# Дипломная работа

# на тему

# Информационная система обучения по курсу «Компьютерные сети»

# Содержание

[Введение](#_Toc191839248)

[Оценка решений существующих информационных систем обучения](#_Toc191839249)

[Введение в информационные системы и их классификация](#_Toc191839250)

[Требования, предъявляемые к информационным системам](#_Toc191839251)

[Компоненты и структура ИС](#_Toc191839252)

[Понятие информационно–образовательной среды](#_Toc191839253)

[Информационные системы в образовании](#_Toc191839254)

[Автоматизированные обучающие системы](#_Toc191839255)

[Модели обучения автоматизированных обучающих систем](#_Toc191839256)

[Обзор информационных систем обучения](#_Toc191839257)

[Разработка алгоритма обучения и компонентов информационной системы](#_Toc191839258)

[Концепция ИС](#_Toc191839259)

[Проектирование ИС](#_Toc191839260)

[Разработка структуры базы данных](#_Toc191839261)

[Разработка интерфейса обучающего курса](#_Toc191839262)

[Разработка системы управления курсом](#_Toc191839263)

[Тестирование ИС](#_Toc191839264)

[Заключение](#_Toc191839265)

[Список использованной литературы](#_Toc191839266)

[Приложение. Исходные коды модулей информационной системы](#_Toc191839267)

**Введение**

На сегодняшний день необходимым условием продвижения в сфере информационных технологий является широкое внедрение стандартов и технологий информационных систем, используемых как для аппаратных средств, так и для программных продуктов. Построение программного обеспечения (ПО) вычислительных и информационных комплексов, основанных на идеологии открытых систем, позволяет успешно решать задачи переносимости ПО на платформы различных производителей, проблемы взаимозаменяемости узлов и устройств и, что самое главное, обеспечивает интеграцию устройств и пользователей в различные информационно–вычислительные и телекоммуникационные сети. Следует особо подчеркнуть то обстоятельство, что на сегодняшний день успешная реализация сколько–нибудь существенных проектов в области информационно–вычислительной техники, управления, информатизации и телекоммуникаций не представляется возможной без согласования разработок с существующими стандартами в области информационных систем и, в ряде случаев, разработки новых стандартов.

В условиях перехода к интегрированным вычислительно–телекоммуникационным системам принципы информационных систем составляют основу технологии интеграции, создания отраслевых, региональных и национальных информационных инфраструктур и их взаимодействия в глобальном масштабе. Таким образом, можно сделать вывод, что технологии информационных систем в настоящее время является той рабочей средой, в рамках которой происходит развитие приоритетных информационно–телекоммуникационных технологий, средств телекоммуникаций и вычислительной техники.

Объектом нашего исследования является информационная система обучения.

Предмет – технология процесса обучения по курсу компьютерные сети.

Цель данного дипломного проекта – разработать программное и информационное обеспечение информационной системы обучения по курсу «Компьютерные сети», используя новые информационные технологии и показать значимость и удобство этой системы для процесса обучения.

Задачи исследования:

1. Обзор и сравнительный анализ существующих информационных систем обучения.
2. Разработка структуры информационной системы по курсу компьютерные сети.
3. Разработка программного обеспечения информационной системы.
4. Разработка контента информационной системы.

# 1. Оценка решений существующих информационных систем обучения

## 1.1 Введение в информационные системы и их классификация

Перед рассмотрением конкретных информационных систем дадим несколько необходимых определений:

**Система** (в предметной области) – это множество взаимосвязанных элементов, каждый из которых связан прямо или косвенно с каждым другим элементом, а два любые подмножества этого множества не могут быть независимыми, не нарушая целостность, единство системы.

**Элемент системы** – это простейшая структурная составляющая системы, которая в рамках данной системы не структурируется.

**Структура системы** – это совокупность устойчивых связей, способов взаимодействия элементов системы, определяющая ее целостность и единство.

**Среда** (в предметной области) – это все, что находится в предметной области за границами системы.

Под **информационным процессом** будем понимать процесс, связанный с изменением количества информации в системе в результате целенаправленных действий при решении поставленной задачи.

Информационная деятельность связана с созданием информационных моделей всех объектов и явлений природы и общества, участвующих в человеческой деятельности, а также с созданием моделей самой этой деятельности.

Как известно, информация как продукт интеллектуальной деятельности человека является ресурсом и с течением времени накапливается, хотя возможны и потери информационных ресурсов.

В процессе познавательной деятельности мы, так или иначе, сталкиваемся с процессом использования накопленного знания, которое становится ценным лишь тогда, когда становится доступным широкому кругу пользователей. В настоящее время, объем информационных потоков, несущих эти знания, существенно увеличился, поэтому стала актуальной задача информатизации различных видов человеческой деятельности. Данное направление включает в себя развитие аппаратных средств и информационных технологий.

**Информационные технологии** (ИТ) – совокупность методов и средств реализации информационных процессов в различных областях человеческой деятельности. Иначе говоря, ИТ есть способ реализации информационной деятельности.

К современным ИТ относят:

* развитие глобальных информационных систем;
* внедрение систем автоматизированной обработки информации;
* развитие систем и средств дистанционного доступа;
* интегрирование гетерогенных систем;
* развитие систем искусственного интеллекта и т.д.

Долгое время преобразование информации и принятие решений являлось функцией человека. Сейчас, когда рост объемов информационных потоков привел к тому, что они превысили объемы усвояемости и обрабатываемости информации человеком, возникла проблема повышения эффективности процессов преобразования информации, определяемая следующими причинами:

* любая информация ценна только в процессе ее использования и при резком возрастании объемов информации принятие решений становится затрудненным, а также возрастает время обработки информационного массива;
* усложнение внутренней структуры системы, появление суперсистем, включающих целые совокупности систем, интеграция гетерогенных систем также приводит к резкому увеличению объемов информационных потоков и времени на их обработку;
* расширение сфер применения ИТ приводит к возникновению новых систем, что, в свою очередь, является дополнительным источником увеличения информационных потоков;
* повышение сложности задач, требуемой для их решения точности и оперативности, приводит к опережающему росту сложности управления по отношению к росту возможностей обработки информации и так далее.

Определим два основных пути развития ИТ, обеспечивающих повышение эффективности процессов преобразования информации в информационных и информационно–управляющих системах:

* совершенствование технических средств автоматизации на основе применения высокопроизводительных вычислительных устройств и систем, что приводит к повышению скорости обработки информации вне зависимости от характера преобразуемой информации;
* совершенствование и расширенное внедрение программного обеспечения.

Для реализации указанных путей необходимо наличие наиболее общих подходов к решению стоящих задач, инвариантных к конкретной содержательной стороне задачи и техническим средствам ее реализации.

Для информационных систем эта задача обостряется в связи с развитием научного знания, существенным увеличением его объемов, когда уже в рамках узких, подотраслевых вопросов объемы процессов восприятия нового знания превышают возможности человека, не говоря уже о возможностях использования межотраслевого опыта. При этом является рациональным решение, когда сочетаются наиболее общие подходы к решению проблемы с их конкретной технической реализацией. Возможность рассматривать любую систему, абстрагируясь от ее технической реализации, возможность переноса опыта по разработке и исследованию систем, решающих один круг задач, к системам, предназначенным для решения задач в иной области, говорит об открытости, как самих систем, так и о принципах и подходах к их построению и исследованию, которые будут сформулированы ниже.

## 1.2 Требования, предъявляемые к информационным системам

Сами по себе средства вычислительной техники не могут осуществить преобразование информации, для этого необходимо наличие прикладного информационного и программного обеспечения, реализующего функции информационной или информационно–управляющей системы (ИС). ИС представляет собой совокупность элементов, находящихся в отношениях и связях между собой и образующих определенную целостность, единство, и предназначенных для осуществления целенаправленного процесса преобразования информации.

Внешняя среда

## Система

Ввод информации

### Самообучение и хранение информации

**Вывод информации**

Рис 1. Основные функции системы

К основным функциям ИС можно отнести:

* организация интерфейса обмена между технической и информационной системами, а также между ИС и внешней средой;
* организация работы и распределение ресурсов собственно ИС;
* самообучение системы, адаптация к изменяющимся условиям.

ИС должны удовлетворять следующим требованиям:

* обеспечивать достоверность соответствия описаний объектов в ИС по отношению к их реальному состоянию;
* иметь дружественный интерфейс процесса управления,
* обладать возможностями развития и самообучения системы;
* обеспечивать полноту представления информации в системе и во взаимодействии системы с внешней средой, своевременность и обоснованность в выработке соответствующих решений, мобильность ИС при работе в условиях гетерогенных технических средств, реализующих систему, защиту информации в системе;
* обеспечивать реализуемость заданного алгоритма;
* надежность работы в реальных условиях.

Развитие средств компьютерной техники, расширенное их внедрение во все сферы науки, техники, сферы обслуживания и быта привели к необходимости объединения конкретных вычислительных устройств и реализованных на их основе ИС в единые информационно–вычислительные системы (ИВС) и среды. При этом возникли следующие проблемы:

* разнородность технических средств ВТ с точки зрения организации вычислительного процесса, архитектуры, системы команд, разрядности процессора и шины данных, ресурсных возможностей, частот синхронизации и так далее, потребовала создания физических интерфейсов, обеспечивающих их совместимость;
* разнородность программных сред, реализуемых в конкретных вычислительных устройствах и системах с точки зрения многообразия операционных систем, различия в разрядности, объемах адресуемой памяти, применяемых языках программирования и так далее, привела к созданию программных интерфейсов между устройствами и системами;
* разнородность реализации одной вычислительной структуры, изготовленной различными производителями, также требовала применения специальных ограничений, либо разработки дополнительных программных и (или) технических средств для интеграции;
* разнородность интерфейсов общения в системе "человек–машина" требовала постоянного переобучения кадров.

Таким образом, необходимость предусмотреть уже на стадии разработки возможность интегрирования разрабатываемого устройства в гомогенные и, особенно, в гетерогенные информационно–вычислительные среды стала актуальной для разработчиков как аппаратных, так и программных средств.

С этой целью, при разработке ИС необходимо соблюдать требование системности, включающее в себя:

1. Систематизацию информационной базы, то есть исключение противоречий и дублирования между отдельными ее частями, обеспечение полного представления информации, согласование времени поиска информации в соответствии со структурой.
2. Организацию и упорядочивание внешних связей ИУС и технических средств автоматизации.
3. Учет условий хранения информации в ИУС.
4. Стандартизацию форм представления информации, форм представления документов, структуры информационной базы, структуры и свойств алгоритмов

Весь процесс разработки можно условно разделить на этапы: анализ системы и разработка ее информационной модели, разработка математической модели (алгоритма), разработка программной модели, разработка документации на ИС.

## 1.3 Компоненты и структура ИС

Рисунок 2 отображает типичную структуру технологического процесса ИС или представление ИС как совокупности функциональных подсистем – сбора, ввода, хранения, поиска, распространения информации.

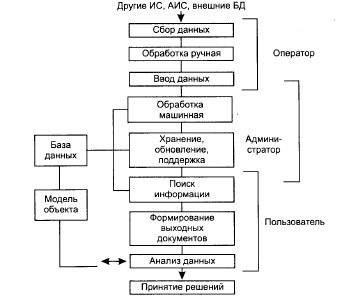


Рис. 2. Основные технологические процессы ИС

Некоторые компоненты данной структуры являются необязательными:

1. Модель объекта может отсутствовать либо отождествляться с базой данных, которая часто интерпретируется как информационная модель предметной области, структурная (для фактографических и табличных) или содержательная (для документальных). В экспертных системах в качестве модели объекта (предметной области) фигурирует база знаний, представляющая собой процедурное развитие понятия БД
2. Модель объекта и БД могут отсутствовать (а соответственно и процессы хранения и поиска данных), если система осуществляет динамическое преобразование информации и формирование выходных документов без сохранения исходной, промежуточной, результирующей информации. Если также отсутствует преобразование информации, то подобный объект не является ИС.
3. Процессы ввода и сбора данных являются необязательными, поскольку вся необходимая и достаточная для функционирования ИС информация может уже находиться в БД и составе модели.

В более общем случае, учитывая специфику организации, управления и технологии выполнения каждой из указанных функций в ИС целесообразно выделять три самостоятельных функциональных подсистемы:

**Подсистема отбора информации**. Информационная система может обрабатывать/перерабатывать только ту информацию, которая в нее введена. Качество работы ИС определяется не только ее способностью находить и перерабатывать нужную информацию в собственном массиве и выдавать ее пользователю, но и способностью отбирать релевантную информацию из внешней среды. Такой отбор осуществляет подсистема отбора информации, которая накапливает данные об информационных потребностях пользователей ИС (внутренних и внешних), анализирует и упорядочивает эти данные, образуя информационный профиль ИС. Аналогично на основании данных о потоках информационной среды формируется описание входных потоков информации.

При заданном критерии качества функционирования ИС и соответствующей системы ограничений в процессе управления ИС решается задача оптимизации комплектования информационного массива ИС, которая определяет алгоритм (или оператор) отбора информации. Указанный оператор осуществляет преобразование входных потоков в информационный массив ИС. К сожалению, многие действующие ИС слабо придерживаются описанной процедуры отбора документов. Отбор информации, как правило, носит слабоуправляемый характер, базируется на интуиции специалистов. Это является следствием сложности и слабой структурируемости как собственно процессов отбора, так и управления этими процессами.

Функции именно этой подсистемы ИС практически не поддаются автоматизации. Исключение составляют только ИС информационного обеспечения управления технологическими процессами и техническими системами.

**Подсистема ввода, обработки/переработки и хранения информации** осуществляет преобразования входной информации и запросов, организацию их хранения и переработки с целью удовлетворения информационных потребностей абонентов ИС.

Реализация функций данной подсистемы предполагает наличие аппарата описания информации (ИПЯ, систем кодирования, ЯОД и т. д.), организации и ведения информации (логическая и физическая организация, процедуры ведения и защиты информации и т. д.), аппарата обработки и переработки информации (алгоритмы, модели и т. д.).

Все три указанные составляющие определяются двумя параметрами ИС: характером обрабатываемой информации и функциями ИС.

Документальные ИС для описания информации используют ИПЯ и систему индексирования, методология построения и использования которых существенно отличается от методологии и принципов использования ЯОД, обеспечивающих описание данных в фактографических ИС. Логическая организация данных фактографических ИС имеет мало общего с организацией информации в документальных ИС. Наконец, различны и аппараты обработки и переработки документальной и фактографической информации. Если в фактографических ИС преимущественно используются математические алгоритмы, то в документальных – эвристические процедуры, требующие затрат интеллектуальной энергии.

