в профессиональной области, лекция должна ориентировать студентов и в этом направлении, давая впоследствии возможность на базе фундаментальных знаний решать прикладные задачи профессиональной области и свободно ориентироваться в информационном пространстве.

Следующий этап предполагает решение типовых задач информатики, лежащих в основе всей последующей деятельности, связанной с этим модулем содержания (инвариантная часть). Тренинг-минимум включает в себя выполнение фронтальной лабораторной работы и самостоятельную работу студентов, которая организуется под управлением многофункционального предметно-ориентированного учебно-информационного средства (УИС). Переход от фронтальной формы работы к индивидуальной осуществляется за счет возможностей адаптивных обучающих программ. Завершается блок контрольным срезом, по результатам которого организуется дальнейшая деятельность студентов.

Рис. 1. Организационные формы обучения информатике в структуре интегральной технологии обучения

Семинар-практикум представляет собой наиболее адекватную форму организации занятий по информатике, учитывающую существенные различия в уровне готовности студентов к изучению дисциплины. Эта форма занятий позволяет организовать работу студентов в малых группах, а также индивидуальный компьютерный практикум – более высокую форму по сравнению с фронтальными лабораторными работами. Характерной чертой является разнотипность заданий по уровню сложности, бóльшая самостоятельность, опора на дополнительную учебную и справочную информацию. Эта форма занятий требует предварительной самостоятельной внеаудиторной подготовки студентов и является базой для самостоятельной учебно-исследовательской и научно-исследовательской деятельности студентов.

Сквозной формой организации обучения информатике является метод проектов. Защита проекта, как результата коллективной деятельности студентов, завершает изучение внешнего модуля дисциплины.

Технология реализована в рамках методической системы обучения информатике студентов Института искусств и Академии управления, сервиса и рекламы Тамбовского государственного университета им. Г.Р. Державина.

ЛИТЕРАТУРА

1. *Гузеев В.В.* Планирование результатов образования и образовательная технология. М.: Нар. образование, 2001.

Поступила в редакцию 21 декабря 2007 г.

АВТОМАТИЗИРОВАННЫЙ ОБУЧАЮЩИЙ ПРОГРАММНЫЙ КОМПЛЕКС ПО ТЕОРИИ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКОЙ СТАТИСТИКЕ, РАЗРАБОТАННЫЙ НА ОСНОВЕ МЕТОДА ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

© А.П. Зубаков, М.Н. Толмачева

В настоящее время проникновение компьютерных информационных технологий в процесс обучения достигло такого уровня, что настоятельной научно-практической проблемой стало создание интерактивных автоматизированных обучающих программных комплексов, а также разработка методик автоматизированного взаимодействия учебного комплекса с обучающимися, ориентированных на улучшение качества усвоения знаний последними [1].

Целью настоящей работы было создание автоматизированного обучающего программного комплекса по курсу «Теория вероятностей и математическая статистика», позволяющего интенсифицировать процесс обучения за счет активного взаимодействия системы «студент – компьютер – преподаватель», реализующего принципы оптимального управления, а также дающего возможность преподавателю осуществлять автоматизированный централизованный сбор информации о процессе обучения для её последующего анализа и принятия управляющих решений [2, 3].

Рассмотрим управляемую систему – систему выбора и решения контрольных заданий при обучении, состояние которой в каждый момент времени характеризуется n-мерным вектором x – набором заданий x_1 , ..., x_n по определенной теме. Предполагаем, что t – то количество этапов, которые необходимо пройти, чтобы оказаться на финише, изменяется дискретно и принимает целочисленные значения 0, 1, ... Так, для нашей

обучающей системы дискретным значениям t могут отвечать наборы заданий x, распределенные по соответствующим темам: Тема 1, Тема 2, …, Тема N.

Обучающемуся необходимо пройти весь путь от начала до конца. Контроль процесса обучения будет сводиться к ответу на контрольные задания x_1 , ..., x_n, каждое из которых имеет определенный вес, назначенный разработчиком системы. Условием перехода от одной темы к следующей за ней теме будет являться превышение суммы весов (баллов), набранных за правильно выполненные задания по предыдущим темам некоторой пороговой величины, определяемой разработчиком системы. Неверный ответ наказывается отрицательным весом (штрафом), высоким у легких и невысоким у сложных заданий.

