, в частности, процесс осуществления на дистанции тестового контроля создает возможность уменьшения его себестоимости и ускорения процесса обучения наряду с обеспечением качества УП. Вместе с тем, система дистанционного обучения обеспечивает информационную культуру между отношениями “преподаватель” (виртуальный преподаватель) и “ученик” (виртуальный ученик). [5]. В связи с этим считаем тему, выдвинутую в данном параграфе, актуальной [1,2,3,5].

При применении информационных технологий в обучении использование систем контроля также становится основанием для достижения высокой эффективности, как и в традиционной системе обучения. При внедрении информационных технологий в процесс обучения использование системы контроля осуществляется полностью на централизованной основе. А это создает возможности и условия для охвата полного контингента обучающихся. Кроме того, в процессе контроля знаний использование компьютеров создает возможность достижения объективности при оценке уровня знаний

обучающегося, то есть оценки результатов контроля вне зависимости от субъекта преподавателя, осуществляющего этот контроль. В настоящее время автоматизированные системы тестирования, используемые для контроля, состоят из частей, выполняющих следующие задачи:

- составление тестов (формирование банка вопросов и задач, определение стратегии оценки и ввода опроса);
- проведение теста (предоставление вопросов и обработка ответов на них);
- мониторинг обученных тем, знаний по направлениям за определенный промежуток времени на базе динамически изменяющейся базы данных с протоколированием качества знаний.

Функциональная схема такой системы контролирующей знания состоящей из этих подсистем представлена на рис. 3.3.

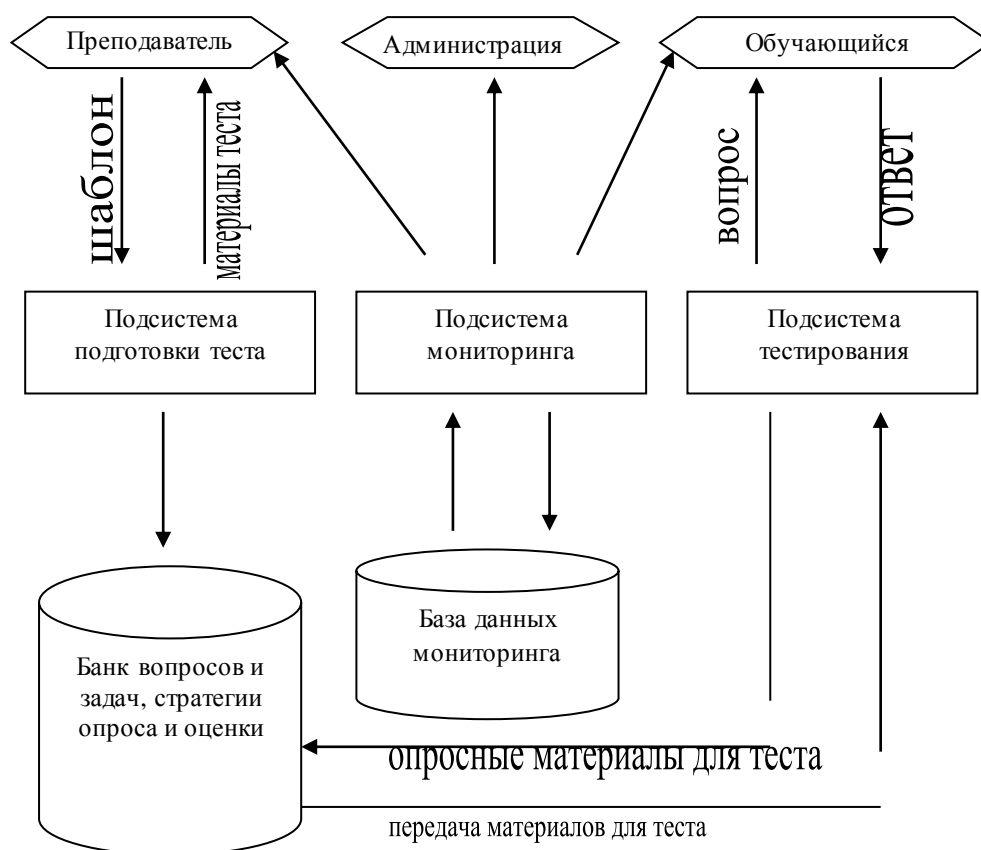


Рис. 3.3. Функциональная схема системы контроля.

В подсистеме *Подготовка теста* непосредственно работает оператор, вводящий данные, предоставленные педагогом (учителем). В целях избежания ошибок при вводе вопросов теста необходимо предоставить вопросы и задачи, варианты ответов и специальный шаблон для пользователя, куда он будет вводить правильный вариант ответа. Результат работы этой подсистемы формируется в качестве базы данных, служащей для тестирования. Студент, тестирующий свои знания, при использовании этой системы должен иметь возможность самостоятельно выбирать количество вопросов и алгоритм, предоставляющий эти вопросы. Также требуется сохранение результатов контроля, проведенного с помощью этой подсистемы, для обучающегося, администрации и учителя для проведения необходимого мониторинга знаний.

Наблюдения показывают, что современные контролирующие системы должны отвечать нескольким требованиям. В частности, система должна предоставлять на любом выделенном компьютере в локальной сети или Интернете сведения по степени сложности по направлениям предметов, разделяя их по темам и видам. Кроме того, при составлении очередного теста и введении его в систему необходимо обращать внимание на его содержание и степень сложности. Требуется, чтобы каждый нововведенный тест одинаково предоставлялся обучаемому из базы данных. А это создает удобство в формировании учителем тестов и подготовке обучаемого к тестовому контролю.

Дистанционное обучение, в частности, технологии проведения дистанционного тестового контроля состоят из электронного учебно-методического комплекса и электронных контролирующих систем, основанных на Web-технологиях. В такой системе разрабатывается дистанционный вид оценки знаний. В дистанционной тестовой контролирующей системе оценки знаний требуется отдельный учет функций зарегистрированных на рис. 3.4 [1,2].

Отдельно выделим свойства обозначенных функций:

- организация динамической связи (режим он-лайн) – здесь предполагается организация диалога в локальных или модемных сетях связи. Создается

возможность, чтобы пользователь (лицо, проходящее тестовый контроль) в любое время мог провести контроль своих знаний, навыков в предоставляемой тестовой системе контроля. Вместе с тем, при одновременном входе в систему или повторном прохождении контроля принимается во внимание тот факт, что вопросы тестового контроля предоставляются в постоянно меняющихся вариантах;

- защита от несанкционированного доступа [4] – посредством этой функции регистрируется каждый пользователь. Перед входением в систему контроля пользователь включается в список пользователей системы. (аутентификация). Этот процесс осуществляется администратором системы тестового контроля, пароль, выбранный пользователем, предоставляется в зашифрованном виде (при этом рекомендуется использовать алгоритм MD5);

- многовариантное предоставление тестовых вопросов – во время повторного входа в систему или для каждого пользователя предоставляется комплекс вопросов в виде отдельных вариантов;

- составление таблицы ответов по каждому тестовому контролю. В этой таблице регистрируется имя-отчество пользователя и его полученная оценка. После каждого контроля пользователю предоставляются его результаты. Если пользователь заново регистрируется, то принимается во внимание то, что его результаты удаляются администратором системы;

- организация нескольких вариантов правильных ответов по каждому тестовому контролю. Целесообразно, чтобы ответов на каждый тестовый вопрос было не больше шести. Во время контроля предоставляется, по крайней мере, три варианта ответов;

- предоставление вопросов, введенных в тестовый комплекс посредством случайной генерации. Вопросы из комплекса тестовых вопросов, составленных из определенного набора вопросов по предмету, предоставляются посредством случайной генерации.



Рис. 3.4. Основные функции системы дистанционного тестирования и контроля

Для предоставляемой системы тестового контроля база данных будет состоять из следующих таблиц [1,2,3,4,6]:

- таблица учета пользователей – foyd_reg;
- таблица учета тестовых вопросов – test_reg;
- таблица вопросов для тестов, которые будут зарегистрированы, – sav_jad;
- таблица ответов (результатов)– sav_jad_j;

Таблица учета пользователей foyd_reg состоит из полей для полного имени, логина и пароля. Эта таблица состоит из следующего SQL-запроса:

```

CREATETABLEfoyd_reg (
  loginchar(16) PRIMARYKEY,
  passwd char(128) NOT NULL,
  fullnamechar(32)
)
  
```

Для предоставления теста пользователям они должны быть зарегистрированы в системе, это осуществляется в таблице test_reg. Эта таблица состоит из следующих запросов:

```

CREATETABLEtest_reg (
  nointPRIMARYKEY,
  desc char(32),
  qmax int,
  tbl char(64),
  autor char(32)
)

```

В этом запросе NO – порядковый номер вопроса теста, DESC – текст, QMAX – максимальное количество вопросов (соответственно количество вопросов полученных случайным выбором), AUTHOR – поле со сведениями об авторе теста.

Тестовые вопросы хранятся в отдельной таблице. Имя этой таблицы указывается в вышеуказанном поле TBL. Имя таблицы ответов, исходя из имени таблицы вопросов, должно оформляться соответствующим образом. Например, если имя таблицы вопросов SAV_JAD, то таблица ответов должна оформляться как SAV_JAD_J [1,6].