**Подсистема подготовки и выдачи информации** непосредственно реализует удовлетворение информационных потребностей пользователей ИС (внутренних и внешних). Для выполнения этой задачи подсистема проводит изучение и анализ информационных потребностей, определяет формы и методы их удовлетворения, оптимальный состав и структуру выходных информационных продуктов, организует сам процесс информационного обеспечения и сопровождения. Выполнение указанных функций требует наличия аппарата описания и анализа информационных потребностей и их выражения на языке ИС (в том числе ЯОД, ИПЯ, языке индексирования и т. д.), а также аппарата непосредственно информационного обеспечения (процедуры поиска и выдачи информации, языки манипулирования данными и т. д.).

Все эти и многие другие составляющие рассматриваемой подсистемы, выполняя одинаковые функции в ИС разных типов, тем не менее существенно отличаются между собой. Особенно заметно это различие при сравнении документальных и фактографических ИС.

Из предыдущего рассмотрения следует, что многие функции различных подсистем ИС дублируются или пересекаются, что является предметом оптимизации при проектировании ИС. Автоматизация ИС в связи с этим сопровождается перераспределением элементов ИС.

Автоматизация предполагает формализованное представление (структуризацию) как функций ИС, так и самой обрабатываемой в ИС информации, которое и позволяет осуществлять ввод, обработку/переработку, хранение и поиск информации с использованием ЭВМ.

Однако любая формализация характеризуется тем или иным уровнем адекватности создаваемого образа реальной действительности (модели) самой действительности. Причем, адекватность модели реальной действительности определяется как свойствами самой действительности, так и возможностями используемого аппарата ее формализованного представления.

С этой точки зрения "уровень автоматизации" ИС тесно связан со "степенью структурируемости" как самой информации, являющейся предметом обработки, хранения и т. д., в ИС, так и самих функций ИС (обработки, хранения и т. д.).

В соответствии с уровнем современных знаний в области формализованного представления информации можно различать информацию трех уровней структурируемости:

1. Жесткоструктурируемая информация – информация, формализованное представление которой современными средствами ее структурирования (в частности, языками описания данных) не приводит к потере адекватности создаваемого образа информации (модели) самой исходной информации. Жесткоструктурируемую информацию будем в дальнейшем называть данными.
2. Слабоструктурируемая информация– информация, формализованное представление которой современными средствами описания информации (в частности, ИПЯ) приводит к значительным потерям адекватности модели информации самой исходной информации. Обработка и поиск такой информации предполагает специальные меры по оценке степени неадекватности модели информации. (В АИПС этой цели служат меры смысловыразительной способности (семантической силы) ИПЯ).
3. Неструктурируемая информация – информация, для которой в настоящее время не существует средств ее формализованного представления с приемлемым на практике уровнем адекватности. Средства представления такой информации должны обладать высокими смысловыразительными способностями. Разработка таких средств в настоящее время идет по линии создания языков описания знаний и ИПЯ с высокой семантической силой.

Приведенная классификация информации по степени ее структурируемости достаточно условна. Однако сама идея учета структурируемости информации оказывается полезной при анализе сущности современных АИС.

Если с этих позиций рассмотреть функции подсистем ИС, то нетрудно видеть, что большинство жесткоструктурируемых функций сосредоточено в подсистеме ввода, обработки/переработки и хранение информации. Две другие подсистемы связаны с реализацией, в основном, слабоструктурируемых и неструктурируемых функций.

Легкость автоматизации функций второй подсистемы ИС на основе использования электронно–вычислительной и телекоммуникационной техники ввода, обработки, хранения и передачи информации привела к неоправданно быстрому и вседовлеющему развитию этих составляющих ИС в ущерб развитию двух других (не менее, а может быть и более важных) ее составляющих. В большинстве современных АИС эти две подсистемы настолько неразвиты, что по сути дела это уже не АИС, а организационно–обособленные подсистемы ввода, обработки, хранения и поиска информации. В дальнейшем будет показано, что говоря об этих системах, целесообразно называть их не АИС, а банки данных или АИПС.

## 1.4 Понятие информационно–образовательной среды

Эффективность любого вида обучения зависит от ряда составляющих:

–материально–технической базы;

–обучающих технологий, используемых при организации и управлении познавательной деятельностью;

–эффективности разработанных методических материалов и способов их доставки.

Другими словами, успешность и качество современного обучения в большей мере зависят от эффективной организации, педагогических условий, качества используемых материалов, педагогического мастерства, подготовленности педагогов к работе в условиях лавинообразного нарастания потока информации, возможности овладения современными методами поиска, отбора и использования информации.

Управление обучением не должно сводится к простому подбору и подготовке обучающего материала. Необходимо создание единой среды обучения, которая является возможностью реализации равноправия всех обучающихся в доступе ко всей информации и средствам обучения, представленным в данной среде, и, в то же время, сохранению индивидуально-независимой траектории обучения в соответствии с личностными запросами индивида.

Современные информационные технологии предоставляют практически неограниченные возможности в размещении, хранении, обработке и доставке информации на любые расстояния, любого объема и содержания. В этих условиях на первый план при организации системы обучения выходит содержательность обучающего материала, при условии нормального технического оснащения организации обучения. Имеется в виду не только отбор материала по содержанию, но и структурная организация учебного материала, включение его в процесс обучения. Требуется создание не просто автоматизированных обучающих программ, но создание именно интерактивных информационных сред общения с обучающимися, созданных на основе значительно расширенных дидактических возможностей современных компьютерных средств обучения и средств телекоммуникационной связи.

Распространение использования сети Интернет и локальных сетей в вузах и других учебных заведений настоятельно требует разработки и применения новых программных продуктов для управления, точнее, направления познавательной деятельностью. Такими программными продуктами могут стать автоматизированные средства обучения, к которым относятся:

–информационно-обучающие сайты;

–информационно-предметные среды обучения;

–электронные гиперссылочные и мультимедийные учебные материалы;

–программы управления поисковой и познавательной деятельностью обучающегося;

–контрольно-обучающие программы;

–тренажеры;

–профильные автоматизированные рабочие места;

–компьютерные лабораторные комплексы.

Каждый отдельный программный продукт из вышеперечисленного списка сам по себе, несомненно, несет определенную пользу обучающемуся. Но эффективность и качество обучения многократно увеличится, если комплексно объединить данные программные продукты в единой оболочке – автоматизированной обучающей среде (АОС) или информационной системой обучения (ИСО).

Эти системы представляют собой комплексы научно-методической, учебной и организационной поддержки процесса обучения, проводимого на базе компьютерных, или, как их также называют, информационных технологий. С позиций современной дидактики введение информационной среды и программного обеспечения внесло огромное количество новых возможностей во все области процесса обучения. Компьютерные технологии представляют собой принципиально новые средства обучения. За счет своего быстродействия и больших резервов памяти они позволяют реализовывать различные варианты сред для программированного и проблемного обучения, строить различные варианты диалоговых режимов обучения, когда так или иначе ответ учащегося реально влияет на ход дальнейшего обучения.

Вследствие этого современный педагог с неизбежностью должен осваивать новые образовательные подходы, опирающиеся на средства и методы индивидуального компьютерного обучения. В общем случае педагог получает доступ к компьютерным средствам, информационной среде и программным продуктам, предназначенным для обеспечения преподавательской деятельности. Все эти средства образуют комплексы автоматизированных обучающих систем.

В рамках информационных систем обучения на сегодняшний день решается ряд задач обучения. В первую группу можно отнести задачи поверки уровня знаний, умений и навыков учащихся до и после обучения, их индивидуальных способностей, склонностей и мотиваций. Для таких проверок обычно используют соответствующие системы (батареи) психологических тестов и экзаменационных вопросов. К этой же группе относятся задачи проверки показателей работоспособности учащихся, что осуществляется путем регистрации таких психофизиологических показателей, как скорость реакции, уровень внимания и т.д.

Вторая группа задач связана с регистрацией и статическим анализом показателей усвоения учебного материала: заведение индивидуальных разделов для каждого учащегося, определение времени решения задач, определение общего числа ошибок и т.д. К этой же группе логично отнести решение задач управления учебной деятельностью. Например, задач по изменению темпа предъявления учебного материала или порядка предъявления учащемуся новых блоков учебной информации в зависимости от времени решения, типа и числа ошибок. Таким образом, эта группа задач направлена на поддержку и реализацию основных элементов программированного обучения.

Третья группа задач ИСО связана с решением задач подготовки и предъявления учебного материала адаптации материала по уровням сложности, подготовки динамических иллюстраций, контрольных заданий, лабораторных работ самостоятельных работ учащихся. В качестве примера уровня таких занятий можно указать на возможности использования различных инструментов информационных технологий. Другими словами, использования программных продуктов, дающих возможность формирования различных сложных лабораторных и др. практических работ. Например, таких, как сборка «виртуального» осциллографа с последующей демонстрацией его возможностей по регистрации усилению или синхронизации различных сигналов. Аналогичные примеры из области химии могут касаться моделирования взаимодействия сложных молекул, поведения растворов или газов при изменение условий эксперимента.

Техническое обеспечение автоматизированных обучающих систем основано на локальных компьютерных сетях, включающих автоматизированные рабочие места (АРМ) учащихся, преподавателя и линии связи между ними. Рабочее место учащегося, кроме монитора (дисплея) и клавиатуры, может содержать принтер, такие элементы мультимедиа, как динамики, синтезаторы звуков, текстовые и графические редакторы. Цель этих всех технических и программных средств состоит в обеспечении учащихся средствами решения, справочным материалом и средствами регистрации ответов.

Таким образом, автоматизированная обучающая среда должна объединить в себе три основных компонента, обеспечивающие современные технологии обучения: информационное, методическое и программное обеспечение образовательного процесса. Такой комплексный подход позволит обеспечить обучающегося педагогическими условиями для успешного освоения учебного материала, свободным графиком изучения, а также индивидуальным маршрутом обучения за счет использования различной глубины представленного материала. Кроме этого, автоматизированная среда должна обеспечить управление познавательной деятельностью, предоставляя в зависимости от успехов в работе обучающегося рекомендации и доступ к различным уровням информации.

Таким образом, программному обеспечению управления познавательной деятельностью необходимо предъявлять следующие требования:

–при разработке программного продукта необходимо обеспечить неограниченное число одновременных подключений;

–необходимо обеспечить возможности разноуровневого интерактивного обучения и контроля;

–для обеспечения доступа всех образовательных единиц, повышения эффективности учебного процесса необходимо размещение разработанных программных сред обучения и контроля на общем сервере, с ранжированием прав доступа для использования в учебном процессе и подготовки предметного материала;

–требуется разработка инструментальных средств создания и оформления предметного материала с использованием гиперссылок, аудио и видео вставок, мультимедиа технологий;

–построение программ с использованием технологии клиент–сервер, протокола HTTP, средств HTML и CGI позволит применять в качестве клиентского места обычный WWW–браузер;

–хранение результатов обучения и тестирования в базе данных;

–необходимо обеспечить безопасность программ и предметного материала.

Безусловно, основное внимание при подготовке различных автоматизированных обучающих сред должно быть уделено разнообразию предоставляемых интерактивных направляющих воздействий со стороны управляющей программы, позволяющей обучающемуся моделировать в интерактивном режиме собственную траекторию и индивидуальную среду обучения. Единая информационно–образовательная среда – это то огромное хранилище разнообразного материала, в котором любой обучающийся должен иметь возможность индивидуальной работы, выбора и сохранения того материала, тех средств обучения, которые лично ему необходимы.

При оформлении предметного материала автоматизированных сред обучения желательно следовать одному разработанному стандарту компоновки и оформления как для печатного учебного материала, так и для всех видов электронных изданий.

Электронные интерактивные учебно–методические материалы (пособия, учебники, другое) – это обычно набор взаимосвязанных веб–документов, объединенных в единую логическую структуру и включающих в себя элементы текста, статических и динамических изображений, аудио и видеоматериалов, элементы меню и навигации, а также средства тестирования и самоконтроля.

Разработку предметного материала желательно начинать с разработки структуры всего курса. Предметный материал для обучения должен иметь не только теорию, но, желательно, полный набор всего дидактического материала (схемы, рисунки, таблицы, графики, упражнения и пояснения к их выполнению, вопросы текущего контроля и правильные ответы), а также сопровождение обучающего материала контролирующей программой, которая должна быть достаточно простой и небольшой по коду. Кроме того, необходимо снабдить материал вопросами итогового контроля. Важным моментом при подготовке к оформлению учебного материала в виде электронного гиперссылочного пособия/учебника является разработка схемы гиперссылок. Это момент в разработке методических материалов, сегодня очень важен, позволяющий сделать обучающий материал удобным для работы.

Современные технологии обучения и возможности современных программных средств требуют нового подхода к составлению и оформлению учебных пособий. Прежде всего, возникает вопрос унификации структуры учебника для облегчения процесса подготовки электронной версии. Это позволяет облегчить процесс обучения, прежде всего, для самого обучающегося.

Основой создания и развития единой информационно-образовательной среды, совершенствования информационных сред различных образовательных учреждений и направлений для повышения качества подготовки специалистов, научных исследований, межличностного и интеркультурного общения является развитие сетевых информационных, мультимедийных и компьютерных технологий обучения.

Таким образом, единая информационно–образовательная среда университета позволит существенно повысить качественный уровень деятельности системы образования, обеспечит создание условий профессионального и научно–исследовательского роста преподавателей, создаст благоприятные условия для расширения сотрудничества ведущих ученых и преподавателей университета с педагогическими коллективами образовательных учреждений города и области различного профиля, активизирует научно–творческую деятельность студентов и учащихся.

## 1.5 Информационные системы в образовании

В отечественной системе образования первые информационные системы создавались еще в 60-е годы.

Можно выделить следующие уровни управленческой деятельности с использованием ЭВМ в системе образования:

1. управление обучением и развитием отдельного учащегося;
2. управление учебным процессом в рамках одного учебного заведения;
3. управление работой группы родственных учебных заведений;
4. управление учебными заведениями по территориальному принципу;
5. управление системой образования страны.

На первом уровне задачи управления совпадают в значительной мере с задачами обучения с помощью компьютеров.