Предполагаем, что на каждом шаге на систему оказывается управляющее воздействие при помощи m-мерного вектора управления u с компонентами u_1, ..., u_m. В данном случае управляющим воздействием является выбор предложенных заданий по теме и переход к последующим темам (на рис. 1 управляющие воздействия показаны линиями, связывающими векторы заданий и сами задания по определенной теме между собой). Таким образом, на определенном этапе тестирования t состояние системы характеризуется вектором контрольных заданий $x(t)$, а управляющее воздействие – вектором выбора того или иного задания $u(t)$. На выбор управления обычно бывают наложены ограничения, которые в нашей обучающей системе связаны с определенным количеством контрольных заданий, а также с пороговой величиной.

Ограничения в достаточно общей форме можно представить в виде

$$u(t) \in U,\ t = 0, 1, \ldots \qquad (1)$$

Здесь U – заданное множество в n-мерном пространстве.

Под влиянием принятого решения, связанного с выбором другого задания, система переходит в следующий момент времени в новое состояние, т. е. к новому заданию. Этот переход можно описать соотношением

$$x(t+1) = f(x(t), u(t)),\ t = 0, 1, \ldots \qquad (2)$$

Здесь $f(x, u)$ – n-мерная функция от n-мерного вектора заданий x и m-мерного вектора выбора заданий u, характеризующая динамику рассматриваемой системы. Эта функция предполагается известной (заданной) и отвечает принятой математической модели рассматриваемого управляемого процесса.

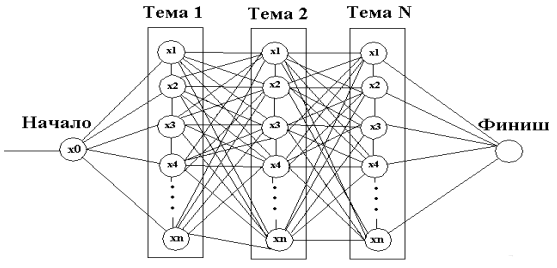

Рис. 1. Схематическое изображение многостадийного процесса

Уровень сложности предлагаемых заданий, а также пороговую величину, определяющую переход к следующей теме, нужно определить еще на начальном этапе построения обучающей системы. Исходя из набранных баллов система будет адаптироваться к каждому пользователю и не предлагать слабому пользователю сложных заданий, по крайней мере, на начальных этапах. Когда слабый пользователь начнет быстрее набирать необходимое количество баллов, ему будут предъявляться к решению уже более сложные задания, что значительно ускорит его процесс обучения.

Зададим еще начальное состояние системы

$$x(0) = x0, \qquad (3)$$

где $x0$ – заданный n-мерный вектор. Таким образом, многошаговый процесс управления описывается соотношениями (1)–(3). Процедура расчета конкретного процесса сводится к следующему. Пусть на некотором этапе t состояние системы $x(t)$ известно. Тогда для определения состояния $x(t + 1)$ необходимо выполнить две операции: 1) выбрать допустимое управление $u(t)$, удовлетворяющее условию (1); 2) определить состояние $x(t + 1)$ на следующем этапе согласно (2). Так как начальное состояние системы задано, то описанную процедуру можно последовательно выполнить для всех $t = 0, 1,\ldots$. Последовательность состояний $x(0)$, $x(1)$, ... часто называется траекторией системы.

Заметим, что выбор управления на каждом шаге содержит значительный произвол. Этот произвол исчезает, если задать цель управления в виде требования минимизации (или максимизации) некоторого критерия оптимальности процесса обучения. Таким образом, мы приходим к постановке задачи оптимального управления Беллмана [3].

Пусть задан некоторый критерий качества процесса управления (критерий оптимальности обучения) вида

$$J = \sum_{t=0}^{N-1} R(x(t), u(t)) + F(x(N)), \qquad (4)$$

здесь $R(x, u)$ и $F(x)$ – заданные скалярные функции своих аргументов, N – момент окончания процесса, $N > 0$. При этом функция R характеризует динамику рассматриваемой системы на каждом шаге процесса, а функция F – характеризует оценку конечного состояния.

Задача оптимального управления формулируется как задача определения допустимых управлений $u(0)$, $u(1)$, ..., $u(N – 1)$ – вариантов выбора задания, удовлетворяющих ограничениям (1), и соответствующей траектории, то есть последовательности заданий $x(0)$, $x(1)$, ..., $x(N)$, которые в совокупности доставляют минимальное значение критерию (4) для процесса (2), (3).