Таблица тестовых вопросов имеет следующий вид:

```

CREATE TABLE sav_jad (
  no int PRIMARY KEY,
  quest TEXT,
  true char(6),
  dif char(1)
)

```

Здесь поле NO – порядковый номер тестового вопроса. Поле QUEST представляет текст вместе с вариантами ответов на тестовые вопросы, например:

С какого года начали свою деятельность INTERNET провайдеры Узбекистана?

1. с 1983 года
2. с 1991 года
3. с 1997 года

4. с 2000 года

В поле TRUE указывается номер варианта правильного ответа. Если правильных ответов несколько, тогда номера ответов пишутся последовательно без пробелов, например, если правильные ответы указаны в вариантах 2 и 3, то в поле вводится «23».

В поле DIF определяется степень сложности тестового вопроса. Если в этом поле стоит знак «Y», то он считается сложным вопросом, и ученику ставится высокий балл, определенный для ответов. Например, для правильного ответа на сложный вопрос ставится 3 балла, на обычные вопросы ставится только 1 балл.

Таблица ответов (результатов) организуется следующим образом:

```
CREATE TABLE sav_jad_j (
  login char (16)NOT NULL PRIMARY KEY,
  markint,
  trueint,
)
```

Здесь в первом поле стоит логин пользователя, во втором поле - его полученная оценка, а в третьем поле – количество правильных ответов.

Система дистанционного тестового контроля, состоящая из вышеуказанных запросов, работает на основе следующего алгоритма [2]:

1. Пользователь регистрируется. Если логин или пароль введены неправильно, учитывая, то это представляет собой несанкционированный вход, работу в системе для этого пользователя.

2. Выводится несколько вариантов тестов. Пользователь выбирает один из них и начинает отвечать. Если он правильно отвечает на все вопросы, то выводится сообщение об этом и ему предлагается войти в другой тест. Передача теста осуществляется только с ведома администратора, потому что надо будет необходимо удалить предыдущие ответы пользователя.

3. Генерация тестовых вопросов посредством случайного выбора.

4. Последовательное предоставление пользователю нескольких выбранных тестовых вопросов. Если пользователь ответит на них правильно, то проверяется степень сложности вопросов и составляется соответствующая сумма баллов.

5. После ответа пользователем на последний вопрос, происходит вывод результатов и ввод его результатов в таблицу результатов. Если пользователь добился результата 86-100%, то он получает «отлично», если 71-85% - «хорошо», если 55-71% - «удовлетворительно», если 0-55% - «неудовлетворительно» и все это объявляется.

Предоставим сценарий основных функций, примененных для системы дистанционного тестового контроля и предназначенных для работы на основе этого алгоритма [1,5,6]:

1. Функция соединения с сервером:

```
function connect()
{
mysql_connect("localhost", "root", "") or die(mysql_error());
mysql_select_db("st");
}
```

2. Функция вывода на экран текста вопросов:

```
function GetQuestText()
{
global $Counter, $text;
$tmp=$Counter+1;
echo "<br><b> Savol $tmp </b><br>";
echo "$text";
echo "<br>";
echo "<form action=$SCRIPT_NAME><br>";
echo "<input type=checkbox name=a1 value=1>1
<input type=checkbox name=a2 value=2>2";
echo "<input type=checkbox name=a3 value=3>3"
```

```

<input type==checkbox name=a4 value=4>4”;
echo “<input type=checkbox name=a5 value=5>5
<input type==checkbox name=a6 value=6>6”;
echo “<br><input type=submit value=\\”Введитеответы!\\”>”;
}

```

3. Функция вывода списка тестовых вариантов в виде таблицы:

```

function GetQuestList()
{
global $SCRIPT_NAME;
echo “<center><i>Список предлагаемых тестов:</i></center><br>”;
echo “<table border cols=3 width=\\”100%\\” bgcolor=#ceffce>”;
echo “<tr align=center bgcolor=#3333ff>”;
echo “<td><b><i><font color=#ffffff> Nomer</font></i></b></td>”;
echo “<td><b><i><font color=#ffffff> Nomi </font></i></b></td>”;
echo “<td><b><i><font color=#ffffff> Muallif </font></i></b></td>”;
$res=mysql_query(“select no, desc, author from test_reg”);
While ($Row=mysql_fetch_row($res))
{
echo “<tr>”;
for($i=0; $i<mysql_num_fields($res); $i++)
echo “<td><a href=$SCRIPT_NAME?tno=$Row[0]>$Row[$i]</a></td>”;
echo “<tr>”;
}
echo “</table>”;
}

```

В заключение можно сказать, что система тестового контроля, основанная на современных Web-технологиях, дает возможность объективной оценки знаний, навыков студента (пользователя). Кроме того, данная система служит для

приобретения студентом навыков для самостоятельной работы, которая является одним из актуальных требований сегодняшнего времени.

3.3. Программное обеспечение автоматизированной системы для тестирования результатов программы, созданной на основе Web- технологий

Наряду с совершенствованием вычислительных машин само собой совершенствуется и их программное обеспечение. Считаем, что в процессе подготовки в высших учебных заведениях технического профиля специалистов, имеющих высокую способность программирования, учитывая их интересы и талант, целесообразно будет проводить постоянный контроль их знаний по программированию. Поэтому при проверке (тестировании) правильности программного продукта, созданного программистом, возникают некоторые проблемы. Одной из таких проблем считается оперативное тестирование созданных программ, проблема экономии времени и средств, расходуемых на проверку правильности полученных результатов. Если учитывать это, протестировав результат созданной программы, считается необходимым разработка автоматизированной системы, которая может дать достаточное заключение о решении. Такие системы в настоящий момент применяются и при проведении мировых олимпиад по программированию, то есть до сегодняшнего времени создан десяток автоматизированных систем, тестирующих результаты программирования. В их число входят Z-training, Campion, Caribbean Online Judge, Fuzhou University Online Judge, Harbin University Online Judge, Infoarena, Lviv National University Online Judge, Moscow Problem Archive with Online Judge, Peking University Online Judge, Saratov State University Problem Archive with Online Judge, Sphere Online Judge, Tianjin University Online Judge, Timus Online Judge и так далее. Эти системы различаются между собой объемом архива задач в базе данных, количеством языков программирования, которые можно в них

использовать. Следует особо подчеркнуть, что вышеуказанными автоматизированными системами нельзя напрямую пользоваться в процессе обучения в учебных заведениях. Во-первых, не существует задач в архиве задач, соответствующих учебным программам. Во вторых, нет перевода имеющихся программ на узбекский язык. В третьих, (основное) для анализа результатов учителю не предоставляется право доступа в систему.

На сегодняшний день в Узбекистане не существует автоматизированной системы, проверяющей результаты программы [19].

Учитывая вышеизложенное, решение задачи создания варианта автоматизированной системы проверки результатов программ, предназначенного для использования в учебных заведениях, рассмотрено в данной диссертационной работе. Эта программная среда предназначена для использования на основе сетевых технологий. То есть алгоритм, дающий решение задачи, был запрограммирован на основе Web-технологий.

Преимущество таких систем в том, что за короткий промежуток времени данное решение тестируется, и программисту предоставляются сведения о правильности и ошибках этой программы. Кроме того, можно проверить время работы программы и с высокой степенью точности определить объем памяти.

Эта система наряду с вышеуказанными возможностями имеет такие недостатки как не полное исполнение обязанностей преподавателя или педагога.

Однако в варианте, создаваемом для использования в процессе обучения, учитель имеет возможность обращаться к системе в качестве администратора и анализировать результаты несколькими способами.

Разработка автоматизированной системы, тестирующей результаты программы, была спроектирована на основе следующих этапов.

1-этап. Проектирование и разработка базы данных. На этом этапе требуется принимать во внимание сохранение всех обработанных этой системой данных в специальной базе данных. В частности, задачи, предлагаемые для составления

программы, их решения и соответствующие им коды, время работы программы, объем памяти должны храниться в архивированном виде в базе данных.

В рабочем состоянии системы все данные о результатах проверки всех введенных в нее решений также должны быть сохранены в базе данных системы и предоставлены для анализа администратору.

2-этап. Разработка проверяющей части. На этом этапе предусматривается проверка правильности написания программного кода решения, предлагаемого для существующих задач, времени его работы и создание системы вычисляющей объем памяти. Принимая во внимание предложение пользователем решения задачи на нескольких языках программирования, разрабатывается вариант проверяющей системы, соответствующий данному языку программирования.

3-этап. Разработка базы задач. На этом этапе предусматривается создание базы задач, касающейся программирования для практических и лабораторных работ на уроках программирования и база тестовых вариантов для проведения теста правильности решения соответствующей задачи. Вместе с тем и правильные коды решений для каждой задачи на предлагаемых языках программирования должны быть внесены в базу в качестве отдельного файла.

4-этап. Разработка Web-интерфейса для создания возможности использования системой на основе сетевых технологий. На этом этапе предусматривается разработка специального Web-интерфейса, управляющего базой данных, созданной на предыдущих этапах.

Общая схема автоматизированной системы, тестирующей результаты программы, разработанной в результате этих этапов, приведена на рис. 3.5.