На втором уровне реальные успехи достигнуты прежде всего в вузах. С одной стороны, государственное высшее учебное заведение достаточно велико по контингенту учащихся и преподавателей и имеет достаточно большую материальную базу для того, чтобы использование компьютеров в управлении было экономически оправдано, с другой – в вузах, особенно технических, наличествуют достаточно профессионально подготовленные кадры для решения проблемы информатизации управления. При этом преследуются следующие цели:

* повышение качества подготовки специалистов за счет совершенствования управления со стороны ректората, деканатов, кафедр;
* повышение качества учебной, учебно–методической, научно–исследовательской деятельности на основе оперативной информации;
* повышение эффективности в разработке учебных планов и программ, составление расписания занятий, других видов аудиторной и внеаудиторной работы.

Традиционными программными подсистемами информационной системы управления вузом являются Абитуриент, Кадры, Учебные планы и программы, Зарплата, Стипендии, Текущая успеваемость, Нагрузки преподавателей, Сессия и другие.

Вместе с тем, эти подсистемы редко образуют единую информационную систему. Неразвитость информационной среды, отсутствие в большинстве вузов полноохватной локальной сети, материальные трудности, неподготовленность управленческого персонала и другие факторы препятствуют созданию систем типа «клиент–сервер» с единым администрированием, гарантией отсутствия противоречивых данных, защитой целостности и конфиденциальности данных.

## 1.6 Автоматизированные обучающие системы

Автоматизированные обучающие системы представляют собой комплексы научно–методической, учебной и организационной поддержки процесса обучения, проводимого на базе компьютерных, или информационных технологий. С позиций современной дидактики введение информационной среды и программного обеспечения внесло огромное количество новых возможностей во все области процесса обучения. Компьютерные технологии представляют собой принципиально новые средства обучения. За счет своего быстродействия и больших резервов памяти они позволяют реализовывать различные варианты сред для программированного и проблемного обучения, строить различные варианты диалоговых режимов обучения, когда так или иначе ответ учащегося реально влияет на ход дальнейшего обучения.

Вследствие этого современный педагог с неизбежностью должен осваивать новые образовательные подходы, опирающиеся на средства и методы индивидуального компьютерного обучения. В общем случае педагог получает доступ к компьютерным средствам, информационной среде и программным продуктам, предназначенным для обеспечения преподавательской деятельности. Все эти средства образуют комплексы автоматизированных обучающих систем.

В рамках автоматизированных информационных обучающих систем на сегодняшний день решается ряд задач обучения. В первую группу можно отнести задачи проверки уровня знаний, умений и навыков учащихся до и после обучения, их индивидуальных способностей, склонностей и мотиваций. Для таких проверок обычно используют соответствующие системы (батареи) психологических тестов и экзаменационных вопросов. К этой же группе относятся задачи проверки показателей работоспособности учащихся, что осуществляется путем регистрации таких психофизиологических показателей, как скорость реакции, уровень внимания и т.д.

Вторая группа задач связана с регистрацией и статическим анализом показателей усвоения учебного материала: заведение индивидуальных разделов для каждого учащегося, определение времени решения задач, определение общего числа ошибок и т.д. К этой же группе логично отнести решение задач управления учебной деятельностью. Например, задач по изменению темпа предъявления учебного материала или порядка предъявления учащемуся новых блоков учебной информации в зависимости от времени решения, типа и числа ошибок. Таким образом, эта группа задач направлена на поддержку и реализацию основных элементов программированного обучения.

Третья группа задач АОС связана с решением задач подготовки и предъявления учебного материала адаптации материала по уровням сложности, подготовки динамических иллюстраций, контрольных заданий, лабораторных работ самостоятельных работ учащихся. В качестве примера уровня таких занятий можно указать на возможности использования различных инструментов информационных технологий. Другими словами, использования программных продуктов, дающих возможность формирования различных сложных лабораторных и др. практических работ. Например, таких, как сборка «виртуального» осциллографа с последующей демонстрацией его возможностей по регистрации усилению или синхронизации различных сигналов. Аналогичные примеры из области химии могут касаться моделирования взаимодействия сложных молекул, поведения растворов или газов при изменение условий эксперимента.

Техническое обеспечение автоматизированных обучающих систем основано на локальных компьютерных сетях, включающих автоматизированные рабочие места (АРМ) учащихся, преподавателя и линии связи между ними. Рабочее место учащегося, кроме монитора (дисплея) и клавиатуры, может содержать принтер, такие элементы мультимедиа, как динамики, синтезаторы звуков, текстовые и графические редакторы. Цель этих всех технических и программных средств состоит в обеспечении учащихся средствами решения, справочным материалом и средствами регистрации ответов.

## 1.7 Модели обучения автоматизированных обучающих систем

В настоящее время разработано большое число электронных учебных материалов, в качестве которых выступают электронные учебники, электронные учебные пособия, автоматизированные обучающие системы и т.п. Существующие электронные учебные материалы решают те или иные задачи обучения с большей или меньшей эффективностью, которая определяется, прежде всего, степенью управляемости обучаемым в процессе обучения. В условиях нарастающего интереса, к созданию различных вариантов электронно–методических материалов возникает необходимость в классификации этих материалов с целью оценки их различия и определения области применения. Уже существует ряд классификаций обучающих систем по различным их свойствам. Однако нет классификации, отражающей управляемость обучаемого системой, что при расширяющемся использовании электронных учебных материалов, является важным на данный момент.

Предлагаемая ниже классификация ранжирует различные реализации электронных учебно–методических материалов по распределению ролей между обучаемым и системой, реализуемых ими в процессе обучения.

**1. Технизация процесса обучения. Технологизация педагогических методов**

Систематическими исследованиями проблем обучения первыми занялись психологи через изучение психофизиологических особенностей обучаемых. В психологии обучение понимается так же как в педагогике – усвоение обучаемым определенной системы знаний, умений и навыков. При этом, с точки зрения психологии, важную роль в обучении играет память, т.е. такие важнейшие психические процессы, как запоминание и забывание, характеризующие усвоение знаний. В результате экспериментов психологов, были получены различные коэффициенты и зависимости, на основе которых были созданы первые модели обучения (так, например, модель Эббингауза, детерминированная формула Терстоуна). Позднее данные модели были переведены в вероятностную форму. Данные модели используются разработчиками систем на последующих этапах развития моделей обучения.

Идея автоматизации учебного процесса на данном этапе сводилась к использованию, главным образом, различных технических средств обучения (ТСО), дополняющих учебный процесс. Все разработки были направлены на создание обучающей технической среды. При этом технологичность процесса обучения определялась объемом применения ТСО как дополнительного средства обучения. Постепенно исследователи переходили к идее применения ТСО не как дополнения учебного процесса, а как устройства, берущего на себя некоторые функции учителя. Т. к. ТСО не обладали свойством управления учебным процессом, реализация с их помощью функций учителя, т.е. замена учителя техническим средством для управления или сопровождения хотя бы части учебного процесса было невозможно. В результате исследователи пришли к необходимости осмыслить сам учебный процесс, формализовать его и описать как технологический процесс.

На данном этапе учебный процесс стал объектом исследований. Был исследован сам учебный процесс, а так же различные способы его организации, основанные на различных педагогических методах. При этом основной принцип построения учебного процесса заключался в системе последовательных, четко описанных действий, выполнение которых ведет к заранее запланированной цели. Первым результатом этих исследований и одновременно основой последующих моделей обучения в начале 60–х годов XX века стала модель программируемого обучения, представленная во множестве изданий. Сутью данной модели является адаптация учебного процесса под четко заданные цели. Цели представлены некоторым эталонным результатом, например, заданные правильные ответы. После сравнения результата с эталоном ставится оценка, которая является единственной характеристикой обучаемого. В зависимости от оценки выбирается следующий этап учебного процесса, при неудовлетворительной оценке могут быть выбраны и альтернативные способы изложения материала. Такие модели могут быть реализованы как линейными так и разветвленными схемами обучения. При использовании только одной характеристики обучаемого идея о построении его модели не рассматривается, объектом управления остается сам учебный процесс, уже внутри которого находится объект – обучаемый.

**2. Реализация моделей обучения на основе метода пакета прикладных программ**

Данный этап охарактеризован реализацией идей программированного обучения в электронных учебно–методических материалах (например, АОС) на основе метода пакета прикладных программ. Основным принципом данного метода является разделение библиотеки стандартных программ и программ, управляющих ресурсами машины и библиотекой. Для взаимодействия пользователя с системой используется диалоговый компонент со специальным входным языком, позволяющим давать четкие команды вызова обучающей системе. Схема процесса обучения в АОС следующая: обучаемому предъявляется порция обучающей информации (ОИ), дается проверочное задание, осуществляется проверка правильности ответов и определяется следующая порция ОИ. При линейной схеме обучения план обучения задается разработчиками заранее с расчетом на среднего обучаемого и не корректируется в процессе обучения. Несколько позднее, реализовали разветвленные (более сложные) схемы обучения, в которых обучаемые были разделены на группы и план обучения задавался для каждой группы отдельно с расчетом на среднего обучаемого этой группы. Характеристикой обучаемого является номер его группы или оценка. Отнесение обучаемого к группе или оценка определяется только по его ответам. Метод ППП позволяет реализовать данные схемы: входной язык диалогового компонента достаточен для принятия ответов обучаемого, а программа, управляющая библиотекой, способна вызвать программы расчета оценок обучаемого и выбрать следующий этап учебного процесса.

АОС с разветвленными схемами обучения позволяли задавать индивидуально план обучения для каждой группы обучаемых, однако такие планы обучения все равно рассчитаны на среднего обучаемого, но уже для группы. Исследователи пришли к пониманию что для эффективного управления таким сложным объектом, как обучаемый, для которого невозможно заранее создать точной и полной траектории обучения, необходимо индивидуализировать процесс обучения для каждого обучаемого, а для этого системе необходимы знания об обучаемом, изучаемой им среде и возможностях управления учебным процессом.

**3. Реализация моделей обучения методом экспертных систем**

Для получения большей эффективности управления обучаемым исследователи обратились к более глубокому изучению понятия «адаптации». Адаптация, как процесс приспособления к объекту управления имеет несколько иерархических уровней, соответствующие различным этапам управления обучаемым:

* Параметрическая адаптация реализуется путем подстройки значений параметров модели обучаемого под его текущее состояние.
* Структурная адаптация реализуется путем перехода от одной структуры к другой, структуры должны быть родственными между собой, но отличаться набором параметров и связей между ними. Например, при разветвленной схеме обучения для каждого типа обучаемого определена соответствующая модель, отличающаяся структурой с моделями других типов обучаемых. Такая структурная адаптация называется адаптацией по статической структуре. Другим способом реализации структурной адаптации является адаптация по функциональной структуре, что предполагает изменение функций управления программой обучения, т.е. изменение схемы взаимодействия системы и обучаемого. Функциональная структурная адаптация и адаптация по статической структуре так же могут быть реализованы системами «без памяти» и системами «с памятью».
* Адаптация объекта управления. Всякий объект представлен в системе ограниченной моделью, все не попавшие в модель параметры и структуры считаются внешней средой. Данная адаптация реализуется путем расширения модели за счет добавления в модель новых параметров или структур из внешней среды.
* Адаптация целей реализуется за счет выбора нового множества целей из множества возможных целей, определенных априори в системе. Все предыдущие уровни адаптации направлены на достижение целей, поставленных перед системой.

Для реализации всех рассмотренных уровней адаптации в моделях с разветвленной схемой обучения не хватало «знаний» об обучаемом. Это привело к созданию моделей обучения, в которых для управления процессом обучения используются модели об обучаемом наряду с наличием в системе экспертных знаний о предмете изучения и педагогических методах. Реализацией данного подхода стало появление в 1982 году новых структур обучающих систем на базе метода экспертных систем (ЭС).

Главным отличием данной модели обучения от предыдущих, является возможность не закладывать априори последовательность шагов обучения, т. к. она строится самой системой в процессе ее функционирования, что и позволяет строить для каждого обучаемого индивидуальный план обучения.

Данные обучающие системы способны выполнять параметрическую и структурную адаптации. Однако, в случае возникновения задачи, для решения которой у системы не достаточно знаний, задача остается не решенной. Это говорит о не достаточности параметров в структуре моделей обучаемого или несоответствии цели, преследуемой системой, целям объекта обучения. В данных системах экспертные знания о предмете и методах изучения должны быть полными, проектироваться априори и в процессе обучения не изменяться. Кроме того, работа системы направлена на достижение одной фиксированной, априори определенной цели обучения. Это делает невозможным реализацию адаптации целей обучения и тем более адаптацию объекта обучения.

**4. Мультиагентный подход к реализации моделей обучения**

В рамках мультиагентного подхода рассматривается возможность реализации адаптации всех уровней, что позволит обеспечить управление объектом – обучаемым на всех этапах процесса обучения.

Основа этого подхода – построение системы как совокупности агентов (агенты пользователя, агенты преподавателя, агенты лекций и даже агенты отдельных объектов знания: определений понятий и правил, задач, методов, результатов, лабораторных работ, комментариев и т.д.). Каждый из агентов имеет семантическое описание своего поля деятельности (свою структуру, свои знания), и соответствует экспертной системе с традиционной структурой . Агент обладает всеми свойствами экспертных систем, а так же памятью своей деятельности. Основная идея применения агентов заключается в том, что каждый агент имеет собственные ресурсы для достижения собственных целей, взаимодействия с другими агентами и разрешения конфликтов с целями других агентов для достижения общей цели. Это позволяет свободно выбирать те цели, которые преследуются на данный момент объектом управления, и соответственно целям выбирать тот эталон (представленный соответствующим агентом), соответствие которому достигается моделью обучаемого на данный момент.

Движущей силой систем, основанных на мультиагентном подходе, является способность агентов вести переговоры. При этом их коммуникация основана на семантических сообщениях (самого высокого уровня), а не на заранее предопределенных сообщениях низшего порядка. Переговоры необходимы для одновременного выполнения функций агентов, когда разные агенты, возможно, имеют разные взаимоисключающие цели и намерения, разные возможности в своих виртуальных мирах, обладают различной информацией. Вопросы взаимодействия агентов разной архитектуры решены применением соответствующего языка коммуникации агентов (ACL) и языка обмена информацией, которые дают возможность агентам эффективно понимать друг друга несмотря на разницу в подходах их построения и функционирования.

Мультиагентная система реализует распределенное управление, которое может быть как централизованным, так и децентрализованным.