Минимизация критерия (4) отвечает выбору управления, обеспечивающего наименьшие затраты на процесс обучения (т. е. сводится к минимизации этапов, которые необходимо пройти, чтобы оказаться в точке финиша – точке окончания процесса обучения) [4].

Для обучения, а также предъявления последовательности заданий необходимо определить форму представления информации.

Документы, представленные в электронной форме на ЭВМ и использующие новейшие информационные технологии, имеют несколько весомых отличий от тра-

диционных документов. Во-первых, это наличие гиперссылок, позволяющих пользователю быстро перемещаться по всему документу. Во-вторых, это способность электронных документов включать в себя помимо текстовой информации и статических графических изображений еще и различные виды аудио-визуальной информации в виде аудио и видео роликов, анимации нескольких статических изображений.

В настоящее время наиболее распространенной технологией, использующей подобную форму хранения информации на компьютере, является язык HTML [5].

Т. о., для реализации поставленной задачи в рамках дипломного проектирования был разработан электронный учебник по курсу «Теория вероятностей и математическая статистика» на языке HTML.

Окно учебника вызывается командой меню «Студент\Учебник». При этом открывается следующее окно (рис. 2).

Чтобы перейти к интересующему разделу, достаточно щелкнуть по его названию левой кнопкой мыши и выбрать необходимый параграф.

Кнопки «Предыдущая», «Следующая» и «Обновить» выполняют те же функции, что и аналогичные кнопки браузера.

Чтобы переходить от одного параграфа к другому, от одной темы к другой, необходимо пользоваться кнопками «Предыдущая», «Следующая», располагающимися на страницах учебника.

Кнопка «Вернуться к содержанию» возвращает пользователя к основному содержанию.

Кнопка «Выход» закрывает окно учебника.

Выбрав средство реализации учебника, необходимо было выбрать язык написания самого комплекса. Среди двух альтернатив: языка PHP (PHP + Apache) и среды программирования Borland Delphi 7 (Delphi7+ язык SQL + сервер Microsoft SQL Server) было принято решение использовать последнюю альтернативу[6–8].

Программный комплекс включает в себя следующие основные элементы.

1. Блок преподавателя.

1.1. Создание/редактирование файла группы.

1.2. Просмотр списка группы.

1.3. Просмотр/редактирование списков вопросов теста.

1.4. Просмотр результатов теста.

2. Блок студента.

2.1. Электронный учебник.

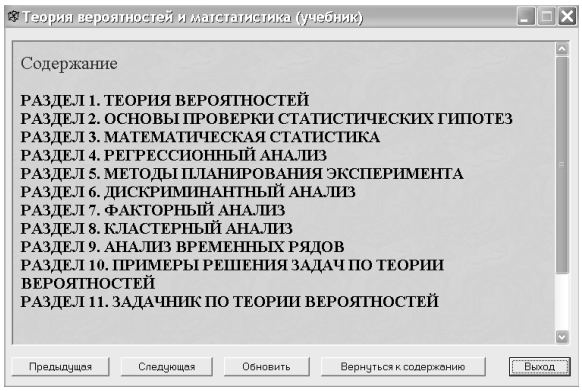

Рис. 2. Окно учебника

2.2. Тестирующая программа.

2.2.1. Выбор фамилии из списка.

2.2.2. Выбор уровня сложности.

2.2.3. Выбор учебной темы для тестирования.

Взаимодействие между блоками системы показано на рис. 3.

В процессе своей работы комплекс использует информационную базу TV_MS.mdf – база Microsoft SQL Server. Для непосредственной работы с данными используются следующие таблицы базы:

- таблицы вопросов к тесту (для каждой темы и каждого уровня сложности отдельно);

- таблица списка студенческой группы (общая для всех групп).

Структура информационной базы данных TV_MS.mdf учебного комплекса

Информационная база программного комплекса содержит двадцать системных таблиц, созданных непосредственно базой данных для хранения системной информации, а также девятнадцать пользовательских таблиц, которые и содержат информацию для тестирования. Остановимся на этих таблицах подробнее.

Восемнадцать из девятнадцати таблиц (TQuestLev11, ..., TQuestLev29) соответствуют девяти учебным темам, подготовленным для тестирования, по два уровня сложности каждая.