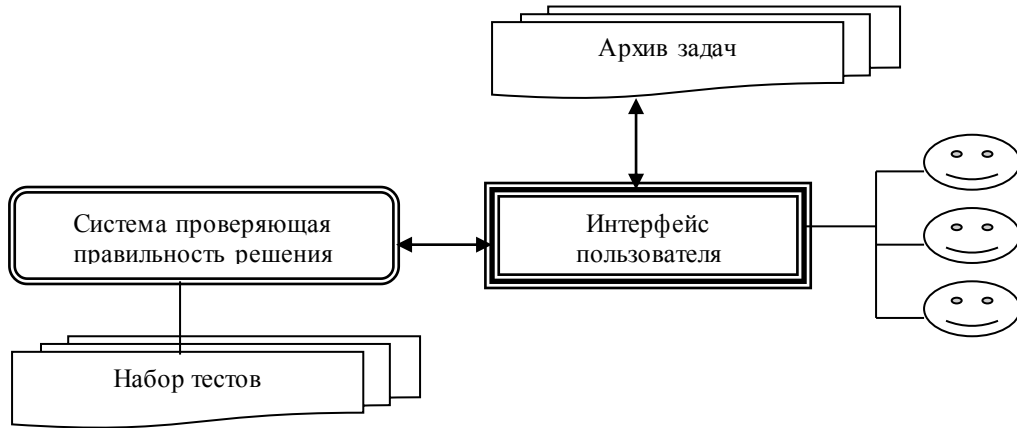


Рис. 3.5. Общая схема автоматизированной системы проверки правильности программы

Основным элементом этой системы является блок, проверяющий правильность результатов программы (рис. 3.6). Эта подсистема состоит из решений, предложенных пользователю задач, и базы чеккерских программ, проверяющих его с базой тестов. Решение каждой задачи дополнено базой заранее определенных тестов.

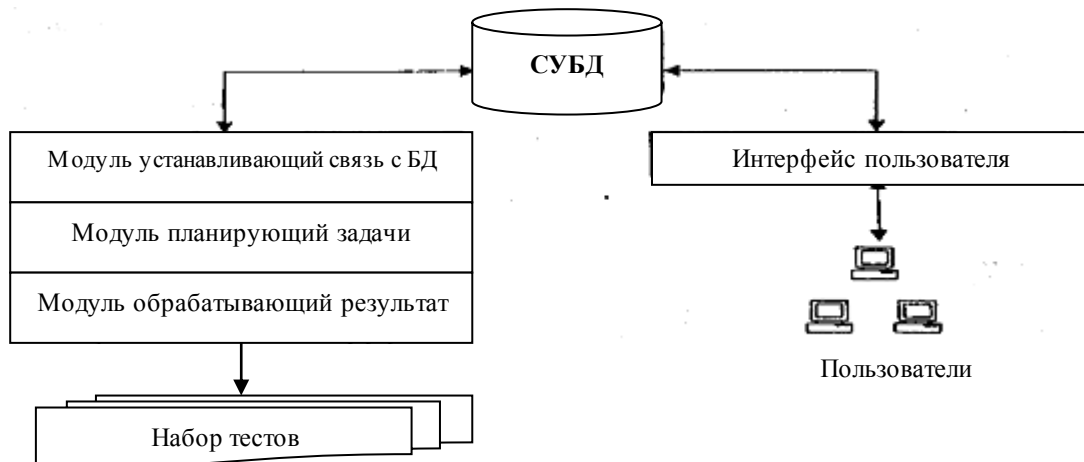


Рис. 3.6. Общая схема блока тестирования результатов программы

При разработке системы, автоматически проверяющей результаты программы, работающей на основе вышеуказанных схем, были использованы РНР-технология, среда создания базы данных MySQL и среда программирования

Delphi 7.0. Интерфейс диалога пользователя с администратором осуществляется на основе Web-технологий.

Интерфейсы диалога изображены на следующих рисунках.

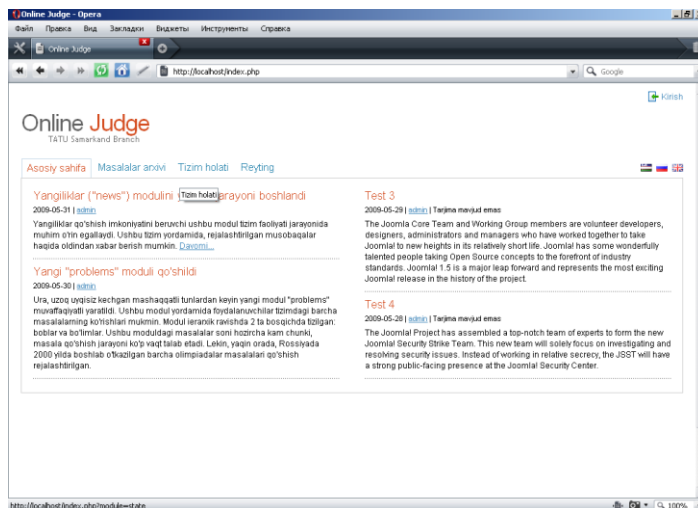


Рис. 3.7. Главная страница системы (основная страница).

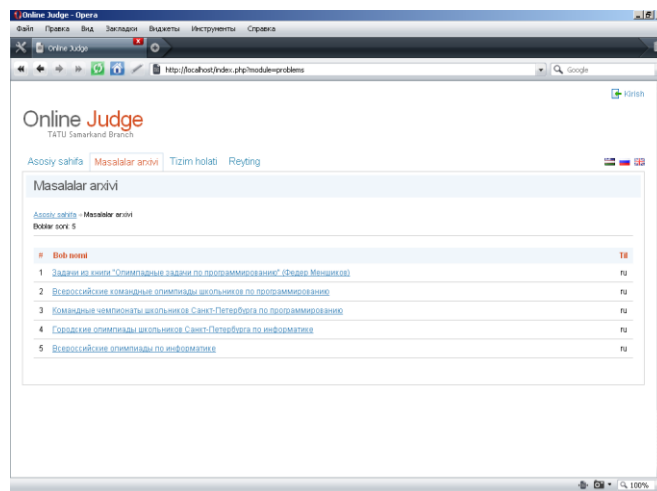


Рис. 3.8. Страница просмотра базы задач

Для входа пользователем в систему нажимается кнопка ВХОД, расположенная в правом верхнем углу страницы, в поле входа (рис. 3.9) и требуется ввести заранее введенный в список адрес e-mail адрес и пароль.

Kirish formasi

E-mail

Parol

[Registrasiya](#)
[Parolni unutdingizmi?](#)

Рис. 3.9. Форма входа в систему пользователя

Если пользователь раньше не проходил регистрацию в системе, тогда необходимо пройти регистрацию с помощью кнопки, изображенной на рис. 3.9. В форме регистрации (рис. 3.10) требуется ввести свой электронный почтовый адрес, имя и новый для себя пароль. Кроме того, в форме регистрации требуется ввести фотографию пользователя размером 110x110 пиксел. При вводе фотографии указывается адрес расположения фотографии.

Если в форме регистрации ввести данные, не соответствующие действительности, введенные данные не регистрируются и указываются его недостатки.

Online Judge - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Online Judge

http://localhost/index.php?modJie=users&action=register

Online Judge
TATU Samarkand Branch

[Asosiy sahifa](#) [Masalalar arxivi](#) [Tizim holati](#) [Reyting](#)

Registrasiya

[Asosiy sahifa](#) - Registrasiya

E-mail:

Ismingiz:

Parol:

Yana bir marfa:

Rasm: (110 x 110)

*to'ldirilishi shart

Рис. 3.10. Форма регистрации пользователей

Ещё одной удобной стороной системы является то, что имеется специальная страница, в которой можно наблюдать текущее состояние пользователей, прошедших регистрацию. В этой подстранице даются сведения об имени пользователя, номер задачи по программированию, выбранном языке программирования. Обновление этих данных может быть осуществлено по договоренности пользователем или администратором. Для этого на знаке обновления на верхней стороне списка выбирается время в секундах.