Централизованное управление выполняется центральным устройством управления, который формирует коллективы агентов и распределяет все возникающие задачи между агентами коллектива.

При децентрализованном управлении известны разные варианты реализации систем, одним из них является применение «контрактной системы» управления. При реализации данного подхода, вершинами сети агентов является множество независимых управляющих агентов (исполнителей), которые обладают информацией о том, какие задачи они способны решать, какие средства использовать, с какими агентами и как взаимодействовать при решении задачи. При возникновении конкретной задачи агент происходят переговоры между агентами и выясняется какой агент какую часть задачи может решить. С помощью такого процесса происходит распределение решения задачи. Все агенты независимы, т.е. исходное состояние графа до начала решения задачи представляет изолированные между собой вершины. Все связи устанавливаются только в процессе функционирования системы при решении задач. Использованию данного подхода препятствует отсутствие эффективного глобального управления работой такой системы, несмотря на то, что такой подход обладает гибкостью и модифицируемостью обучающей системы.

Таким образом, для каждой конкретной задачи обучения составляется определенный коллектив агентов, что говорит о смене структуры и целей решающей системы в зависимости от поставленной задачи. Формирование коллективов агентов для решения задач обучения позволяет реализовать любой уровень адаптации, т.к. эта процедура предполагает формирование каждый раз структуры системы, ее представления об объекте управления, т.е. обучаемом и целей обучающей системы, адаптируемые под цели, преследуемые на данный момент объектом управления.

## 1.8 Обзор информационных систем обучения

Систематические исследования в области компьютерной поддержки процесса обучения имеют более чем 30–летнюю историю. За этот период в США, Канаде, Англии, Франции, Японии, России и ряде других стран было разработано большое количество компьютерных систем учебного назначения, ориентированных на различные типы ЭВМ

Сферы применения компьютерных средств поддержки процесса обучения гораздо шире, чем только учебные заведения. Это крупные промышленные предприятия, военные и гражданские организации, ведущие самостоятельную подготовку и переподготовку кадров. Кроме того, в цивилизованных странах становится уже стандартом снабжать новые сложные машины и технологии компьютерными обучающими системами, облегчающими и ускоряющими процесс их освоения и внедрения. За рубежом разработку "мягкого" компьютерного продукта учебного назначения (методических и программно–информационных средств) считают весьма дорогостоящим делом в силу его высокой наукоемкости и необходимости совместной работы высококвалифицированных специалистов: психологов, преподавателей-предметников, компьютерных дизайнеров. Несмотря на это, многие зарубежные крупные фирмы финансируют проекты создания компьютерных учебных систем в учебных заведениях и ведут собственные разработки в этой области.

В методологическом плане разработка и использование компьютерных средств поддержки обучения, в первую очередь – "мягкого" продукта, с самого начала развивались по двум направлениям, слабо связанным между собой. Первое направление опирается в своей основе на идеи программированного обучения. В его рамках разрабатываются и эксплуатируются автоматизированные обучающие системы (АОС) по различным учебным дисциплинам. Ядром АОС являются так называемые авторские системы, позволяющие преподавателю–разработчику вводить свой учебный материал в базу данных и программировать с помощью специальных авторских языков или других средств алгоритмы его изучения. Характерными представителями АОС, построенных на алгоритмах программированного обучения, длительное время являлись: за рубежом система PLATO, в нашей стране семейство АОС ВУЗ. С начала 90–х годов в России и странах СНГ распространяются инструментальные среды для создания компьютерных курсов на ПЭВМ типа IBM PC зарубежного (Private Tutor, LinkWay, Costoc) и отечественного производства: АДОНИС, АСОК, УРОК и др.

Второе направление компьютеризации обучения является как бы вторичным приложением "мягкого" продукта компьютеризации различных отраслей человеческой деятельности (науки, техники, экономики и др.). Это отдельные программы, пакеты программ, элементы автоматизированных систем (АСУ, САПР, АСНИ, АСУП и др.), предназначенные для автоматизации трудоемких расчетов, оптимизации, исследования свойств объектов и процессов на математических моделях и т.п. Применение таких программных систем в учебном процессе носит более массовый характер, чем использование универсальных АОС, как в нашей стране, так и за рубежом, но, в силу своей разобщенности в содержательном плане и отсутствия единой дидактической платформы, менее известно, систематизировано и обобщено в научно–методической литературе. Среди многочисленных работ в нашей стране по адаптации отраслевых программных разработок для целей обучения определенной системностью и попытками дидактических и технических обобщений выделяются работы по созданию учебно–исследовательских САПР и АСНИ.

С начала 80-х годов интенсивно развивается новое направление в компьютеризации обучения – интеллектуальные обучающие системы (ИОС), основанные на работах в области искусственного интеллекта. Существенной частью ИОС являются модели обучаемого, процесса обучения, предметной области, на основе которых для каждого обучаемого может строиться рациональная стратегия обучения. Базы знаний ИОС могут содержать наряду с формализованными знаниями экспертные знания в предметных областях и в сфере обучения.

"Персональная революция" 80-х гг. принесла в сферу обучения не только новые технические, но и дидактические возможности. Это доступность ПЭВМ, простота диалогового общения и, конечно же, графика. Применение графических иллюстраций в учебных компьютерных системах позволяет не только увеличить скорость передачи информации обучаемому и повысить уровень ее понимания, но и способствует развитию таких важных для специалиста любой отрасли качеств, как интуиция, профессиональное "чутье", образное мышление. А на рынке компьютерных технологий появляются еще более перспективные для целей профессиональной подготовки технические и программные новинки. Это оптические внешние запоминающие устройства на компакт-дисках CD-ROM (Compact Disk Read Only Memory) с большими объемами памяти (сотни мегабайт), инструментальные программные средства гипертекста, мульти – и гипермедиа, системы "виртуальной реальности".

Компьютер, снабженный техническими средствами мультимедиа, позволяет широко использовать дидактические возможности графики и звука. С помощью систем гипертекста можно создавать перекрестные ссылки в массивах текстовой информации, что облегчает поиск нужной информации по ключевым словам, выделенным в тексте. Системы гипермедиа позволяют связать друг с другом не только фрагменты текста, но и графику, оцифрованную речь, звукозаписи, фотографии, мультфильмы, видеоклипы и т.п.

Использование таких систем позволяет создавать и широко тиражировать на лазерных компакт-дисках "электронные" руководства, справочники, книги, энциклопедии.

Развитие информационных телекоммуникационных сетей дает новый импульс системам дистанционного обучения, обеспечивает доступ к гигантским объемам информации, хранящимся в различных уголках нашей планеты.

Новые аппаратные и программные средства, наращивающие возможности компьютера, переход в разряд анахронизма понимания его роли как вычислителя постепенно привели к вытеснению термина "компьютерные технологии" термином "информационные технологии". Под этим термином понимают процессы накопления, обработки, представления и использования информации с помощью электронных средств. Так, суть информатизации образования определяют как создание условий учащимся для свободного доступа к большим объемам активной информации в базах данных, базах знаний, электронных архивах, справочниках, энциклопедиях.

Следуя этой терминологии, можно определить информационные технологии обучения (ИТО) как совокупность электронных средств и способов их функционирования, используемых для реализации обучающей деятельности. В состав электронных средств входят аппаратные, программные и информационные компоненты, способы применения которых указываются в методическом обеспечении ИТО.

Впечатляющий прогресс в развитии аппаратных и инструментальных программных средств ИТО предоставляет хорошие технические возможности для реализации различных дидактических идей. Однако, как показывает анализ отечественных и зарубежных компьютерных систем учебного назначения, ряд из них по своим дидактическим характеристикам нельзя назвать даже удовлетворительными. Дело в том, что уровень качества "мягкого" продукта учебного назначения закладывается на этапе его проектирования при подготовке учебного материала для наполнения баз данных АОС и электронных учебников, при создании сценариев учебной работы с компьютерными системами моделирующего типа, при разработке задач и упражнений и т.п.

К сожалению, методические аспекты ИТО отстают от развития технических средств. Да это и неудивительно, поскольку в методическом плане ИТО интегрируют знания таких разнородных наук, как психология, педагогика, математика, кибернетика, информатика. Разработка средств ИТО для поддержки профессионального образования осложняется еще и необходимостью хорошо знать содержание предметной области и учитывать присущую ей специфику обучения. Именно отставание в разработке методологических проблем, "нетехнологичность" имеющихся методик являются одними из основных причин разрыва между потенциальными и реальными возможностями ИТО.

Теперь рассмотрим некоторые примеры информационных систем обучения и попытаемся выяснить наиболее актуальные технологии построения ИСО на сегодняшний день.

Сначала рассмотрим методические аспекты технологии создания "мягкого" продукта учебного назначения, положенные в основу системы Комплексов Автоматизированных ДИдактических Средств (системы КАДИС), разработанной и развиваемой в центре новых информационных технологий при Самарском государственном аэрокосмическом университете (СГАУ).

В комплексе обобщаются опыт и результаты многолетних исследований по компьютерной поддержке инженерной подготовки. Эти исследования были начаты в конце 70-х гг. на кафедре конструкции и проектирования летательных аппаратов СГАУ.

Одна из первых версий инструментальной среды получила название системы автоматизированного проектирования автоматизированных учебных курсов (САПР АУК). В дальнейшем, несмотря на расширение ее функций от разработки АУК до подготовки целостных комплексов, включающих набор АУК, тренажеров, учебных ППП, это название было сохранено.

В состав САПР АУК входят следующие компоненты: учебное пособие, АУК для освоения и закрепления методики проектирования учебных комплексов, программные средства, информационное обеспечение.

Информационное обеспечение САПР АУК включает базы данных двух типов: базы данных с учебным материалом и журнал. Учебный материал содержит для каждого АУК блоки информации, упражнения, словарь терминов и понятий с их синонимами и определениями, условия вызова подключаемых программ (тренажеров, учебных ППП и т.п.). В журнале накапливается статистика по работе учащихся со всеми АУК.

Программные средства САПР АУК реализуют четыре вида интерфейсов: учащихся, преподавателей–пользователей и преподавателей–разработчиков учебных комплексов, администратора САПР АУК. Структурно все программы также можно разделить на четыре основные части: "проигрыватель" учебных комплексов, обеспечивающий работу учащихся и преподавателей-пользователей; инструментальную оболочку, позволяющую преподавателям-разработчикам наполнять базу данных учебных комплексов; набор программных утилит, реализующих некоторые дополнительные функции в работе преподавателей–разработчиков; утилиты администратора САПР АУК.

Томский Государственный университет является разработчиком очень многих интересных систем обучения. В том числе одна из достаточно интересных и простых разработок – Виртуальный университет. Первые версии информационной системы обучения являлись «локальными» и похожими на нашу систему.

На сегодняшний день наиболее востребованными и эффективными информационными системами обучения являются “сетевые” системы управления обучением (LMS) и системы управления содержимым обучения (LCMS).

Вслед за развитием систем управления сайтом (CMS – Content Management System), стали появляться специализированные системы, в частности для управления обучением.

В англоязычной литературе можно встретить следующую аббревиатуру систем управления обучением:

* LMS – Learning Management System (система управления обучением);
* CMS – Course Management System (система управления курсами);
* LCMS – Learning Content Management System (система управления учебным материалом);
* MLE – Managed Learning Environment (оболочка для управления обучением);
* LSS – Learning Support System (система поддержки обучения);
* LP – Learning Platform (образовательная платформа);
* VLE – Virtual Learning Environments (виртуальные среды обучения).

Основным фундаментом электронного обучения обычно являются системы LMS и LCMS. LMS предполагает автоматизацию административного управления учебным процессом, а LCMS – автоматизацию управления содержимым (контентом) учебного процесса, хотя на практике границы между этими системами весьма относительны.

Обе системы управляют содержанием курсов и отслеживают результаты обучения. Оба инструмента могут управлять и отслеживать контент, вплоть до уровня учебных объектов. Но система управления обучением, в то же время, может управлять процессом смешанного обучения, составленного из онлайнового контента, мероприятий в учебных классах, встреч в виртуальных учебных классах и т.п. В противовес этому, система управления учебным контентом может руководить содержимым на уровень ниже учебного объекта, что позволяет перестраивать и перенаправлять онлайн-контент. Некоторые LCMS умеют динамически строить учебные объекты в соответствии с профилями пользователей или стилями обучения.

Таким образом, система управления обучением обеспечивает инфраструктуру, позволяющую любому образовательному учреждению планировать, проводить и управлять учебными программами любых форматов на выбор. Она также поддерживает многочисленные средства разработки курсов и легко интегрируется с популярными системами управления содержимым обучения. В этой роли, как катализатор общей учебной среды, LMS может интегрировать в LCMS учебные объекты через технические спецификации и стандарты, а также нести ответственность за управление учебным контентом, включая проигрывание и проверки, хранение контент–репозитория, соединение и разъединение объектов контента, внедрение объектов контента в смешанные процессы, сбор результатов обучения по отдельным курсам.

В недавнем прошлом все электронные обучающие ресурсы создавались с использованием специфичных инструментальных средств, требующих свою среду разработки и функционирования. Разработчики курсов или должны были изучить эти инструментальные средства, или работать с программистами, имеющими опыт работы с ними. Содержимое разрабатывалось заново от курса к курсу и требовалось много сил на разработку и испытания курса.

Learning Content Management System отделяет контент от средств доставки контента. Содержимое может быть создано однократно и доставлено многочисленными способами. LCMS также устраняет потребность в специализированных навыках программирования, поскольку позволяет авторам вставлять содержание в предварительно запрограммированные шаблоны. Поскольку контент создается в виде небольших объектов, разработчики могут повторно использовать содержимое, созданное другими авторами, экономя при этом время на разработку, а также обеспечивая доставку непротиворечивой информации обучающимся.

Таким образом, в связи с бурным ростом объёма информации, интенсивности ее потока возникают трудности в усвоении материала, подготовке учебных и методических материалов. Для устранения вышеизложенных недостатков необходим совершенно новый подход, стиль и новая методика, основанная на использования самых современных информационно-педагогических технологий, где значительный упор делается на возможности современных информационных систем и телекоммуникаций.

Новый подход организации учебного процесса, установления контакта между преподавателем и студентом состоит в том, что преподаватель теперь все в большей степени выполняет функцию координатора. У преподавателя появляется возможность введения коррекции на отклонение от идеальной траектории перехода с одного этапа в следующий. Расширяются возможности обучаемого, т.е. теперь у него появляется возможность войти и в мир знаний преподавателя, воспользоваться базой знаний, виртуальными библиотеками, установить контакт с виртуальными преподавателями, а также произвести объективную самооценку формируемых знаний.