Таблицы для тестовых тем имеют следующую структуру:

1 поле – ID (int 4) – порядковый номер вопроса;

2 поле – NumQuest (int 4) – четырехзначный номер вопроса, в котором первые две цифры соответствуют уровню сложности и номеру учебной темы, а последние – номеру вопроса;

3 поле – Question (char 800) – текст вопроса по данной учебной теме;

4 поле – Reply (char 500) – варианты ответов на поставленный;

5 поле – TruthReply (int 4) – номер правильного варианта ответа;

6 поле – Ves (int 4) – числовой вес каждого задания.

Девятнадцатая пользовательская таблица (TGroupStud) предназначена для хранения списка студенческой группы и имеет следующую структуру:

1 поле – ID (numeric 9) – порядковый номер студента;

2 поле – fam (char 30) – фамилия студента;

3 поле – nam (char 20) – имя студента.

Кроме этих девятнадцати основных таблиц во время тестирования для каждого студента создается отдельная таблица, в которую заносятся результаты тестирования. Она имеет следующую структуру:

1 поле – NumQuest (Integer) – четырехзначный номер вопроса;

2 поле – Reply (Integer) – номер правильного варианта ответа;

3 поле – StudName (CHAR 60) – фамилия и имя студента;

4 поле – NumTest (CHAR 2) – номер попытки тестирования;

5 поле ThemeTest (CHAR 60) – последняя тема, по которой проводилось тестирование.

Электронный учебник выполнен в одном окне. Вся теоретическая информация учебника хранится в файлах, написанных на языке HTML, в папке data. Для их отображения используется элемент управления WebBrowser (страница компонентов Delphi7 – Internet).