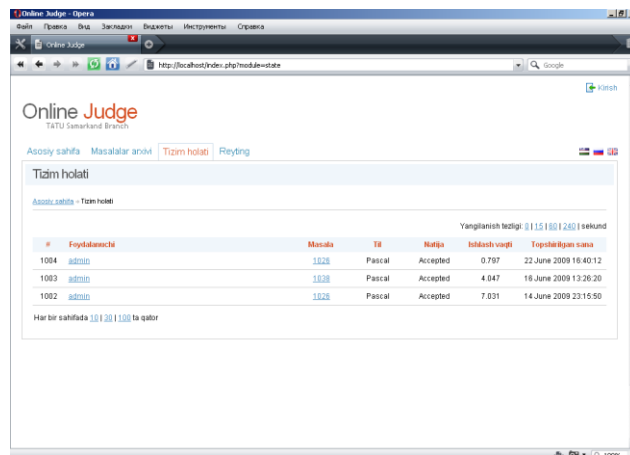


Рис. 3.11. Страница наблюдения за пользователями системы

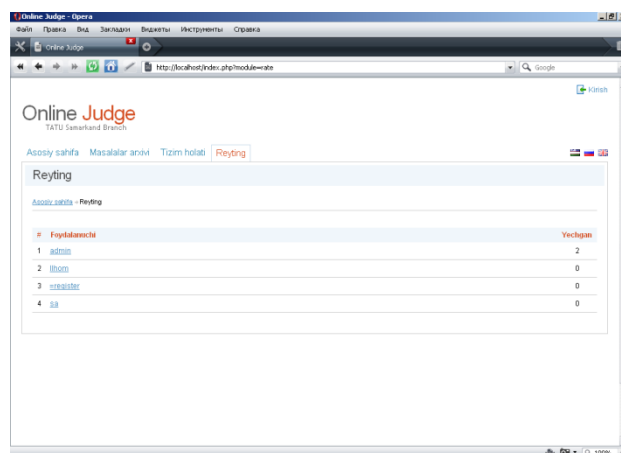
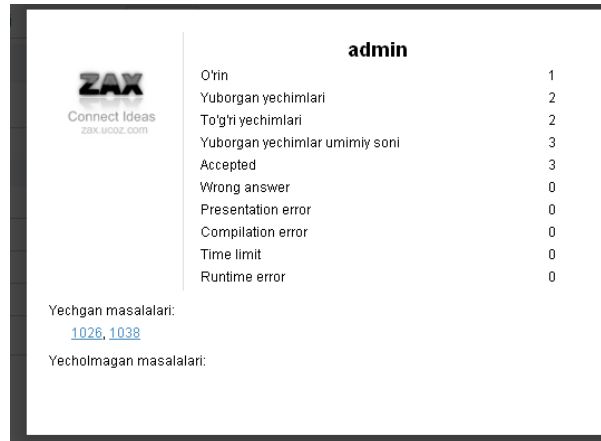


Рис. 3.12. Страница контроля рейтинга пользователей






admin	
O'rin	1
Yuborgan yechimlari	2
To'g'ri yechimlari	2
Yuborgan yechimlar umimiy soni	3
Accepted	3
Wrong answer	0
Presentation error	0
Compilation error	0
Time limit	0
Runtime error	0

Yechgan masalalari:
[1026](#), [1038](#)

Yecholmagan masalalari:

Рис. 3.13. Окно наблюдения за состоянием каждого пользователя системы

Можно следить за рейтингом каждого пользователя, использовавшего систему или передавшего серверу решение программы. Кроме того, еще одной выгодной стороной системы является то, что используется интерфейс на трех языках: узбекский, русский, английский. Для выбора языка интерфейса на форме главной страницы    выбирается соответствующий знак флага.

3.4. Виртуальная модель обучения предмета “Информатика” на основе Web-технологий

Для визуализации информации, используемой в учебном процессе, необходимо определить суть, выражающую его основное содержание, и найти график, состоящий из комплекса схем, цвета, размеров и других составляющих. В этом процессе основной целью является оживление процесса усвоения сути информации. Визуализация программ является областью предмета, и она изучает способы объяснения сути этих программ путем демонстрации. В предмете «Информатика», а также в программировании, используются простые способы визуализации. Например, при составлении программ с 1960 года используется правила “структурной записи”. Написание блоков и операторов программы в ступенчатом виде повышает их свойство понятности. Появление дисплеев и

принтеров, работающих в графическом режиме, создало возможность использовать графические изображения при визуализации. Визуализация программ вбирает в себя несколько аспектов предоставления материала в визуальном виде. В качестве примера этому, можно указать такие аспекты, как визуализация в виде текста или графика, изображения, степени абстрактности, статической или динамической визуализации. Кроме того, средства визуализации предоставляют полезную информацию без влияния на структуру или исполнение программы. Например, визуальное объяснение ошибок при использовании в программе различных видов значений оказывает большую помощь в понимании системы типов для тех, кто только начинает заниматься программированием. Поэтому при визуализации программ обращается основное внимание на их структуры и визуализацию процесса выполнения. Термин визуализации программ обычно связан со статическим представлением текста программы. В качестве примера статических изображений можно указать диаграммы, отображающие проект программы, абзацы, отдельно выделенные в тексте, примечания в виде блок-схем и другие объекты. Такие виды визуализации программы можно осуществить с помощью анализа программы, последовательной регулировки или таких программных или аппаратных средств как принтер.

Термин “анимация алгоритма” означает абстрактное изображение динамического процесса, выполняемого в программе. Один только алгоритм может быть выражен через различный программный код (текст) (например, в различных языках программирования). Анимация алгоритма означает демонстративное изображение алгоритмического процесса без связи с каким-либо программным кодом в частном случае. Однако его значение с точки зрения обучения очень велико. Он дает возможность повышения эффективности обучения основ программирования. Внизу рассмотрим примеры визуализации программ. На рис. 3.14 указана программа упорядочивания чисел методом «вставки» в список чисел (1,6,4,3). На рисунке программа предоставлена в виде графика. Таковую среду программирования можно использовать для согласования

анимаций непосредственно на компьютере или исполнения непосредственно на Web-странице. Использование офисных приложений для создания анимаций и для согласования дает возможность использования их учениками и студентами.

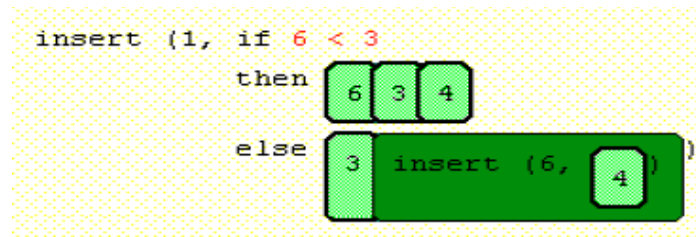


Рис. 3.14. Графическая визуализация программы

В качестве примера программных средств, используемых при решении задач визуализации, которые изображены выше, можно указать функциональную систему программирования WinHPE. На рис. 3.15 приведена анимация трех видов двоичного дерева. Они, как в офисных приложениях, получены путем обычного диалога. Если первые две анимации демонстрируют различные топографические элементы, то третий указывает на анимацию с исключением его некоторых частей.

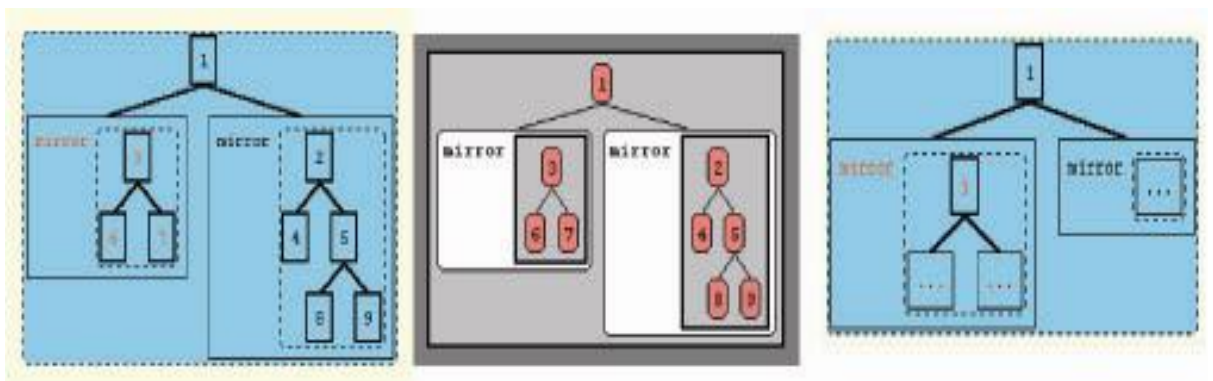


Рис. 3.15.

Главной целью анимации алгоритмов является их использование в учебном процессе. Например, при использовании системы БАЛСА [1], созданной Марком Бровном, анимация, созданная с примечаниями преподавателя, запускается на

каждом компьютере, привязанном к каждому студенту. Студент может контролировать скорость и направление анимации или менять их. Однако нельзя изменить содержание анимации.

Системой анимации алгоритмов можно пользоваться и другими способами. Например, графические анимации могут служить хорошим наглядным пособием для лекций по теме алгоритмов. Создаются такие возможности как повторение анимаций, показанных на предыдущем занятии, или контролирование системы исполнения анимаций.

При таком способе использования анимаций алгоритмов только учитель является активным участником. В некоторых случаях этот способ (например, в случае большого числа студентов) может пригодиться. Однако, тот факт, что студент не является активным участником, только наблюдает за анимациями и не может влиять на процесс можно считать недостатком этого способа. Студент в этом случае оказывается в состоянии наблюдателя телевизора или фильма. Этот недостаток может быть частично компенсирован устными пояснениями и поощрением по демонстрации анимаций и задаванием вопросов студентами.

Демонстрация анимаций алгоритмов в компьютерных классах может дать более высокую эффективность. Ученики и учитель имеют свои компьютеры и могут одновременно параллельно наблюдать за анимациями. Например, система, созданная в университете Бровна под названием САТ, предназначена для использования именно таким способом. В нем предусмотрен вывод анимации с помощью Web-браузера и на компьютер ученика, и на компьютер учителя. Учитель имеет панель контроля, с помощью которого может влиять на процесс демонстрации, на течение процесса с помощью изменения некоторых параметров. Студент имеет возможность наблюдать за алгоритмами в различном виде (устное определение, блок-схемы, анимационное выражение). Хотя он не может влиять на процесс анимации, он может изменять или отбрасывать различные виды анимации или активировать их.