Рассмотрим наиболее популярные LMS на сегодняшний момент:

MOODLE – Modular Object–Oriented Dynamic Learning Environment.

* Официальный сайт: [www.moodle.org](http://www.moodle.org/)
* Поддержка: IMS/SCORM спецификаций
* Платформа: PHP, MySQL, PostgreSQL
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: есть

Дизайн и разработка Moodle направляются особой философией обучения, которую можно вкратце назвать "педагогика социального конструкционизма" (social constructionist pedagogy).

Конструкционизм утверждает, что обучение особенно эффективно, когда учащийся в процессе обучения формирует что–то для других. Это может быть что угодно, от высказывания утверждения или написания сообщения в интернет до более комплексных произведений, таких как картина, дом или пакет программ.

Например, вы можете прочесть эту страницу несколько раз, и всё равно на завтра ничего не помнить. Но если вы попытаетесь объяснить эти идеи кому–нибудь другому своими словами или изготовить слайд–презентацию, объясняющую эти концепции, Вы лучше поймёте их и лучше интегрируете в свои собственные идеи. Вот почему люди делают конспекты во время лекций, даже если никогда не читают их потом.

Claroline

* Официальный сайт: [www.claroline.net](http://www.claroline.net/)
* Поддержка: IMS/SCORM спецификаций
* Языки приложения: PHP, JAVA
* СУБД: MySQL
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: есть
* Демонстрационный сайт: <http://demo.opensourcecms.com/claroline/>

Приложение было создано в Бельгии институте педагогики и мультимедиа католического университета в Лувене.

Dokeos

Платформа построения сайтов дистанционного обучения, основанная на ветке (fork) Claroline. Ветка представляет собой клон свободно распространяемого программного продукта, созданный с целью изменить приложение-оригинал в том или ином направлении.

Dokeos – результат работы некоторых членов первоначальной команды разработчиков Claroline, которые задумали:

* изменить ориентацию приложения. Теперь оно подойдет скорее организациям, чем университетам.
* организовать (скорее выставить на продажу) набор дополнительных сервисов для платформы. Название Dokeos относится как к приложению, так и к сообществу, которое предлагает набор различных сервисов к платформе: хостинг, интегрирование контента, разработка дополнительных модулей, тех. поддержка и т.д.

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

ATutor

Система создана канадскими разработчиками. Включает в себя весь необходимый e–learning инструментарий. Есть русскоязычная версия.

* Официальный сайт: [www.atutor.ca](http://www.atutor.ca/)
* Поддержка: IMS/SCORM
* Языки приложения: PHP, JAVA
* СУБД: MySQL
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: есть
* Демонстрационный сайт: <http://www.atutor.ca/atutor/demo/login.php>

LAMS

* Официальный сайт: [http://www.lamscommunity.org](http://www.lamscommunity.org/)
* Языки приложения: Java
* СУБД: MySQL
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: нет
* Демонстрационный сайт: <http://lamsinternational.com/demo/intro_to_lams.html>

Спецификация IMS Learning Design была подготовлена в 2003 году. В ее основу положены результаты работы Открытого университета Нидерландов (Open University of the Netherlands – OUNL) по языку образовательного моделирования «Educational Modelling Language» (EML), при помощи которого описывается «метамодель» разработки учебного процесса.

На основе данной спецификации была создана «Система управления последовательностью учебных действий» Learning Activity Management System (LAMS). LAMS предоставляет преподавателям визуальные средства для разработки структуры учебного процесса, позволяющие задавать последовательность видов учебной деятельности.

LAMS представляет собой революционно новое приложение для создания и управления электронными образовательными ресурсами. Она предоставляет преподавателю интуитивно понятный интерфейс для создания образовательного контента, который может включать в себя различные индивидуальные задания, задания для групповой работы и фронтальную работу с группой обучаемых.

OLAT

* Официальный сайт: [http://www.olat.org](http://www.olat.org/)
* Стандарты: SCORM/IMS (IMS Content Packaging, IMS QTI)
* Языки приложения: Java
* СУБД: MySQL, PostgreSQL
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: есть
* Демонстрационный сайт: [http://demo.olat.org](http://demo.olat.org/)

Разработка системы началась еще в 1999 году в University of Zurich, Switzerland, где она является основной образовательной платформой электронного обучения.

OpenACS

Open Architecture Community System это система для разработки масштабируемых, переносимых образовательных ресурсов. Она является основой для многих компаний и университетов, занимающихся использованием технологий электронного обучения.

* Официальный сайт: [http://openacs.org](http://openacs.org/)
* СУБД: ORACLE
* Лицензия: GNU General Public License (GPL)
* Поддержка русского языка: есть

Таким образом современные широкомасштабные информационные системы обучения представляют собой сетевые информационные среды обучения, которые могут быть реализованы как при дистанционном обучении, так и при очной форме.

## 2. Разработка алгоритма обучения и компонентов информационной системы

## 2.1 Концепция ИС

Целью дипломной работы является создание программного обеспечения – информационной системы обучения по курсу «Компьютерные сети».

Проанализировав существующие системы обучения и учитывая специфику разработка нашей информационной системы обучения должна включать в себя:

– разработку концепции;

– проектирование ИС;

– разработку репозитория;

– разработку эргономического интерфейса для работы с данными курса;

– разработку системы управления курсом;

– тестирование.

Пользователями системы являются студенты, которые заходят под своим логином и паролем и изучают материал по курсу «Компьютерные сети», а затем проходят тестирование для контроля изученного материала.

Также система содержит настройки администратора, доступные при входе под администраторскими логином и паролем. Администратор имеет возможность настраивать списки пользователей, а также управлять списком тем курса и тестовых модулей.

Общую структуру проекта можно представить следующим образом:

*Список курсов*

*Список пользователей*

*Статистические показатели*

*список*

*тестов*

**База данных**

**Приложение для управления курсом**

*блок регист-и*

*система управления списком курсов*

*блок статистики*

*блок админ-ра*

*Система навигации по курсу*

Рис. 3. Общая структура ИС обучения

Данная ИС обучения предназначена для более удобного управления электронным учебником по компьютерным сетям и тем самым повышения эффективности обучения и самообучения по данному направлению.

Для работы информационной системы не требуется никакого специализированного программного обеспечения.

## 2.2 Проектирование ИС

После определения концепции проекта необходимо смоделировать основные структурные компоненты, их взаимосвязи и процессы, происходящие в нашей информационной системе. Для этого предназначено большое количество диаграмм, которые позволяют наглядно в соответствии со стандартами построения информационных систем изобразить необходимые компоненты системы.

Рассмотрим несколько базовых диаграмм:

1. Диаграмма прецедентов отражает взаимодействие вариантов использования системы и действующих лиц. Она отражает требования к системе с точки зрения пользователя. Помогает провести анализ требований, который подразумевает выделение процессов и требований и их формулировку.

Заказчиком формулируются требования к информационной системе, разработчик изучает автоматизируемый процесс, при этом выявляет основные характеристики будущей системы – составляет спецификации.

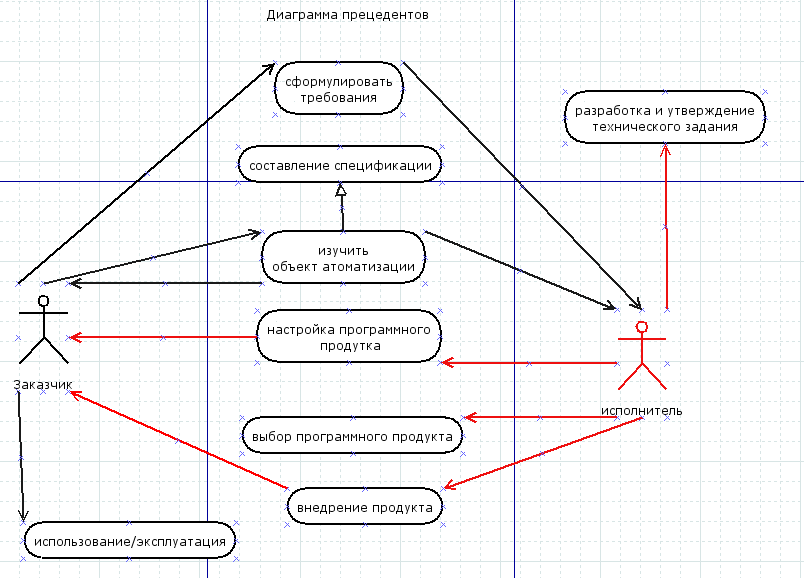


Рис. 4 Диаграмма прецедентов

2. Диаграмма компонентов показывает, как выглядит модель на физическом уровне. На ней изображаются компоненты программного обеспечения системы и связи между ними.

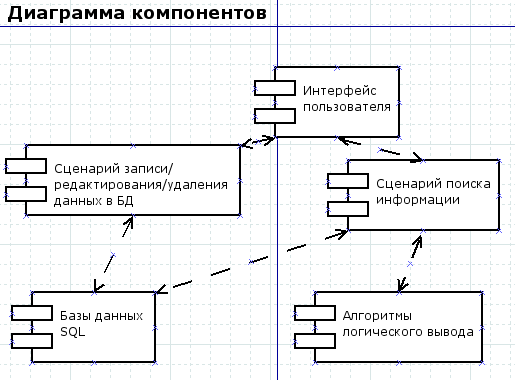


Рис. 5. Диаграмма компонентов

## 2.3 Разработка структуры базы данных

База данных информационной системы обучения представляет собой набор текстовых файлов, в которых содержится структурированная информация по списку пользователей, их результатах обучения, темам курса обучения, тестовому набору.

## 2.4 Разработка интерфейса обучающего курса

Теоретический материал курса представлен в виде гипертекстовых страниц – наиболее удобной форме представления электронных ресурсов. Все главы курса имеют единый стиль оформления и строятся по шаблону: оглавление темы в виде гиперссылок и сам текст главы с большим количеством иллюстраций, контрольные вопросы в конце каждой темы.

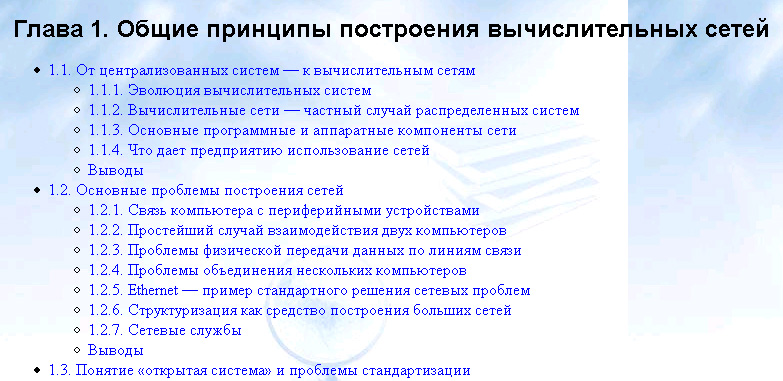


Рис. 6. Пример страницы курса

Разработка web-приложения велась с помощью следующих средств: язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS. Для создания гипертекстовых страниц и элементов каскадных таблиц стилей использовался редактор Macromedia Dreamveawer.

Гипертекстовый курс встраивается в среду Delphi с помощью специализированного компонента – веб–браузера.

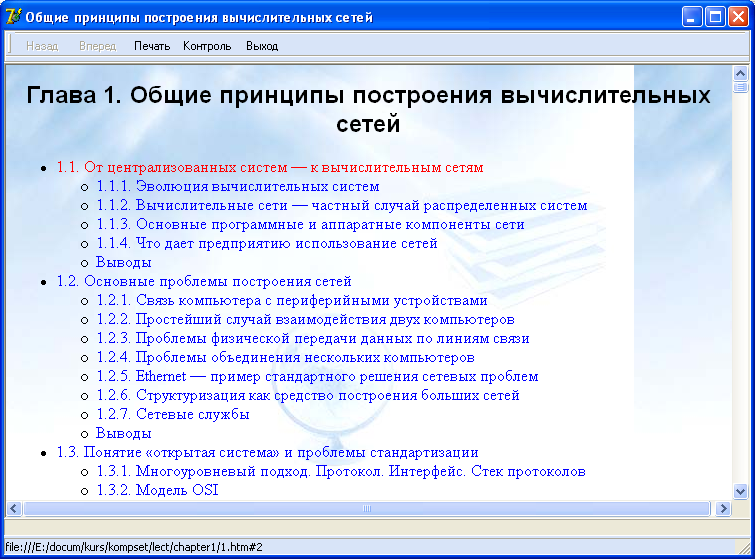


Рис. 7. Компонент «веб–браузер» среды Delphi

Интерфейс тестового комплекса полностью реализован на Delphi. Материал для тестов берется из специальной базы – текстовых файлов. Варианты тестовых наборов генерируются случайным образом. Тестовый материал представлен в виде вопросов с четырьмя вариантами ответа, из которых один правильный.

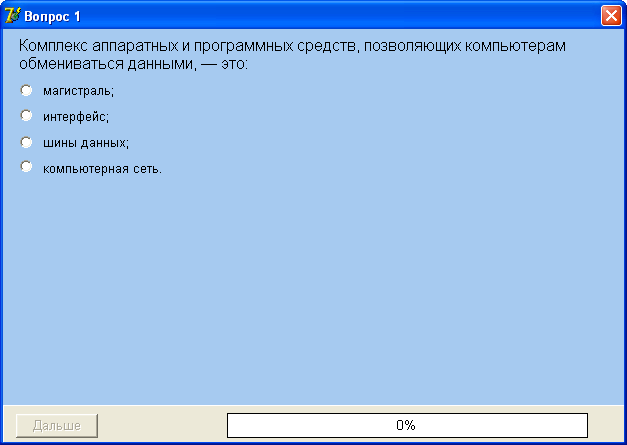


Рис. 8. Тестовый комплекс системы

После прохождения теста осуществляется вывод результатов.