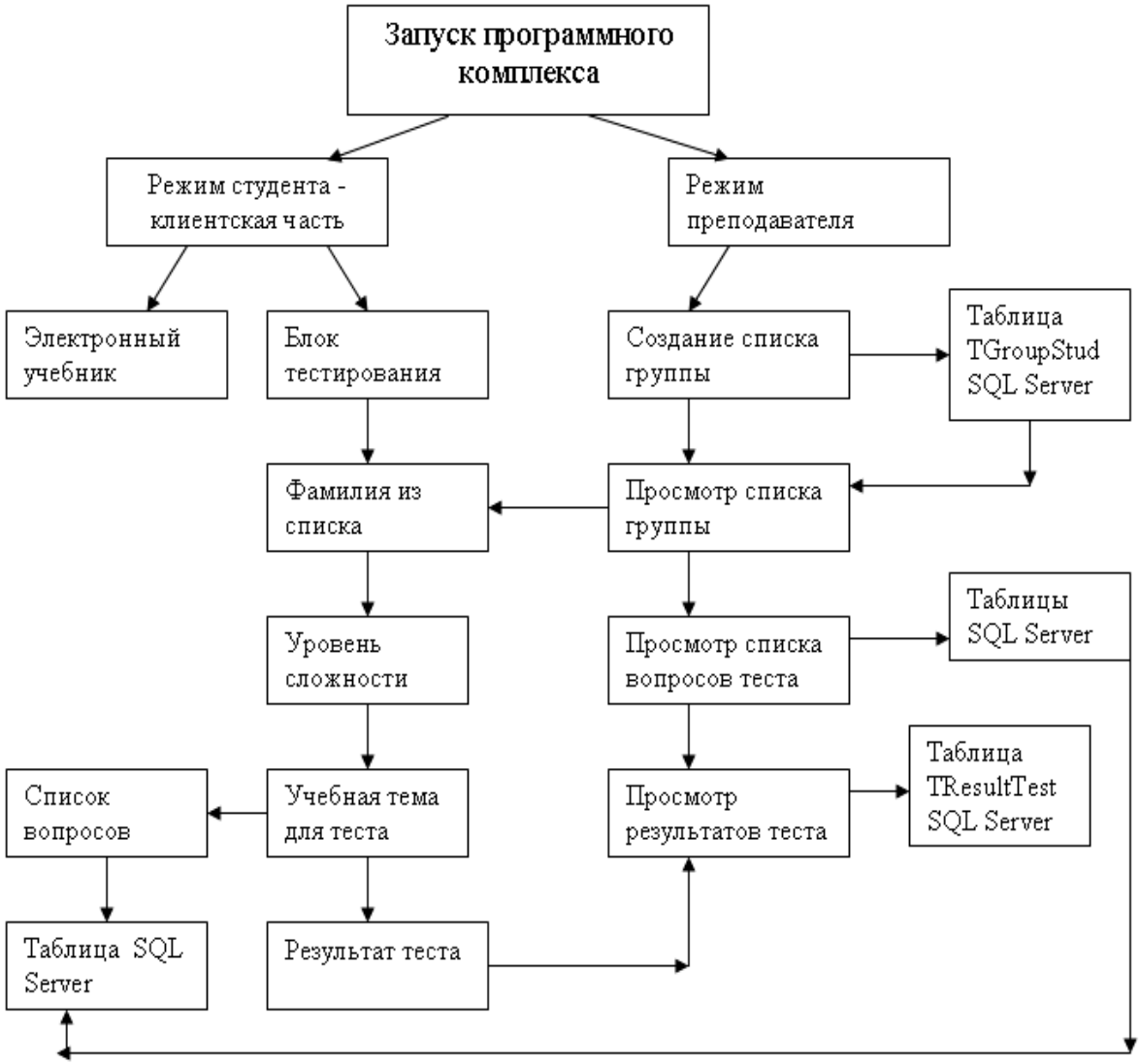

Рис. 3. Взаимодействие блоков программного комплекса

Этот элемент имеет ряд встроенных функций для навигации. При реализации учебника использованы следующие из них:

Navigate – для перехода на страницу, указываемую в качестве параметра.

GoBack – для возврата на предыдущую (до перехода по гиперссылке) страницу.

GoForward – для возврата на следующую (до перехода по гиперссылке) страницу.

Refresh – для обновления содержимого страницы.

Эти функции используются в следующих командах учебника:

- «Вернуться к содержанию» – переход в содержание учебника;
- «Предыдущая» – возврат на предыдущую страницу;
- «Следующая» – возврат на следующую страницу;
- «Обновить» – обновить содержимое текущей страницы учебника.

По команде «Выход» это окно выгружается из памяти, и программа позволяет пользователю перейти к блоку тестирования.

Блок тестирования выполняет следующие функции:

1. Устанавливает соединение с сервером;
2. Получает от сервера информацию о группе;
3. Выводит список группы на экран;
4. Посылает на сервер информацию о пользователе;
5. Обеспечивает выбор уровня сложности теста;
6. Обеспечивает выбор учебной темы для тестирования;
7. Загружает список вопросов теста в соответствии с выбранными учебной темой и уровнем сложности;
8. Определяет результат тестирования;
9. Отправляет результаты тестирования на сервер к преподавателю.

Связь клиентской части программы (режим «Студент») с серверной частью (режим «Преподаватель») и самим сервером (MS SQL Server) осуществляется посредством файла связи с данными connectid.udl и установкой его основных настроек на соответствующих закладках (подробнее см. Руководство пользователя).

Вопросы теста предлагаются только после того, как пользователь (студент) выбрал свои фамилию и имя из списка (CBName:TComboBox), указал номер попытки

тестирования (поле EdtNumTes: TEdit), уровень сложности (RBLev1, RBLev2: TRadioButton) и тему (CBTheme: TComboBox), по которой бы желал пройти тестирование. Связь списка студентов с соответствующей таблицей БД осуществляется посредством компонентов ADOQuery2, DataSource2, обеспечивающими связь с необходимыми данными посредством запросов на языке SQL. Создание персональной таблицы для студента, куда будут записываться данные о ходе и результат тестирования, обеспечивается при нажатии кнопки «Принять введенные личные данные» (BNameOk) посредством компонентов ADOQuery3, DataSource3, обеспечивающими связь с необходимыми данными посредством запросов на языке SQL.

Номера вопросов теста, текст вопросов и варианты ответов к ним отображаются в таблице DBGridQuestion: TDBGrid с помощью компонентов ADOQuery1, DataSource1, обеспечивающими связь с необходимыми данными посредством запросов на языке SQL. Текст вопросов и варианты ответов для удобства пользователя дублируются в полях DBMQuest: TDBMemo, DBMReply:TDBMemo, расположенных под таблицей вопросов.