Анимации являются полезным средством для самостоятельных занятий студентов. Обобщенно анимация на Web-технологиях могут дать более высокий эффект, потому что их независимость от массового использования и используемой платформы создает дополнительные возможности. Студент может повторять пройденные анимации в удобном для себя виде и скорости и в любое удобное для себя время.

Создание средств визуализации алгоритмов и их использование направлено на формирование у студентов навыков понимания сложных частей алгоритма и соображений об абстрактных процессах и вещах. Однако форма использования студентами анимаций не всегда приводит к такой цели. Например, студенты относятся к вопросам-ответам, касающихся процессов анимации, как к дополнительному средству проверки усвоения ими понятия алгоритма. Это способ, похожий на проверку знаний перед экзаменом. Настоящая цель, преследуемая анимациями, то есть направление студентов на размышление и повышение их абстрактного мышления, остается в стороне. Толкование программы анимации, созданной в среде программирования как средство проведения тестов, также присуще такому подходу, и она не дает возможность полностью усвоить процесс визуализации.

Еще одним способом использования анимации алгоритмов является их использование на лабораторных занятиях. Авторы системы анимации алгоритмов ГАИГС изложили четыре способа их использования [4]:

1. Начальные лабораторные занятия: В этом способе анимации объясняются на лабораторных занятиях перед лекционными занятиями. Студентам вместе с задачами раздаются карточки вопросов-ответов. В процессе ответа на вопросы студент получает представление об алгоритме. После этого учитель проверяет степень усвоения понятия алгоритма. Также студенту можно поручить задачу изменения (модификации) алгоритма и визуализации его нового образа.

2. Занятия по закреплению: На этих занятиях даются задания по закреплению знаний, полученных по понятию алгоритм перед лекционными занятиями.

3. Занятия по совершенствованию: Студент работает над знаниями, полученными во время лекции. Он наблюдает исполнение алгоритма для различных начальных наборов данных и разрабатывает альтернативные варианты для совершенствования его исполнения. Результаты альтернативных вариантов анализируются, и обосновывается лучшая исполнимость этого варианта.

4. Занятия по сравнению. На этих лабораторных занятиях студент анализирует различные исполнения одного алгоритма и изучает их «хорошие и плохие стороны» путем сравнения. Лабораторные задания, материалы, предоставленные студенту, должны быть направлены на заключение об эффективности исполнений алгоритма в различных вариантах.

Еще в одной форме использования систем анимации алгоритмов требуется, чтобы студент сам создавал собственные анимации [2]. Этот путь с точки зрения процесса обучения считается самым полезным, потому что при этом студент является активным участником, сам находит свои ошибки и исправляет их, в результате чего добивается больших знаний. Однако этот способ требует знания студентом какого либо языка программирования и способности правильного его применения при создании программ анимации. Поэтому если выбирать задания по уровню знаний студента вероятность достижения положительного результата будет высокой.

Когда был создан MacromediaFlash, он использовался в основном для создания анимаций, однако после того, как пользователи узнали какие виды действий можно в нем создавать, он начал широко использоваться во всех областях. Компания Macromedia, учитывая предложение об усилении инструментария программы Flash, превратили его из простой программы создания анимаций в среду, создающую мощные программные средства. Теперь пользователь может использовать не только возможности анимации, но и пользоваться не воображаемыми интерактивными возможностями.

На сегодняшний день создание только анимаций, какими бы они не были феноменальными, не соответствует требованиям, интерактивность при этом очень простая. А ActionScript создает следующие дополнительные возможности:

- Индивидуализация свойств для каждого пользователя;
- Возможность полного управления фильмом и его параметрами;
- Анимация элементов фильма посредством программы (не используя монтажный стол);
- Вывод данных из Флэш и прием данных (создание форм, блоков и так далее) из внешних источников;
- Создание динамических проектов, связанных с текущим временем и датой;
- Управление высотой звука и его балансом, а также еще много других возможностей.

Дополнительно можно сказать, что во Flash можно создавать не только веб материалы, но и самостоятельные программы и мини игры. Для этого требуется знание хотя бы основ ActionScript.

ActionScript – это такой специальный язык, посредством которого мы можем объяснить Flash то, что от него хотим. При этом он дает возможность работы не только с простыми командами и указаниями, но и ввода команд, основанных на логическое мышление. Как и во всех других языках, ActionScript имеет свою структуру, синтаксис, ключевые слова и символы. Их правильное использование приводит к достижению высокой эффективности.

Событие - это такой объект, который при демонстрации фильма выполняет конкретный скрипт. Например, событие для кнопки *on (release) { ... }* происходит при нажатии кнопки мышки. Каждый скрипт выполняется при происшествии какого-либо события. А в фильме событий может быть много: нажатие кнопок, издание звука и так далее.

Действия играют основную роль при формировании скрипта. Обычно под действием понимается строка, состоящая из приказа и указания.

Например:

```
sny.c=0;
```

```
sny.gotoAndPlay("True");
```

Действия – это набор строк в скрипте. Они пишутся между “{” и обычно заканчиваются знаком “;”.

Функции - это подпрограммы, используемые в основной программе.

Синтаксис объявления функций:

```
1.      function myFunction(arg1, arg2, ...) {
```

```
// выполняемые действия;
```

```
}
```

```
2.      myFunction=function(arg1, arg2, ...){
```

```
/* действия */ }
```

Вызов функции без параметра: *myFunction()*;

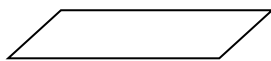
Вызов функции с параметром *myFunction(значение arg1, значение arg2, ...)*;

[12].

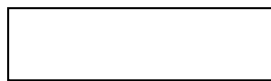
Алгоритм – это упорядоченная последовательность приказов, необходимых для решения какой-либо задачи. Они обычно изображаются с помощью блок-схемы. Блок-схемы состоят из следующих элементов:



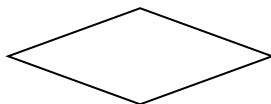
- начало и конец алгоритма



- ввод начальных данных



- вычисление арифметических и логических выражений



- изменение управления вследствие какого-либо условия, процесс повторения



- вывод результатов на внешнее устройство

Алгоритмы в основном делятся на 3 вида: *линейные, разветвляющиеся и повторяющиеся (циклические)*.

В алгоритмах, имеющих линейное строение указания, выполняются в порядке следования. Их блок-схемы создаются с помощью блоков начала, остановки, ввода-вывода и блоков заранее известных процессов, которые располагаются последовательно в одну линию.

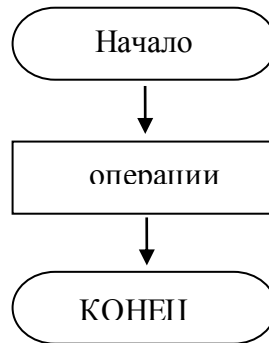


Рис. 3.16.

Для решения задачи создания линейного алгоритма надо сперва определить имена переменных, составляющих начальные данные, их вид и диапазон изменения. Затем следует выяснить имена, тип и диапазон изменения промежуточных и итоговых переменных. Теперь остается выяснить, какой алгоритм будет обрабатывать эти начальные данные и будет получать промежуточные и итоговые результаты [10].

Пример 1. Условия задачи: *Заданы действительные числа a и b . Найдите их сумму.*

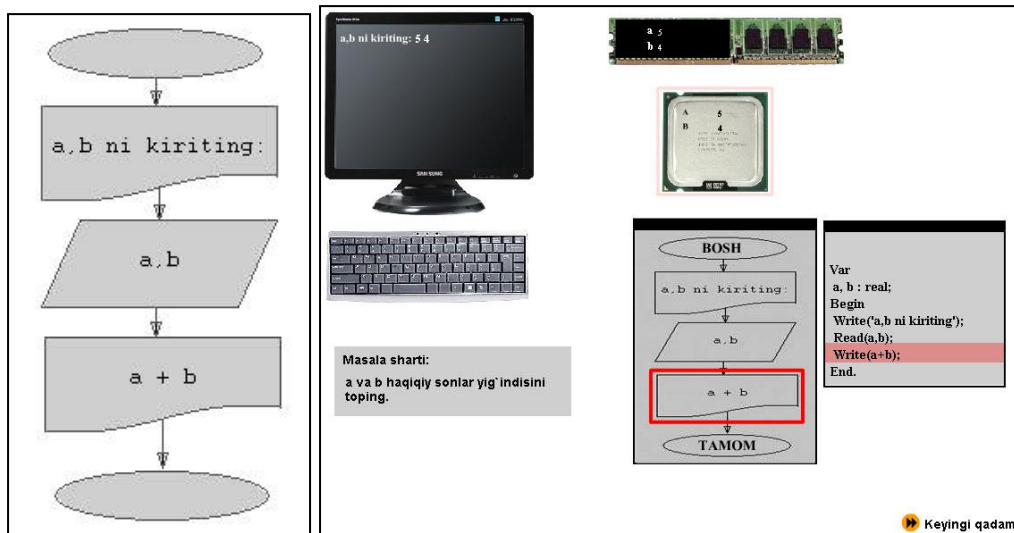


Рис. 3.17. Визуализации линейных алгоритмов

Вид на алгоритмическом языке Паскал:

var

a, b :real;

Begin

Write('Введите a и b ');

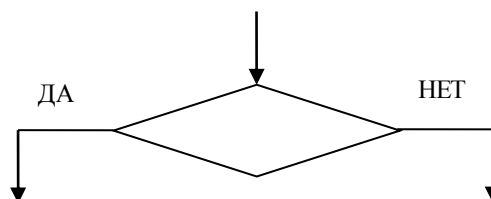
Read(a,b);

Write(a+b);

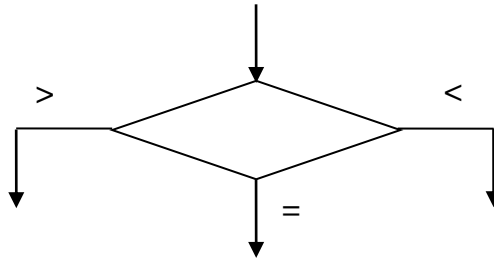
End.