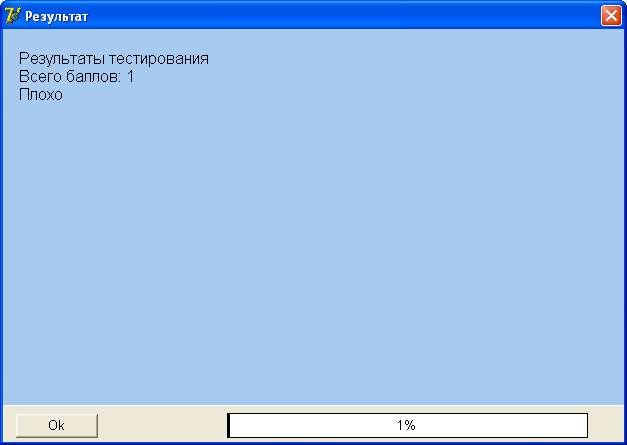


Рис. 9. Страница результатов теста

В зависимости от результатов обучающийся может перейти на новый уровень обучения, то есть ему станет доступно изучение новой темы, или, в случае неудовлетворительного результата, продолжит изучение существующей.

## 2.5 Разработка системы управления курсом

При реализации информационной системы обучения мы придерживались следующих принципов:

* использовалась итерационная (спиральная) модель разработки, т.к. полное завершение работ на каждом из этапов жизненного цикла не обязательно;
* в процессе разработки информационной системы было необходимо тесное взаимодействие с заказчиком и пользователями системы;
* использовалась объектная модель разработки программного обеспечения ИС;
* разработка велась с помощью средств визуальной разработки приложений;
* тестирование и развитие проекта осуществлялось одновременно с разработкой.

В ходе проектирования и разработки информационной системы была применена методология RAD.

Методология разработки информационных систем, основанная на использовании средств быстрой разработки приложений, получила в последнее время широкое распространение и приобрела название методологии быстрой разработки приложений – RAD (Rapid Application Development).

Данная методология охватывает все этапы жизненного цикла современных информационных систем.

RAD – это комплекс специальных инструментальных средств быстрой разработки прикладных информационных систем, позволяющих оперировать с определенным набором графических объектов, функционально отображающих отдельные информационные компоненты приложений.

Под методологией быстрой разработки приложений обычно понимается процесс разработки информационных систем, основанный на трех основных элементах:

* небольшой команде программистов (обычно от 2 до 10 человек);
* тщательно проработанный производственный график работ, рассчитанный на сравнительно короткий срок разработки (от 2 до 6 мес.);
* итерационная модель разработки, основанная на тесном взаимодействии с заказчиком – по мере выполнения проекта разработчики уточняют и реализуют в продукте требования, выдвигаемые заказчиком.

Основные принципы методологии RAD можно свести к следующему:

* используется итерационная (спиральная) модель разработки;
* полное завершение работ на каждом из этапов жизненного цикла не обязательно;
* в процессе разработки информационной системы необходимо тесное взаимодействие с заказчиком и будущими пользователями;
* необходимо применение CASE–средств и средств быстрой разработки приложений;
* необходимо применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы;
* необходимо использование прототипов, позволяющее полнее выяснить и реализовать потребности конечного пользователя;
* тестирование и развитие проекта осуществляются одновременно с разработкой;
* разработка ведется немногочисленной и хорошо управляемой командой профессионалов;
* необходимы грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ.

CASE-технологии (Computer Aided Software/System Engineering) охватывают обширную область поддержки многочисленных технологий проектирования информационных систем: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл программного обеспечения.

Обычно к CASE-средствам относят любое программное средство, автоматизирующее ту или иную совокупность процессов жизненного цикла и обладающее следующими основными характерными особенностями:

* использование специальным образом организованного хранилища проектных метаданных (репозитория);
* мощные графические средства для описания и документирования информационных систем, обеспечивающие удобный интерфейс с разработчиком и развивающие его творческие возможности;
* интеграция отдельных компонент CASE–средств, обеспечивающая управляемость процессом разработки ИС;

Таким образом, исходя из особенностей данных технологий наиболее широко и эффективно в обучении возможно применять именно CASE–средства. Важным фактором, влияющим на успех внедрения подобных систем, является методологический системный подход к их проектированию и реализации. В основе такого подхода лежит использование CASE–технологий, позволяющих выполнять моделирование информационной системы на всех фазах ее разработки: на стадии структурного анализа, проектирования и реализации.

В качестве основного CASE-средства для разработки нашей системы была выбрана среда визуальной разработки Borland Delphi. Основными преимуществами данной среды являются:

* Быстрота и легкость разработки приложения.
* Высокая производительность разработанного приложения
* Низкие требования разработанного приложения к ресурсам компьютера.
* Наращиваемость за счет встраивания новых компонент и инструментов в среду Delphi.
* Возможность разработки новых компонент и инструментов собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных текстах)
* Удобное выстраивание иерархии объектов и тем самым структуры системы

Теперь рассмотрим структуру нашей системы управления курсом:

|  |
| --- |
| Главная страница |

Авторизация

|  |
| --- |
| Страница администратора |

|  |
| --- |
| Изменение списка пользователей |

|  |
| --- |
| Страница выбора тем курса |

|  |
| --- |
| Режим изучения элементов курса |

|  |
| --- |
| Изменение элементов курса (тем, тестов) |

|  |
| --- |
| Режим тестирования |

Рис. 10. Структура программного обеспечения информационной системы управления обучением

Главная страница представляет собой окно с названием системы и полями ввода логина и пароля для авторизации в системе и работы под своим профилем.

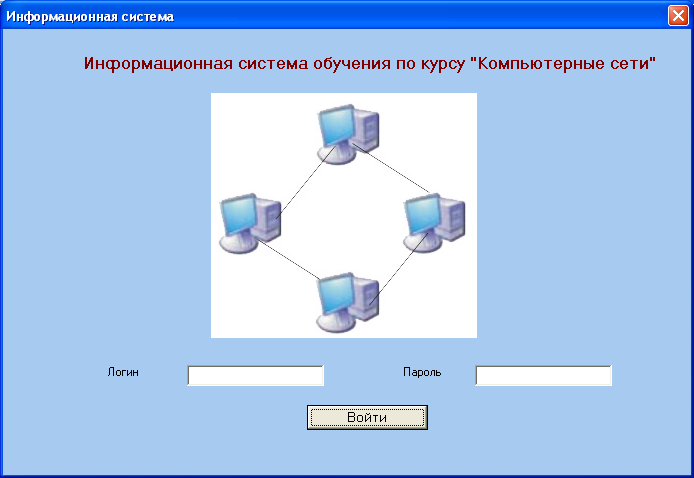


Рис. 11. Заголовочное окно информационной системы

Далее мы попадаем в окно, содержащее краткую информацию о курсе «Компьютерные сети» и выбор тем курса для изучения. Причем в данном окне отображаются только те темы, которые доступны конкретному пользователю в результате прохождения контрольного тестирования. Например, если пользователь зашел в информационную систему первый раз, то в списке тем будет лишь первая.

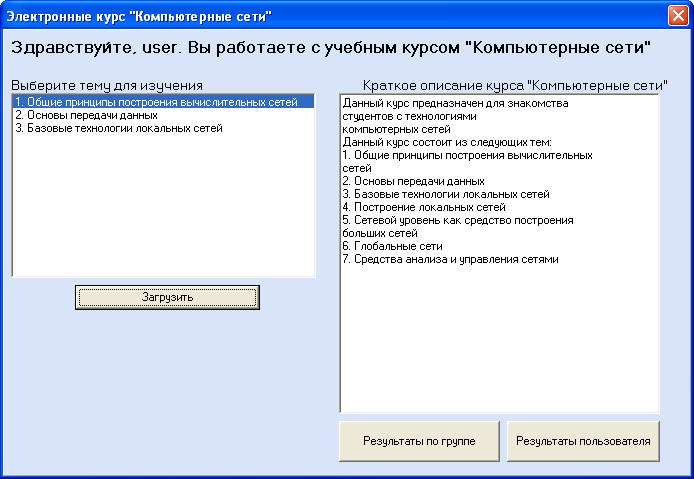


Рис. 12. Окно выбора тем курса

Также в данном окне можно посмотреть свою статистику обучения, а именно: количество баллов, полученных в результате контрольного тестирования по каждой теме, а также статистику группы пользователей в сравнении со своей, где отображается время нахождения внутри курса, количество пройденных тем и общее количество набранных баллов.

После выбора нужной темы и нажатия на кнопку «Загрузить» мы попадаем в режим обучения.

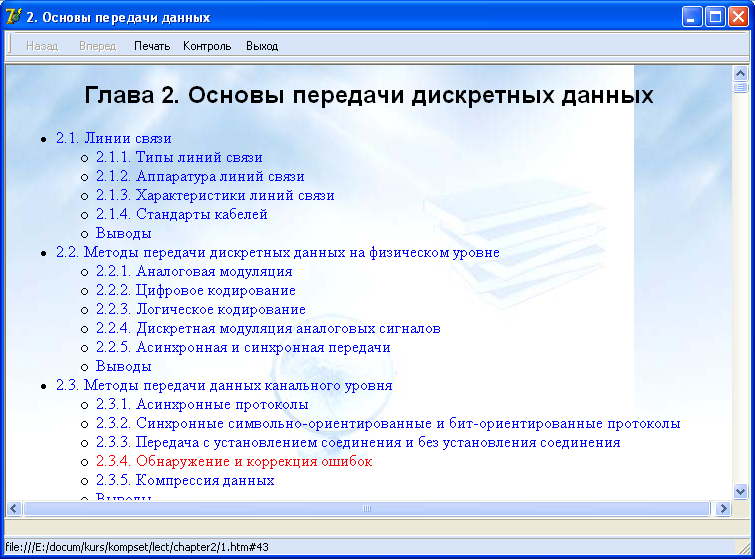


Рис. 13. Окно обучения по выбранной теме.

В данное окно встроен браузер, который позволяет достаточно просто и удобно отображать нужную информацию для чтения. Кнопки, расположенные на панели инструментов, позволяют переместиться назад вперед по тексту, распечатать текст и перейти на страницу контроля знаний, которая рассмотрена выше.

В системе существует также администраторский раздел, который открывается при наборе на главной форме администраторского логина и пароля.

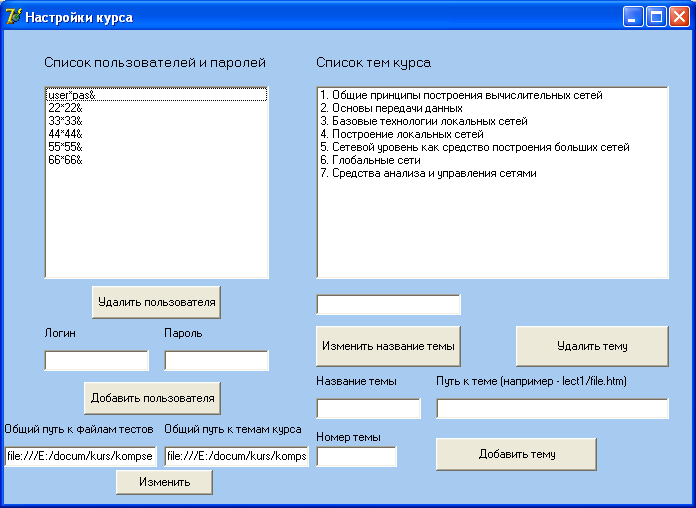


Рис. 14. Окно с настройками системы

Здесь имеется инструмент для управления пользователями, а также списком тем курса.

## 2.7 Тестирование ИС

Тестирование осуществлялось одновременно с разработкой системы (согласно методологии RAD).

Тестирование информационной системы предполагает проверку корректности работы приложения при введении данных.

Тестирование на допустимость вводимых значений подразумевает проверку корректности вводимых данных. Например, при запуске системы происходит проверка вводимого и имеющегося в системе логина и пароля регистрации пользователя.

Таким образом, вследствие применения методологии RAD конфигурационное управление и управление изменениями ИС достаточно легко осуществимо. Это может способствовать модернизации и развитию информационной системы обучения.

Заключение

В результате выполнения дипломной работы была создана информационная система обучения по курсу «Компьютерные сети».

Для создания программного обеспечения ИС использовалась среда визуального проектирования Delphi, сам курс реализован с помощью технологий гипертекста.

Информационная система включает в себя средства для управления курсом (администраторский раздел), средства для обучения и контроля по курсу, а также средства отображения статистической информации.

Таким образом, основная цель дипломной работы достигнута и данная система готова к использованию и последующим модернизациям в современных условиях в РГГУ.

Список использованной литературы

1. Алиев В.С. Информационные технологии и системы финансового менеджмента: учеб. пособие. – М.: «ФОРУМ»: ИНФРА–М, 2007. – 320 с.
2. Гук М. Аппаратные средства локальных сетей. Энциклопедия. – Спб: Издательство «Питер», 2000 – 576 c.
3. Долятовский В.А., Долятовская В.Н. Исследование систем управления: Учебное-практическое пособие. – Москва: ИКЦ «МарТ», 2003 – 256 с.
4. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы построения автоматизированных информационных систем: Учебное пособие. – М: ФОРУМ: ИНФОРМА–М, 2007 – 416 с.
5. Колисниченко Д.Н. Сделай сам компьютерную сеть: монтаж, настройка, обслуживание – Спб.: Наука и Техника, 2004 – 400 с.
6. Компьютерные сети. Принципы, технологии, протоколы. В.Г. Олифер, Н.А. Олифер. – Спб.: Питер, 2001. – 672 с.
7. Компьютерные сети. 4-е изд./Э. Таненбаум – Спб.: Питер, 2003 – 992 с.
8. Новиков Ю.В., Кондратенко С.В. Локальные сети: архитектура, алгоритмы, проектирование. М.: ЭИОТ, 2000 – 312 с.
9. Новые педагогические и информационные технологии в системе образования: Учеб. пособие для студ. пед. вузов и системы повыш. квалиф. пед.кадров/ Под ред. Е.С. Полат. – М.: Издательский центр «Академия», 2001.– 272 с.
10. Пантер М., Синипер Р.Б. Проектирование и внедрение компьютерных сетей. Учебный курс. – 2–е изд., перераб. и доп: Пер. с англ. – Спб.: БХВ– Петербург, 2004. – 752 с.
11. Саак А.Э, Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: Учебник для вузов. – Спб.: Питер, 2005. – 320 с.
12. Семакин И.Г. Информационные системы и модели. Элективный курс: Учебный пособие / И.Г. Семакин, Е.К. Хеннер. – М.: БИНОМ. Лаборатория знаний, 2005. – 303 с.
13. Соловов А.В. Дидактика и технология электронного обучения в системе КАДИС // "Индустрия образования". 6. – М.: МГИУ, 2002, – с. 54–64.
14. Хортон У., Хортон К. Электронное обучение: инструменты и технологии / Пер. с англ. – М.: КУДИЦ–ОБРАЗ, 2005. – 640 с.