Ввод номеров вопросов и вариантов ответов на них происходит при нажатии кнопок «Добавить» (BtnAddRec:TButton) и «Вставить запись» (BtnInsertRec:TButton).

Если студент уверен в своих ответах, то при нажатии кнопки «Принять ответы» (BtnReplyOk:TButton) происходит обработка полученных данных и, в зависимости от результата, программа либо переводит пользователя на следующую учебную тему, либо завершает процесс тестирования. Нажатие кнопки «Получить общий результат и завершить тестирование» (BResultTest:TButton), во взаимодействии с компонентами ADOQuery4, DataSource4, обеспечивающими связь с необходимыми табличными данными посредством запросов на языке SQL, дает возможность пользователю увидеть набранное количество баллов и получить отметку, соответствующую данному количеству баллов. Результаты тестирования каждого студента записываются в таблицы БД, которые создаются для каждого студента в отдельности. Выбор своих фамилии и имени студент может сделать только после того, как преподаватель внес в соответствующую таблицу базы список группы студентов, создание и редактирование которого осуществляется при выборе режима «Преподаватель» (блок преподавателя).

ЛИТЕРАТУРА

1. *Мельников А.В., Цытович П.Л.* Принципы построения обучающих систем и их классификация // Педагогические и информационные технологии в образовании. 2001. №4.
2. *Пак Н.И.* Нелинейные технологии обучения в условиях информатизации: учеб. пособие. Красноярск: Изд-во РИО КГПУ, 1999.
3. *Попов Э.В.* Экспертные системы: решение неформализованных задач в диалоге с ЭВМ. М.: Наука, 1987. 288 с.
4. *Бояринов А.И., Кафаров В.В.* Методы оптимизации в химической технологии. М.: Химия, 1975. С. 257–268.
5. *Булгаков С.В.* Анализ инструментальных средств разработки мультиагентных систем // Тезисы докладов междунар. конф. молодых ученых по математическому моделированию и информационным технологиям. Новосибирск, 29–31 окт. 2002 г. Новосибирск, 2002.
6. *Архангельский А.Я.* Программирование в Delphi 7. М.: Бином-Пресс, 2003. С. 551–708.
7. *Бобровский С.И.* Delphi 7: учебный курс. СПб.: Питер, 2005. С. 363–383.
8. *Гофман В.Э., Хомоненко А.Д.* Работа с базами данных в Delphi. СПб.: БХВ-Петербург, 2002. С. 41–403.
9. *Кларин М.В.* Инновации в обучении. М.: Наука, 1997. 224 с.

Поступила в редакцию 24 декабря 2007 г.

ОБУЧАЮЩИЕ СВОЙСТВА ПРОЦЕССА РАЗРАБОТКИ ТЕСТОВЫХ ЗАДАНИЙ

© В.В. Зубец

В последнее десятилетие в системе российского образования все более широко применяются тестовые задания. Прежде всего, они используются как средства объективного контроля знаний; реже – как средство обучения – тестирование с указанием ошибок и правильных ответов. Между тем, сам процесс разработки тестовых заданий в определенной предметной области несет в себе полезные педагогические свойства.

Традиционными формами практических занятий по «описательным» дисциплинам являются либо пересказ студентом содержания лекции или учебника, либо подготовка реферата. Эти формы плохи своей пассивностью, механическим заучиванием материала. Если раньше реферат переписывался от руки и студент хотя бы знал его содержание, то теперь, в эпоху Интернета, «автор» реферата часто его даже и не читает.

С целью повышения активности в изучении материала курса «Государственные стандарты РФ в области информационных технологий» автор предлагал студентам разрабатывать тестовые задания на определенные темы. При этом студенты должны были проанализировать материал, выявить главное, определить связи между понятиями, т. е. построить определенную модель предметной области. На основе этой модели разрабатывались тесты. Далее преподаватель вместе с группой обсуждал их качество. При этом рассматривались такие свойства, как содержательность, валидность, краткость и однозначность, трудность и др. После отбора и редактирования лучших заданий, они кодировались для использования на компьютере, и проводилось тестирование студентов.

В результате проведенного педагогического эксперимента можно сделать следующие выводы:

- активность студентов на занятии заметно выросла, поскольку в обсуждении участвовали, по крайней мере, несколько человек;

- большинство студентов верно выявляли особенности структуры предметной области, что отразилось в тестах;