При решении различных задач порядок выполнения указаний связан с исполнением какого-либо условия, то есть алгоритм разветвляется. Разветвление выражается через блок „изменение управления при условии“.

Результат проверки условия бывает двух видов: для положительного исхода “да”, для отрицательного исхода “нет”.



Когда разветвление является знаком значения математического выражения (арифметическое условие): ставятся знаки „>“ – больше, „<“ – меньше и „=“ – равно [10].



Пример 2.

Условие задачи: *Заданы действительные числа x и y . вычислите z , при этом если $x > y$, то $z = x - y$, иначе $z = y - x + 1$.*

Блок-схема:

Визуальное изображение:

Masala sharti:
Haqiqiy x va y berilgan. z ni hisoblang, bunda agar $x > y$ bo'lsa, $z = x - y$, aks holda $z = y - x + 1$

Keyingi qadam

Вид на алгоритмическом языке Паскаль:

`var`

`x, y, z : real;`

`begin`

`Write('введите x y:');`

`Read(x,y);`

`if x > y then`

`z := x - y`

else

$z := y-x+1;$

Writeln(z);

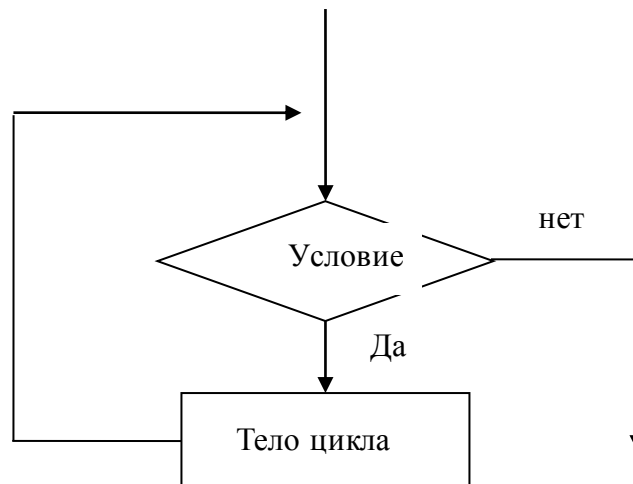
end.

Повторяющиеся алгоритмы вбирают в себя много раз повторяющуюся часть, которая называется *телом цикла*. Повторение выполняется пока выполняется какое-либо условие.

Повторяющиеся алгоритмы бывают трех видов: *циклические, итерационные и бесконечные*.

Циклические алгоритмы возникают, когда переменная повторения изменяется согласно арифметической прогрессии.

Во многих случаях невозможно определить сколько раз выполняются итерационные алгоритмы. Повторение продолжается до тех пор, пока выполняется конкретное условие:



Бесконечные алгоритмы используются в языках программирования, где имеются операторы, позволяющие вести диалог с ЭВМ [10].

Пример 3.

Условие задачи: *Задано натуральное число n . Найдите $n!$.*

Вид на алгоритмическом языке Паскаль:

```

var
i, n, f : integer;
Begin
Write('Введите n:');
read(n);
i := 1;
f := 1;
  While i <= n do
f := f * i;
  Write(f);
End.

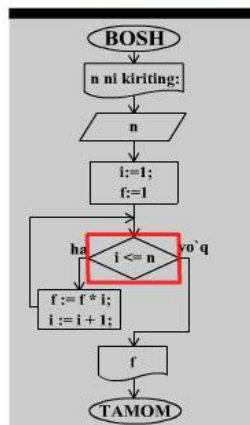
```

Блок-схема:



Masala sharti:
Naturan n soni berilgan. n! ni toping.

Визуальное изображение:



```

Var
i, n, f : integer;
Begin
Write('n ni kiriting:');
Read(n);
i := 1;
f := 1;
While i <= n do
  f := f * i;
Write(f);
End.

```

▶▶ Keyingi qadam

Каждый сложный алгоритм можно представить в виде вышеуказанных 3 структур.

Таким образом, виртуальная модель, представленная в этом, осуществляет следующее:

- были изучены и проанализированы существующие системы визуализации алгоритмов, использующие семантическую модель учебного процесса;

- было изучено их применение в учебном процессе;

- были изучены методы и необходимые программные системы для создания анимационных изображений;

- были выбраны образцовые задачи, касающиеся программирования согласно учебной программе предмета «Информатика», и процесс их исполнения был визуализирован с помощью анимационных изображений на базе семантической модели.

Основным результатом данного раздела магистерской диссертации являются предложенные визуальные изображения процессов выполнения линейных, разветвляющихся и повторяющихся алгоритмов (программ), которыми можно пользоваться на занятиях по предмету «Информатика» на 1 курсе.

В созданных анимациях была создана функция для обработки данных введенных пользователем:

```
_global.trim = function(a) {
  t = true;
  while (t) {
    if (a.charAt(0) == " ") {
      a = a.slice(1);
    } else {
      t = false;
    }
  }
  t = true;
  while (t) {
    if (a.charAt(a.length-1) == " ") {
      a = a.substr(0, a.length-1);
```

```

    } else {
        t = false;    }
    }
    return a;
};

```

Эта функция удаляет лишние пробелы во введенных строках (которые пользователь ввел по незнанию). Следующий код выполняется при нажатии кнопки Enter, он делит введенные строки на числа. Если данных достаточно, то логической переменной *entered* присваивается значение *true*:

```

on (keyPress "<Enter>") {
    _parent.output._visible = true;
    Selection.setFocus(_parent.output);
    str=trim(_parent.input.text)
    a=str.split(" ",3)
    if (a.length=3){
        _root.ram.m1=a[0];
        _root.ram.m2=a[1];
        _root.ram.r=a[2];
        _root.cpu.m1=a[0];
        _root.cpu.m2=a[1];
        _root.cpu.r=a[2];
        _parent.entered=true;
    }
}

```

Следующая строка указывает на функцию, которая должна быть выполнена при изменении значения переменной *entered*:

```

_root.mon.watch("entered",watchMonitor);

```

При изменении значения переменной *entered* выполняются действия, которые находятся в теле следующей функции *watchMonitor()*.

```
function watchMonitor(id,oldval,newval){  
    _root.ram.gotoAndPlay("read");  
    _root.key.gotoAndPlay("default");  
    _root.ram.ramactivate.gotoAndPlay("active");  
}
```

ActionScript код, использованный при визуализации разветвляющегося алгоритма:

```
stop();  
if (a>b){  
    bs.gotoAndPlay("true");  
    code.gotoAndPlay("true");  
    cpu.gotoAndPlay("calculate");  
}  
else{  
    bs.gotoAndPlay("false");  
    code.gotoAndPlay("false");  
    cpu.gotoAndPlay("calculate");  
}
```

В соответствии с выполнением или не выполнением условия фильмы переходят в соответствующий кадр. В циклическом процессе выполняются точно такие же действия.

Выводы по III главе

1. Разработан специальный пользовательский интерфейс по виртуализации учебного процесса, и была создана система организации диалога с пользователем.

2. Разработана и внедрена технология проведения дистанционного тестового контроля, основанного на Web-технологиях, и система использования его программного обеспечения в учебном процессе в разрезе конкретных предметов.

3. Разработано программное обеспечение автоматизированной системы, тестирующей результаты работы программ, основанных на Web-технологиях, и были определены принципы его внедрения в учебный процесс.

4. Предложена виртуальная модель обучения предмету “Информатика и информационные технологии”, основанная на Web-технологиях, и было создано программное обеспечение по внедрению этой модели непосредственно в учебный процесс.

ЗАКЛЮЧЕНИЕ

Научно-практические результаты, полученные в настоящей магистерской диссертационной работе, были сформированы в качестве математического и программного обеспечения виртуализации учебных процессов на основе Web-технологий, и на их основе были сделаны следующие заключения:

1. На основе системного анализа проанализированы возможности информационно-коммуникационных технологий в организации виртуальных процессов, на их базе была разработана базовая концепция по их использованию ИКТ в учебных процессах.

2. На основе Web-технологий была создана интеллектуальная система управления учебными процессами и были определены принципы их внедрения. Определены программные среды, используемые для виртуализации учебных процессов на базе Web-технологий, с использованием семантических и математических моделей. Сформулированы цель магистерской диссертации и основные задачи, решаемые в его рамках.