Приложение

Исходные коды модулей информационной системы

1. Модуль начальной страницы

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, jpeg;

type

Tzagl = class(TForm)

Button1: TButton;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Button2: TButton;

Image1: TImage;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

agl: Tzagl;

fil,fil1:textfile;

implementation

uses Unit19, Unit44, tester\_, Unit4;

{$R \*.dfm}

procedure Tzagl.Button1Click(Sender: TObject);

var s,s1,s2,log,pas:string;i:integer;k:boolean;

begin

s:='';

reset(fil);

while not eof(fil) do

begin

readln(fil,s2);

s:=s+s2;

end;

closefile(fil);

k:=true;

s1:='';

for i:=1 to length(s) do begin

if (s[i] = '\*') then

begin

log:=s1;

s1:='';

end;

if (s[i] = '&') then

begin

pas:=s1;

if (edit1.text=log)and(edit2.Text=pas) then

begin

k:=false;

ElektKursInf.show;

zagl.Hide;

end;

s1:='';

end;

if (s[i] <> '&')and(s[i] <> '\*') then s1:=s1+s[i];

end;

if (edit1.Text='admin')and(edit2.Text='pasw') then

begin

Form4.Show;

zagl.Hide;

end

else

if k then showmessage('Введите верные логин и пароль');

end;

procedure Tzagl.Button2Click(Sender: TObject);

var s3,s4:string;

begin

if (edit1.text<>'')and(edit2.Text<>'') then begin

append(fil);

s3:=edit1.text+'\*'+edit2.text+'&';

writeln(fil,s3);

closefile(fil);

append(fil1);

s4:=edit1.text+'\*0&0$0#';

writeln(fil1,s4);

closefile(fil1);

showmessage('Вы успешно зарегистрированы!')

end

else showmessage('Введите логин и пароль!')

end;

procedure Tzagl.FormCreate(Sender: TObject);

begin

assignfile(fil,'bd\user.txt');

assignfile(fil1,'bd\result.txt');

end;

end.

2. Модуль выбора темы курса

unit Unit19;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

Menus, StdCtrls, DBCtrls, ExtCtrls, Db, Provider, DBClient, MConnect;

type

TElektKursInf = class(TForm)

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

Button1: TButton;

ListBox1: TListBox;

ListBox2: TListBox;

Memo1: TMemo;

Label3: TLabel;

Button2: TButton;

Button3: TButton;

ListBox3: TListBox;

procedure Exit1Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

ElektKursInf: TElektKursInf;

fil1,fil:textfile;

implementation

uses Unit44, Unit1, Unit3, Unit5;

{$R \*.DFM}

procedure TElektKursInf.Exit1Click(Sender: TObject);

begin

ElektKursInf.Hide;

zagl.close;

end;

procedure TElektKursInf.FormClose(Sender: TObject;

var Action: TCloseAction);

begin

ElektKursInf.Hide;

zagl.Close;

end;

procedure TElektKursInf.FormShow(Sender: TObject);

var i,j:integer;s0,s,s2,k:string;

begin

assignfile(fil,'bd\path1.txt');

reset(fil);

s:='';

while not eof(fil) do

begin

readln(fil,s);

s0:=s;

end;

closefile(fil);

assignfile(fil,'bd\path.txt');

reset(fil);

Listbox1.Clear;

s:='';

while not eof(fil) do

begin

readln(fil,s);

Listbox1.Items.Add(s0+s);

end;

closefile(fil);

assignfile(fil,'bd\themes.txt');

reset(fil);

Listbox3.Clear;

s:='';

while not eof(fil) do

begin

readln(fil,s);

Listbox3.Items.Add(s);

end;

closefile(fil);

Label3.Caption:='Здравствуйте, '+zagl.Edit1.Text+'. Вы работаете с учебным курсом "Компьютерные сети"'; //ComboBox1.ItemIndex := 0;

s:='';

assignfile(fil1,'bd\result.txt');

reset(fil1);

while not eof(fil1) do

begin

readln(fil1,s2);

for i:=1 to length(s2) do begin

if (s2[i] = '\*') then begin

if s=zagl.Edit1.Text then begin

j:=i+1;

while s2[j]<>'#' do begin

if s2[j]='&' then k:=s2[j+1];

j:=j+1;

end;

end;

end;

s:=s+s2[i];

end;

s:='';

end;

closefile(fil1);

{showmessage(k);}

if k='1' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

end;

if k='2' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

end;

if k='3' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

listbox2.items.Add(listbox3.Items.Strings[3]);

end;

if k='4' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

listbox2.items.Add(listbox3.Items.Strings[3]);

listbox2.items.Add(listbox3.Items.Strings[4]);

end;

if k='5' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

listbox2.items.Add(listbox3.Items.Strings[3]);

listbox2.items.Add(listbox3.Items.Strings[4]);

listbox2.items.Add(listbox3.Items.Strings[5]);

end;

if k='6' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

listbox2.items.Add(listbox3.Items.Strings[3]);

listbox2.items.Add(listbox3.Items.Strings[4]);

listbox2.items.Add(listbox3.Items.Strings[5]);

listbox2.items.Add(listbox3.Items.Strings[6]);

end;

if k='7' then begin

listbox2.items.Add(listbox3.Items.Strings[1]);

listbox2.items.Add(listbox3.Items.Strings[2]);

listbox2.items.Add(listbox3.Items.Strings[3]);

listbox2.items.Add(listbox3.Items.Strings[4]);

listbox2.items.Add(listbox3.Items.Strings[5]);

listbox2.items.Add(listbox3.Items.Strings[6]);

listbox2.items.Add(listbox3.Items.Strings[7]);

end;

listbox2.ItemIndex:=0;

end;

procedure TElektKursInf.ComboBox1Change(Sender: TObject);

begin

{with ClientElektKurs do

begin

while not EOF do

begin

ListBox2.items.add(FieldByName('Kurs').AsString);

next;

end;

end;}

end;

procedure TElektKursInf.Button1Click(Sender: TObject);

begin

ListBox1.ItemIndex:=ListBox2.ItemIndex;

Form37.WebBrowser1.Navigate(ListBox1.Items.Strings[ListBox1.ItemIndex]);

//Form37.ComboBox1.Text:=ListBox1.Items.Strings[ListBox1.ItemIndex];

Form37.ToolButton2.Enabled:=false;

Form37.ToolButton3.Enabled:=false;

Form37.Show;

Button1.Enabled:=false;

Label3.Caption:='';

//ElektKursInf.Hide;

end;

procedure TElektKursInf.Button2Click(Sender: TObject);

begin

Form3.Show;

end;

procedure TElektKursInf.Button3Click(Sender: TObject);

begin

Form5.Show;

end;

end.

3. Модули статистики по группам и индивидуально

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids;

type

TForm3 = class(TForm)

StringGrid1: TStringGrid;

procedure FormShow(Sender: TObject);

procedure StringGrid1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

s,s2:string;

fil1:textfile;

i,j:integer;

implementation

{$R \*.dfm}

procedure TForm3.FormShow(Sender: TObject);

var s0:double;

begin

StringGrid1.Cells[0,0]:='Логин';

StringGrid1.Cells[1,0]:='Время обучения(мин.)';

StringGrid1.Cells[2,0]:='Количество пройденных тем';

StringGrid1.Cells[3,0]:='Количество баллов';assignfile(fil1,'bd\result.txt');

reset(fil1);

s:='';

i:=1;

while not eof(fil1) do

begin

readln(fil1,s2);

for j:=1 to length(s2) do begin

if s2[j]='\*' then begin

StringGrid1.Cells[0,i]:=s;

s:='';

continue;

end;

if s2[j]='&' then begin

s0:=strtofloat(s)/60000;

StringGrid1.Cells[1,i]:=floattostr(s0);

s:='';

continue;

end;

if s2[j]='$' then begin

StringGrid1.Cells[2,i]:=s;

s:='';

continue;

end;

if s2[j]='#' then begin

StringGrid1.Cells[3,i]:=s;

s:='';

continue;

end;

s:=s+s2[j];

end;

i:=i+1;

StringGrid1.RowCount:=i;

end;

closefile(fil1);

end;

end.

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids;

type

TForm5 = class(TForm)

StringGrid1: TStringGrid;

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

s,s1,s2,s3:string;

fil1:textfile;

i,j:integer;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm5.FormShow(Sender: TObject);

begin

StringGrid1.Cells[0,0]:='1 тема';

StringGrid1.Cells[1,0]:='2 тема';

StringGrid1.Cells[2,0]:='3 тема';

StringGrid1.Cells[3,0]:='4 тема';

StringGrid1.Cells[4,0]:='5 тема';

StringGrid1.Cells[5,0]:='6 тема';

StringGrid1.Cells[6,0]:='7 тема';

assignfile(fil1,'bd\result1.txt');

reset(fil1);

s:='';

s3:='';

while not eof(fil1) do

begin

readln(fil1,s2);

s:=s+s2;

end;

for j:=1 to length(s) do begin

if s[j]='\*' then begin

if s1=zagl.Edit1.Text then begin

i:=j+1;

while s[i]<>'\*' do begin

if s[i]='!' then begin

StringGrid1.Cells[0,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='$' then begin

StringGrid1.Cells[1,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='#' then begin

StringGrid1.Cells[2,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='%' then begin

StringGrid1.Cells[3,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='^' then begin

StringGrid1.Cells[4,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='@' then begin

StringGrid1.Cells[5,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

if s[i]='~' then begin

StringGrid1.Cells[6,1]:=s3;

s3:='';

i:=i+1;

continue;

end;

s3:=s3+s[i];

i:=i+1;

end;

end;

s1:='';

end;

s1:=s1+s[j];

if s[j]='~' then s1:='';

end;

closefile(fil1);

end;

end.

4. Модуль обучения по курсу

unit Unit44;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

OleCtrls, SHDocVw, ToolWin, ComCtrls, StdCtrls, ExtDlgs, ExtCtrls, Menus;

const

HTMLID\_FIND = 1;

HTMLID\_VIEWSOURCE = 2;

HTMLID\_OPTIONS = 3;

type

TForm37 = class(TForm)

WebBrowser1: TWebBrowser;

StatusBar1: TStatusBar;

ProgressBar1: TProgressBar;

OpenDialog1: TOpenDialog;

CoolBar1: TCoolBar;

ToolBar1: TToolBar;

ToolButton2: TToolButton;

ToolButton3: TToolButton;

ToolButton6: TToolButton;

ComboBox1: TComboBox;

ToolButton8: TToolButton;

ToolButton1: TToolButton;

Timer1: TTimer;

PopupMenu1: TPopupMenu;

N1: TMenuItem;

procedure ComboBox1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure ToolButton1Click(Sender: TObject);

procedure ToolButton2Click(Sender: TObject);

procedure ToolButton3Click(Sender: TObject);

procedure ToolButton4Click(Sender: TObject);

procedure ToolButton5Click(Sender: TObject);

procedure ToolButton6Click(Sender: TObject);

procedure WebBrowser1StatusTextChange(Sender: TObject;

const Text: WideString);

procedure WebBrowser1ProgressChange(Sender: TObject; Progress,

ProgressMax: Integer);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure ToolButton7Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure WebBrowser1BeforeNavigate2(Sender: TObject;

const pDisp: IDispatch; var URL, Flags, TargetFrameName, PostData,

Headers: OleVariant; var Cancel: WordBool);

procedure WebBrowser1NavigateComplete2(Sender: TObject;

const pDisp: IDispatch; var URL: OleVariant);

procedure ToolButton8Click(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{procedure ExecWB(cmdID: OLECMDID; cmdexecopt: OLECMDEXECOPT); overload;}

{ Public declarations }

end;

var

Form37: TForm37;

z,time1:integer;m:string;

implementation

uses Unit19, kontr;

{$R \*.DFM}

procedure TForm37.ComboBox1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

{if Key = VK\_RETURN then

WebBrowser1.Navigate(ComboBox1.Text);}

end;

procedure TForm37.ToolButton1Click(Sender: TObject);

begin

{ if OpenDialog1.Execute then

begin

WebBrowser1.Navigate(OpenDialog1.FileName);

ComboBox1.Text := OpenDialog1.FileName;

end;}

timer1.Enabled:=false;

close;

end;

procedure TForm37.ToolButton2Click(Sender: TObject);

begin

z:=z+1;

WebBrowser1.GoBack;

ToolButton3.Enabled:=true;

end;

procedure TForm37.ToolButton3Click(Sender: TObject);

begin

z:=z–1;

if z>–1 then WebBrowser1.GoForward else ToolButton3.Enabled:=false;

end;

procedure TForm37.ToolButton4Click(Sender: TObject);

begin

WebBrowser1.Stop;

end;

procedure TForm37.ToolButton5Click(Sender: TObject);

begin

WebBrowser1.Refresh;

end;

procedure TForm37.ToolButton6Click(Sender: TObject);

var

PostData, Headers: OLEvariant;

begin

WebBrowser1.ExecWB(OLECMDID\_PRINT, OLECMDEXECOPT\_DODEFAULT, PostData, Headers);

end;

procedure TForm37.WebBrowser1StatusTextChange(Sender: TObject; const Text: WideString);

begin

StatusBar1.SimpleText := Text;

end;

procedure TForm37.WebBrowser1ProgressChange(Sender: TObject; Progress, ProgressMax: Integer);

begin

ProgressBar1.Max := ProgressMax;

ProgressBar1.Position := Progress;

end;

procedure TForm37.FormClose(Sender: TObject; var Action: TCloseAction);

begin

timer1.Enabled:=false;

Form37.Hide;

ElektKursInf.Show;

ElektKursInf.Button1.Enabled:=true;

end;

procedure TForm37.ToolButton7Click(Sender: TObject);

const CGID\_WebBrowser: TGUID = '{ED016940–BD5B–11cf–BA4E–00C04FD70816}';

{var

CmdTarget : IOleCommandTarget;

vaIn, vaOut: OleVariant;

PtrGUID: PGUID;}

begin

{New(PtrGUID);

PtrGUID^ := CGID\_WebBrowser;

if WebBrowser1.Document <> nil then

try

WebBrowser1.Document.QueryInterface(IOleCommandTarget, CmdTarget);

if CmdTarget <> nil then

try

CmdTarget.Exec( PtrGUID, HTMLID\_FIND, 0, vaIn, vaOut);

finally

CmdTarget.\_Release;

end;

except

// nothing

end;

Dispose(PtrGUID);}

end;

procedure TForm37.FormShow(Sender: TObject);

begin

time1:=0;

timer1.Enabled:=true;

Form37.Caption:=ElektKursInf.ListBox2.Items.Strings[ElektKursInf.ListBox2.ItemIndex];

end;

procedure TForm37.WebBrowser1BeforeNavigate2(Sender: TObject;

const pDisp: IDispatch; var URL, Flags, TargetFrameName, PostData,

Headers: OleVariant; var Cancel: WordBool);

var l:integer;

begin

end;

procedure TForm37.WebBrowser1NavigateComplete2(Sender: TObject;

const pDisp: IDispatch; var URL: OleVariant);

begin

Combobox1.Text:=WebBrowser1.LocationURL;

if (ElektKursInf.ListBox1.Items.Strings[ElektKursInf.ListBox1.ItemIndex]<>WebBrowser1.LocationURL)then ToolButton2.Enabled:=true else ToolButton2.Enabled:=false;

end;

procedure TForm37.ToolButton8Click(Sender: TObject);

begin

Form2.Show;

Form37.Hide;

end;

procedure TForm37.Timer1Timer(Sender: TObject);

begin

time1:=time1+timer1.interval;

end;

procedure TForm37.N1Click(Sender: TObject);

begin

close;

end;

end.