3. На основе общего определения электронных учебных ресурсов разработаны дискретные математические модели виртуализации УП и на их базе созданы соответствующие алгоритмы для реализации программного обеспечения. Разработана технология создания семантических моделей электронно-учебных ресурсов и на их базе определены закономерности отношений между контентом электронного учебного ресурса.

4. Разработана многоэтапная математическая модель, учитывающая семантику УП, и были изучены свойства адекватности и устойчивости этих моделей. Была определена роль семантической модели в виртуализации учебного процесса и разработаны алгоритмы их внедрения.

5. Создано соответствующее программное обеспечение для размещения электронных учебных ресурсов, в систему образования, основанного на Web-технологиях.

6. Разработан специальный пользовательский интерфейс по виртуализации учебного процесса, использующий семантические и другие виды моделей, и была создана система организации диалога с пользователем.

7. Разработана и внедрена технология проведения дистанционного тестового контроля, основанного на Web-технологиях, и система использования его программного обеспечения в учебном процессе в разрезе конкретных предметов.

8. Разработано программное обеспечение автоматизированной системы, тестирующей результаты работы программ, основанных на Web-технологиях, и были определены принципы его внедрения в учебный процесс.

9. Предложена виртуальная модель обучения предмету “Информатика и информационные технологии”, основанная на Web-технологиях, и было создано программное обеспечение по внедрению этой модели непосредственно в учебный процесс.

ЛИТЕРАТУРА

1. Каримов И.А. Юксак маънавият – енгилмас куч. –Тошкент: «Маънавият», 2008. -176 б. (на русском...)
2. Каримов И.А. Жаҳон молиявий-иқтисодий инқирози, Ўзбекистон шароитида уни бартараф этишнинг йўллари ва чоралари. – Т.: Ўзбекистон, 2009, 56 б.
3. Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлиги Хайъатининг 2009 йил 21 февралдаги 2/7-сонли қарори. 3-илова.
4. “Компьютерлаштиришни янада ривожлантириш ва ахборот-коммуникация технологияларини жорий этиш чора тадбирлари тўғрисида” Ўзбекистон Республикаси Вазирлар Маҳкамасининг қарори. Тошкент ш. 2002 йил 6 июн.
5. Аванесов В.С. Композиция тестовых заданий М.Ж Высш.обр., 1998.
6. Аванесов В.С. Математические модели педагогического измерения. М. Высш.обр.,1994.
7. Аванесов В.С. Научные проблемы тестового контроля знаний. М.: Высш.обр., 1994. – 203 с.
8. Аванесов В.С. Теоретические основы разработки заданий в тестовой форме. М. : Вис. Обр. 1995. – 189 с.
9. Аванесов В.С. Форма тестовых заданий. М.: Высш.обр., 1991. – 167 с.
10. Андреев А.Б., Моисеев В.Б., Усачев Ю.Е., Усманов В.В. Концептуальный подход к созданию интеллектуальной системы анализа знаний. М.: Высш.обр., 2001
11. Аргерих Л. и др. Профессиональное РНР программирование, 2-е издание. – Пер. с англ. – СПб: Символ – Плюс, 2003 – 1048 с., ил.
12. Башмаков, А.И. Разработка компьютерных учебников и обучающих систем. [Текст]./А.И.Башмаков, И.А.Башмаков – М.: Информационно-издательский дом «Филинь». – 2003. – 616 с.

13. Богданов И.В., Крутки И.А., Чмыхова Е.В. Проектирование учебного процесса на базе современных информационных технологий. М.,2001.
14. Бойназаров И.М. Ўқув жараёнини бошқаришнинг Web – технологияларга асосланган интеллектуал тизимини яратиш. //ТАТУ хабарлари журнали 2009 йил 2-сон, 108-111-б.
15. Бойназаров И.М. Электрон ўқув ресурсларини Web-саҳифаларга жойлаштиришнинг дастурий таъминоти. //Таълим муаммолари журнали 2009 йил 1-сон.
16. Бойназаров И.М., Саидов О.О. «BookCreator» - электрон дарслик ва китоблар яратиш учун дастурий восита. ЭҲМ учун яратилган дастурга муаллифлик гувоҳномаси. №DGU 01966. Тошкент. 08.06.2010.
17. Брусиловский П.Л. Адаптивные обучающие системы в World Wide Web: обзор имеющихся в распоряжении технологий.//Авт.пер.на сайте <http://ifets.ieee.org/russian/depository/WWWITS.html>.
18. Васильев В.Н., Парфенов В.Г. Командный чемпионат мира по программированию АСМ 1998/99. Северо-восточный Европейский регион. СПб.: СПбГИТМО(ТУ), 1998.
19. Внедрение виртуального образовательного процесса при обучении технических дисциплин // Международная научная конференция НУУ, 2009г.
20. Гаевский А.Ю., Романовский В.М. 100% самоучитель по созданию Web-страниц и Web-сайтов. Учебное пособие. М.: Технолоджи – 3000. 2005, 464 с.
21. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2008. - 366 с.
22. Гордеев А.В. Операционные системы. Учебник для вузов. 2-е издание, 416 с. СПб.: Питер. 2007.
23. Гофман В. Э., Хомоненко А.Д. Работа с базами данных в Delphi. -СПб.: БХВ-Петербург, 2001. -656 с.

24. Гроднева С., Заберин Ю. Интернет в Вашем доме. М.: «Рипол Классик», 2001, 480 с.
25. Гусева А.И., Смольникова И.А., Филиппов С.А., Чиркова М.А. Основы применения ИКТ в учебном процессе. М.: Академия АЙТИ, 2005.- 220 с.
26. Доррер А.Г. Модель процесса интерактивного обучения как автомат с линейной тактикой. /А.Г.Доррер //Материалы III Всероссийской научно-практической конференции студентов «Молодежь и современные информационные технологии». ТомскЖ ТПУ. -2005.
27. Доррер, А.Г. Динамическое моделирование процесса интерактивного обучения. /А.Г.Доррер //Материалы Всероссийской научно-практической конференции «Лесной и химический комплексыЖ проблемы и решения» - КрасноярскЖ СибГТУ. – 2005. – с.253-258.
28. Доррер, А.Г. Моделирование и разработка интерактивных обучающих систем адаптацией. /А.Г.Доррер //Материалы IX Всероссийской научно-практической конференции «Проблемы информатизации региона. ПИР-2005» - КрасноярскЖ ИПЦ КГТУ. – 2005. –с.147-152.
29. Доррер, Г.А. Моделирование вычислительных системЖ Учебное пособие. – Красноярск: КГТУ. – 2004. -188с.
30. Евтюхин Н.В. Структуризация знаний и технология разработки компьютерных мастер-тестов. М.: Высш.обр., 1999.
31. Зайцева Л.В. Методы и модели адаптации к учащимся в системах компьютерного обучения //Educational Technology & Society 6(4) 2003. ISSN 1436-4522.–Р.204-211.
32. Ивлиев М.К. Разработка тестовых заданий для компьютерного тестирования. М.: Высш.обр., 2001.
33. Ивлиев М.К., Блинов А.В., Беленький Я.И. Методические рекомендации для пользователей автоматизированной системы контроля знаний по сценариям на естественном языке СОАВТОР2.2. М.: Высш.обр., 1989.

34. Ивлиев М.К., Кинжер А.И., Новиков В.А. Методические рекомендации по составлению тестовых заданий и использованию автоматизированной системы контроля знаний. М.: Высш.обр., 1997.
35. Информационная технология управления. Учебное пособие для вузов. /Под ред. Б.А. Титоренко. 2-е изд. доп. –М.:ЮНИТИ – ДАНА, 2003. – 439с
36. Кабальнов Ю.С., Минасов Ш.М., Тархов С.В. Модели представления и организация хранения информации в сетевой информационно-обучающей системе // Вестник. – Т.5. – №2(10), – Уфа, 2004.
37. Капустин М.А., Капустин П.А., Копылова А.Г. Flash MX для профессиональных программистов. Интернет-университет информационных технологий - ИНТУИТ.ру, 2006
38. Карпенко Д.С., Карпенко О.М., Шлихунова Е.Н. Система автоматического повышения качества тестовых заданий и мониторинг процесса усвоения знаний. М.: Высш.обр., 2001.
39. Карпенко М. П. Дистанционные технологии – массовое образование XXI века // Образование в России. Федеральный справочник. – М., 2001. С. 309.
40. Карпенко М.П. Проблемы измерения знаний и образовательные технологии. М.: Высш.обр., 1997.
41. Карпов Б.И. Delphi: Специальный справочник. СПб.: Питер, 2001. -648 с.
42. Карпова Т. С. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2001. -304 с.
43. Колин К. К. Фундаментальные основы информатики: социальная информатика. – М.: Академический Проект; Екатеринбург: Деловая книга, 2000. С. 36.
44. Колискиченко Д.Н. РНР 5. В теории и на практике. Самоучитель. Изд-во «Нит» Санк-Петербург, 2007, 631 с.
45. Колисниченко Д.Н. Самоучитель РНР5.–СПб.: Наука и Техника, 2007.– 640 с.
46. Коршунов С. В. Вливаясь в болонский процесс // Инженерное образование. Москва, Знание, 2004. №4.