5. Модуль контрольного тестирования

unit tester\_;

interface

uses

SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,

Forms, Dialogs, StdCtrls, ExtCtrls,

jpeg, Gauges;

type

TForm1 = class(TForm)

// вопрос

Label5: TLabel;

// альтернативные ответы

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

// радиокнопки выбора ответа

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

RadioButton3: TRadioButton;

RadioButton4: TRadioButton;

Image1: TImage; // область вывода иллюстрации

Button1: TButton;

Panel1: TPanel;

RadioButton5: TRadioButton;

Gauge1: TGauge;

Timer1: TTimer;

procedure FormActivate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure RadioButtonClick(Sender: TObject);

// Эти объявления вставлены сюда вручную

procedure Info;

procedure VoprosToScr;

procedure ShowPicture; // выводит иллюстрацию

procedure ResetForm; // "очистка" формы перед выводом очередного вопроса

procedure Itog;

procedure FormCreate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject); // результат тестирования

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1; // форма

implementation

uses Unit19, Unit1, Unit44;

{$R \*.DFM}

const

N\_LEV=4; // четыре уровня оценки

N\_ANS=4; // четыре варианта ответов

var

f,fil1:TextFile;

fn:string; // имя файла вопросов

level:array[1..N\_LEV] of integer; // сумма, соответствующая уровню

mes:array[1..N\_LEV] of string; // сообщение, соответствующее уровню

score:array[1..N\_ANS] of integer; // оценка за выбор ответа

summa:integer; // набрано очков

vopros:integer; // номер текущего вопроса

otv:integer; // номер выбранного ответа

// вывод информации о тесте

procedure Tform1.Info;

var

s,buf:string;

begin

readln(f,s);

Form1.Caption := s;

buf:='';

repeat

readln(f,s);

if s[1] <> '.'

then buf := buf +s + #13;

until s[1] ='.';

Label5.caption:=buf;

end;

// прочитать информацию об оценках за тест

Procedure GetLevel;

var

i:integer;

buf:string;

begin

i:=1;

repeat

readln(f,buf);

if buf[1] <> '.' then begin

mes[i]:=buf; // сообщение

readln(f,level[i]); // оценка

i:=i+1;

end;

until buf[1]='.';

end;

// масштабирование иллюстрации

Procedure TForm1.ShowPicture;

var

w,h: integer; // максимально возможные размеры картинки

begin

// вычислить допустимые размеры картинки

w:=ClientWidth–10;

h:=ClientHeight

– Panel1.Height –10

– Label5.Top

– Label5.Height – 10;

// вопросы

if Label1.Caption <> ''

then h:=h–Label1.Height–10;

if Label2.Caption <> ''

then h:=h–Label2.Height–10;

if Label3.Caption <> ''

then h:=h–Label3.Height–10;

if Label4.Caption <> ''

then h:=h–Label4.Height–10;

// если размер картинки меньше w на h,

// то она не масштабируется

Image1.Top:=Form1.Label5.Top+Label5.Height+10;

if Image1.Picture.Bitmap.Height > h

then Image1.Height:=h

else Image1.Height:= Image1.Picture.Height;

if Image1.Picture.Bitmap.Width > w

then Image1.Width:=w

else Image1.Width:=Image1.Picture.Width;

Image1.Visible := True;

end;

// вывести вопрос

Procedure TForm1.VoprosToScr;

var

i:integer;

s,buf:string;

ifn:string; // файл иллюстрации

begin

vopros:=vopros+1;

caption:='Вопрос ' + IntToStr(vopros);

// прочитать вопрос

buf:='';

repeat

readln(f,s);

if (s[1] <> '.') and (s[1] <> '\')

then buf:=buf+s+' ';

until (s[1] ='.') or (s[1] ='\');

Label5.caption:=buf; // вывести вопрос

{Иллюстрацию прочитаем, но выведем только после того, как прочитаем альтернативные ответы и определим максимально возможный размер области формы, который можно использовать для ее вывода.}

if s[1] <> '\'

then Image1.Tag:=0 // к вопросу нет иллюстрации

else // к вопросу есть иллюстрация

begin

Image1.Tag:=1;

ifn:=copy(s,2,length(s));

try

Image1.Picture.LoadFromFile(ifn);

except

on E:EFOpenError do

Image1.Tag:=0;

end;

end;

// Читаем варианты ответов

i:=1;

repeat

buf:='';

repeat // читаем текст варианта ответа

readln(f,s);

if (s[1]<>'.') and (s[1] <> ',')

then buf:=buf+s+' ';

until (s[1]=',')or(s[1]='.');

// прочитан альтернативный ответ

score[i]:= StrToInt(s[2]);

case i of

1: Label1.caption:=buf;

2: Label2.caption:=buf;

3: Label3.caption:=buf;

4: Label4.caption:=buf;

end;

i:=i+1;

until s[1]='.';

// здесь прочитана иллюстрация и альтернативные ответы

// текст вопроса уже выведен

if Image1.Tag =1 // есть иллюстрация к вопросу

then ShowPicture;

// вывод альтернативных ответов

if Form1.Label1.Caption <> ''

then begin

if Form1.Image1.Tag =1

then Label1.top:=Image1.Top+Image1.Height+10

else Label1.top:=Label5.Top+Label5.Height+10;

RadioButton1.top:=Label1.top;

Label1.visible:=TRUE;

RadioButton1.visible:=TRUE;

end;

if Form1.Label2.Caption <> ''

then begin

Label2.top:=Label1.top+ Label1.height+10;

RadioButton2.top:=Label2.top;

Label2.visible:=TRUE;

RadioButton2.visible:=TRUE;

end;

if Form1.Label3.Caption <> ''

then begin

Label3.top:=Label2.top+ Label2.height+10;

RadioButton3.top:=Label3.top;

Label3.visible:=TRUE;

RadioButton3.visible:=TRUE;

end;

if Form1.Label4.Caption <> ''

then begin

Label4.top:=Label3.top+ Label3.height+10;

RadioButton4.top:=Label4.top;

Label4.visible:=TRUE;

RadioButton4.visible:=TRUE;

end;

end;

Procedure TForm1.ResetForm;

begin // сделать невидимыми все метки и радиокнопки

Label1.Visible:=FALSE;

Label1.caption:='';

Label1.width:=ClientWidth–Label1.left–5;

RadioButton1.Visible:=FALSE;

Label2.Visible:=FALSE;

Label2.caption:='';

Label2.width:=ClientWidth–Label2.left–5;

RadioButton2.Visible:=FALSE;

Label3.Visible:=FALSE;

Label3.caption:='';

Label3.width:=ClientWidth–Label3.left–5;

RadioButton3.Visible:=FALSE;

Label4.Visible:=FALSE;

Label4.caption:='';

Label4.width:=ClientWidth–Label4.left–5;

RadioButton4.Visible:=FALSE;

Label5.width:=ClientWidth–Label5.left–5;

Image1.Visible:=FALSE;

end;

// определение достигнутого уровня

procedure TForm1.Itog;

var

i:integer;

buf:string;

begin

buf:='';

buf:='Результаты тестирования'+ #13

+'Всего баллов: '+ IntToStr(summa);

i:=1;

while (summa < level[i]) and (i<N\_LEV) do

i:=i+1;

buf:=buf+ #13+mes[i];

Label5.Top:=20;

Label5.caption:=buf;

end;

procedure TForm1.FormActivate(Sender: TObject);

begin

end;

// щелчок на кнопке Button1

procedure TForm1.Button1Click(Sender: TObject);

var s,s2,s1,s3,s4,s5,s6,s7,k,tim1:string;i,j,l,l1,l2:integer;tim:integer;

begin

case Button1.tag of

0: begin

Button1.caption:='Дальше';

Button1.tag:=1;

RadioButton5.Checked:=TRUE;

// вывод первого вопроса

Timer1.Enabled:=true;

Gauge1.Visible:=true;

Button1.Enabled:=False;

ResetForm;

VoprosToScr;

end;

1: begin // вывод остальных вопросов

summa:=summa+score[otv];

RadioButton5.Checked:=TRUE;

Button1.Enabled:=False;

ResetForm;

if not eof(f)

then VoprosToScr

else

begin

summa:=summa+score[otv];

closefile(f);

Button1.caption:='Ok';

Form1.caption:='Результат';

Button1.tag:=2;

Button1.Enabled:=TRUE;

Itog; // вывести результат

end;

end;

2: begin // завершение работы

Timer1.Enabled:=false;

s:='';

assignfile(fil1,'bd\result.txt');

reset(fil1);

s:='';

while not eof(fil1) do

begin

readln(fil1,s2);

s:=s+s2;

end;

closefile(fil1);

s1:='';

for i:=1 to length(s) do begin

if (s[i] = '\*') then begin

if s1=zagl.Edit1.Text then begin

j:=i+1;

s3:='';

while s[j]<>'#' do begin

if s[j]='&' then begin

tim:=time1+StrToInt(s3);

delete(s,i+1,j–1–i–1);

tim1:=FloatToStr(tim);

insert(tim1,s,i+1);

s3:='';

end;

if s[j]='$' then begin

if summa>=7 then begin

l:=StrToInt(s[j–1])+1;

delete(s,j–1,1);

s7:=IntToStr(l);

insert(s7,s,j–1);

s3:='';

end

else l:=StrToInt(s[j–1]);

if summa>=7 then begin

s5:=s[j+1]+s[j+2];

l2:=StrToInt(s5);

delete(s,j+1,2);

l2:=l2+summa;

s6:=IntToStr(l2);

if length(s6)=1 then s6:=' '+s6;

insert(s6,s,j+1);

end;

end;

s3:=s3+s[j];

j:=j+1;

end;

end;

end;

s1:=s1+s[i];

if (s[i]='#') then s1:='';

end;

rewrite(fil1);

s4:='';

for i:=1 to length(s) do

begin

s4:=s4+s[i];

if s[i]='#' then begin

writeln(fil1,s4);

s4:='';

end;

end;

closefile(fil1);

assignfile(fil1,'bd\result1.txt');

reset(fil1);

s:='';

while not eof(fil1) do

begin

readln(fil1,s2);

s:=s+s2;

end;

closefile(fil1);

s1:='';

for i:=1 to length(s) do begin

if (s[i] = '\*') then begin

//showmessage(s1);

if s1=zagl.Edit1.Text then begin

j:=i+1;

s3:='';

while s[j]<>'~' do begin

if (s[j]='!')and(l=1)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='$')and(l=2)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='#')and(l=3)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='%')and(l=4)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='^')and(l=5)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='@')and(l=6)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

if (s[j]='~')and(l=7)and(summa>StrToInt(s[j–1])) then begin

delete(s,j–1,1);

s6:=IntToStr(summa);

insert(s6,s,j–1);

end;

s3:=s3+s[j];

j:=j+1;

end;

end;

end;

s1:=s1+s[i];

if (s[i]='~') then s1:='';

end;

rewrite(fil1);

s4:='';

for i:=1 to length(s) do

begin

s4:=s4+s[i];

if s[i]='~' then begin

writeln(fil1,s4);

s4:='';

end;

end;

closefile(fil1);

Form1.Close;

ElektKursInf.Button1.Enabled:=true;

end;

end;

end;

// Процедура обработки события OnClick

// для компонентов RadioButton1–RadioButton4

procedure TForm1.RadioButtonClick(Sender: TObject);

begin

if sender = RadioButton1

then otv:=1

else if sender = RadioButton1

then otv:=2

else if sender = RadioButton3

then otv:=3

else otv:=4;

Button1.enabled:=TRUE;

end;

// обеспечивает настройку компонентов

procedure TForm1.FormCreate(Sender: TObject);

begin

Image1.AutoSize := False;

Image1.Proportional := True;

RadioButton1.Visible := False;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

Gauge1.Progress:=Gauge1.Progress+1;

if Gauge1.Progress=100 then begin

ResetForm;

summa:=summa+score[otv];

Button1.caption:='Ok';

Form1.caption:='Результат';

Button1.tag:=2;

Button1.Enabled:=TRUE;

Itog; // вывести результат

Timer1.Enabled:=false;

end;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

ElektKursInf.Button1.Enabled:=true;

end;

procedure TForm1.FormShow(Sender: TObject);

var z:integer;s0:string;

begin

assignfile(fil,'bd\path1.txt');

reset(fil);

s0:='';

while not eof(fil) do

begin

readln(fil,s0);

break;

end;

closefile(fil);

randomize;

ResetForm;

z:=random(4)+1;

fn:=copy(s0,9,length(s0))+IntToStr(ElektKursInf.ListBox1.ItemIndex+1)+IntToStr(z)+'.txt';

assignfile(f,fn);

try

reset(f);

except

on EFOpenError do

begin

ShowMessage('Файл теста '+fn+' не найден.');

Button1.caption:='Ok';

Button1.tag:=2;

Button1.Enabled:=TRUE;

exit;

end;

end;

Info; // прочитать и вывести информацию о тесте

GetLevel; // прочитать информацию об уровнях оценок

end;

end.