47. Котеров Д.В. Самоучитель PHP4. – СПб.: БХВ – Петербург, 2003. – 576 с.:
48. Кузнецов М.В. MySQL 5 (+CD). СПб.: BHV-Санкт-Петербург, 2006.-1024 с.
49. Куссуль Н.Н., Шелестов А.Ю. Использование PHP. Самоучитель. – М.: Издательский дом «Вильямс», 2005. – 272 с.
50. Курейчик В.М., Зинченко Л.А. Эволюционная адаптация интерактивных средств открытого образования // Открытое образование. – 2001. – № 1. – С.43-50.
51. Минасоев Ш.М., Тархов С.В. Проект «Гефест» – один из вариантов практической реализации технологий электронного обучения в вузе в условиях интеграции традиционного и дистанционного обучения // Educational Technology & Society 8(1) 2005. ISSN 1436-4522. – P.134-147.
52. Мазуркевич А. PHP: настольная книга программиста. – Мн.: Новое знание, 2003. – 480 с. ил.
53. Малахоткина И.Е. – Информационные технологии в преподавании естественнонаучных дисциплин, МОУСОШ №3 г. Мирный. -234 с.
54. Марков А.В. Как учат экономистов в университетах США//М.: Высш.обр., 1991. - №1
55. Мясникова О.В., Новикова М.Б., Прончев Г.Б. О методическом обеспечении изучения основ программирования. /в кН. «Информатизация образования – 2009Ж материалы Международной научно-методической конференции». – ВолгоградЖ Изд-во ВПГУ «Перемена», 2009, с.269-272.
56. Назиров Ш.А., Кравченко С.Е., Бабакулов И.Х. Автоматизированная система проверки решений для организации соревнований по программированию среди студентов и школьников в локальных сетях и сети Интернет. //Журнал: Информатика ва энергетика муаммолари, Тошкент, 2008 йил 6-сон. 54-60 б.
57. Нейман Ю.М., Хлебников В.А. Введение в теорию моделирования и параметризации педагогических тестов. М.: Высш.обр., 2000.
58. Норенков И.П. Стандартизация в области компьютерных образовательных технологий. // Информационные технологии. – 2003. – № 1. – С.36-40.

59. Околелов О.П. Электронный учебный курс // Высшее образование в России. 1999. №4.
60. Олифер Б.Г. Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов. 2-е изд. -СПБ.:Питер, 2005.-864с.
61. Переверзев В.Ю. Критериально-ориентированные педагогические тесты для итоговой аттестации студентов М.: Высш.обр., 1999
62. Полат Е.С., Бухаркина М.Ю., Моисеева М.В., Петров А.Е. Новые педагогические и информационные технологии в системе образования. –М.: Издательский центр «Academia», 2003.-215с.
63. Применение виртуального учебного процесса в обучении естественнонаучных дисциплин // InfoCOM2009 г.
64. Раджабов Б.Ш., Бойназаров И.М. Web-технологиялар асосида масофавий тест-назорат ўтказиш технологияси. //Тошкент. Informatika va energetika muamolari журнали. 2009 йил №4-сон, 74-78-бетлар.
65. Раджабов Б.Ш., Бойназаров И.М., Мамажанов Р.Я. Web-технологиялар асосида виртуал ўқув қўлланмалар яратишнинг дастурий интерфейсини ишлаб чиқиш. //Тошкент. ТАТУ хабарлари 2008 йил №4-сон, 101-103-б.
66. Раджабов Б.Ш., Мамажонов Р.Я., Медетов С.К. “Програмный инструментарий для решения задач стохастической оптимизации”. Иқтисодий ахборотларни қайта ишлаш ва ҳимоя қилишни таъминловчи технологияларни яратиш муаммолари ва ечимлари мавзусидаги Республика илмий-амалий анжумани. Тошкент, 2009.- 190б.
67. Раджабов Б.Ш., Пазылов М.С., Исмаилов А.А. Дастурий маҳсулотларни ҳимоялашнинг кўпбосқичли алгоритми. //ТАТУ Хабарлари журнали, 2008 йил 1-сон, 59-62-бетлар.
68. Раджабов Б.Ш., Хидирова Ч.М. Ўқув жараёнини виртуаллаштириш тизиминининг компоненталари ва дастурий таъминоти. Тошкент, ТАТУ хабарлари, 2009, №3.

69. Растригин Л.А., Эренштейн М.Х. Адаптивное обучение с моделью обучаемого. - Рига: Зинатне, 1988. - 160 с.
70. Рафиқов И.А., Бобомуродов О.Ж. Эксперт тизимлари тамойилларига асосланган интеллектуал ўқитиш тизими тузилмаси ва модели. //Журнал: ТАТУ хабарлари, 2008 йил 3-сон. 25-27 бетлар.
71. Роберт И. Современные информационные технологии в образовании: дидактические проблемы; перспективы использования.- М: Школа-Пресс, 1994.-205с.
72. Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам: Пер. с англ. - М.: Наука. Гл. ред. физ.-мат. лит., 1986. - 496 с.
73. Селевко Г.К. Современные образовательные технологии. -М: Народное образование, 1998г.-255с.
74. Семкин В.А. Семантическая модель контента образовательных электронных изданий. //Автореферат диссертация. Тюмен, 2004 г.
75. Смирнова И.Е. Начала Web-дизайна. – СПб.: БХБ-Петербург, 2005. –256 с.
76. Соловов А.В. Дидактический анализ проблематики электронного обучения //Труды Междунар.конф. «IEEE International Conference on Advanced Learning Technologies».– Казань: КГТУ, 2002. – С. 212-216.3.
77. Тартаковский Л.М., Тартаковский А.Л. Использование ресурсов Интернет для создания комплекса учебных материалов. – Москва, 2004, Международная научно-практическая конференция «Проблемы и перспективы сотрудничества государств-участников СНГ в реформировании единого (общего) образовательного пространства», - 520с.
78. Тартаковский Л.М., Тартаковский А.Л. Применение ресурсов Интернет для создания комплекса учебных материалов. – Москва, 2005, 2-я Всероссийская научно-практическая конференция «Образовательная среда сегодня и завтра», Тезисы докладов,-131с.

79. Тартаковский Л.М., Тартаковский А.Л. Создание и использование учебных материалов с помощью ресурсов Интернет. – Москва, 2003, Российская академия образования. Институт информатизации образования, Ученые записки, вып. 8, -176с
80. Тархов С.В. Управление адаптивным обучением и его оптимизация на базе теории абстрактных автоматов и Марковских процессов//Информационные технологии моделирования и управления. Научн.-технич. Журнал. – №1(19). – Воронеж.: Научная книга, 2005. – С.39-45.
81. Ульман Л. Основы программирования на PHP: Пер. с англ. – М.: ДМК Пресс, 2001. – 288 с.: ил. (Самоучитель)
82. Учебный процесс и контроль знаний в системе виртуального образования //InfoCOM. 2009 г.
83. Фазылов Ш.Х., Хусаинов Н.О., Махмудова Д.М. Метод построения модели обучаемого для создания автоматизированных обучающих систем. //Журнал: Информатика ва энергетика муаммолари, Тошкент, 2008, 4-сон. 3-6 б.
84. Фаронов В. В. Delphi 2005. Разработка приложений для баз данных и Интернета. - СПб.: Питер, 2006. -603 с.
85. Харрис Э. PHP/MySQL для начинающих. /Пер. с англ. – М.: КУДИЦ – ОБРАЗ, 2005, - 384 с.
86. Холмуродов Р.И., Лутфуллаев М.Х. Замонавий ахборот технологиялари асосида ўқитиш. //Монография. Тошкент. «Фан» нашриёти, 2003 й.
87. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы Web-технологий. Интернет-университет информационных технологий - ИНТУИТ.ру, 2003
88. Человеческий фактор. Вбт.Т.3. Моделирование деятельности, профессиональное обучение и отбор операторов: Пер. с англ./ Холдинг Д., Голдстейн Н., Эбертс Р. и др. (Часть 2.Профессиональное обучение и отбор операторов). - М.: Мир, 1991. - 302 с.
89. Чельшкова М.Б. Применение математических моделей для разработки педагогических тестов. М.: Высш.обр., 1995. -154 с.

90. Чельшкова М.Б. Теория и практики конструирования педагогических тестов. М.: Высш.обр., 2002
91. Чошанов М.А. Сертификация тяжелых учителей математики в США . М.: Высш.обр., 2000.
92. Hiltz, S. R. Correlates of Learning in a Virtual Classroom //International Journal of Man-Machine Studies. 39, 1993. P.71–98. 1981. P.c.5–10.
93. <http://www.webclub.ru/> - Российский клуб Web-мастеров.
94. <http://www.webclub.ru/> - Российский клуб Web-мастеров.
95. Jensen, K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use. [Текст]. – Berlin: Springer. – Vol.1 – 1996, Vol.2.- 1997, Vol.3 – 1997.
96. Radjabov B.Sh., Medetov S.K., Boynazarov I.M. Realization of software tools “Stochastic optimization” in optimization process of data transmission network. //International Conference on Asia 2009 in conjunction with International Summit on Information and Communication technologies. September 21-25, 2009. TUIT. 310-313-б.
97. Trentin G. Telematics and on-line teacher training: the POLARIS Project // Journal on Computer Assisted Learning. BlackwellScienceLtd. V. 13, 1997. P. 261 – 270.

ПРИЛОЖЕНИЯ