

А.В.Гуща

**СИСТЕМА ТЕСТОВОГО КОНТРОЛЯ ЗНАНИЙ НА
ОСНОВЕ КОГНЕМНОЙ МОДЕЛИ (СКЗ-КМ):
МАКЕТНАЯ ВЕРСИЯ 2012 г.¹**

СОДЕРЖАНИЕ

Введение

- I. Описание системы СКЗ-КМ.
- II. Обзор способов и методов интеллектуального контроля знаний.
- III. Методика интеллектуального контроля знаний на основе модели представления вербальных языковых знаний тексты основных программ.
- IV. Тексты основных программ.
Заключение.
Список использованной литературы.

ВВЕДЕНИЕ

На сегодняшний день в России, как и во всем мире, стремительно развиваются информационные технологии, в том числе в сфере обучения и контроля знаний. Современное состояние науки и образования в нашей стране, обусловленное экономическими факторами, характеризуется повышением требований к качеству подготовки специалистов, и определяет постоянный поиск новых методов и средств повышения эффективности образовательного процесса.

В последние годы в этой сфере наблюдается стремительное усиление интереса к автоматизации промежуточного и финального контроля результатов обучения учащихся самых различных учебных заведений, начиная от школ и заканчивая коммерческими курсами. Самым популярным видом такого контроля является тестирование. Стремительный рост быстродействия компьютерных систем, уменьшение цен на вычислительную технику, появление качественных и мощных систем программирования увеличило потребность в системах, позволяющих объективно, быстро и надежно оценивать знания учащихся, предлагая интересные формы взаимодействия с ними.

Применение современных информационных технологий в образовании позволяет:

- индивидуализировать подход и дифференцировать процесс обучения;
- контролировать обучаемого с диагностикой ошибок и обратной связью;
- обеспечить самоконтроль учебно-познавательной деятельности;

¹ В данном материале приводятся результаты исследований, выполненных при поддержке гранта РГНФ №12-04-12039в

- сократить время обучения за счет трудоемких вычислений на компьютере;

Новая универсальная форма обучения – это дистанционное образование, которое базируется на использовании широкого спектра как традиционных, так и новых информационных и телекоммуникационных технологий и технических средств. При этом процесс обучения и контроля его результатов не зависит от местоположения студента и очень гибок во времени.

В настоящей работе предложена реализация модели и методики интеллектуального контроля знаний, основанных на представлении вербальных языковых знаний. На основе этой модели разработана информационная система контроля знаний.

Описание системы СКЗ-КМ

1. Общее описание системы

1.1. Цель, назначение и область применения разработки

Целью разработки системы является повышение эффективности процесса обучения, реализация методики контроля теоретических знаний, применимой в различных областях приложения.

Система предназначена для создания, применения и оценки качества учебных тестов, которые представляют собой набор тестовых заданий открытого типа специфической формы.

При разработке данного программного продукта была поставлена задача использования оригинальной информационной модели тестового задания и базы тестовых заданий, основанной на модели представления вербальных языковых знаний, предложенной Ю.Н. Карауловым (Рис.1.). В этом состоит основное отличие данного программного продукта от остальных.

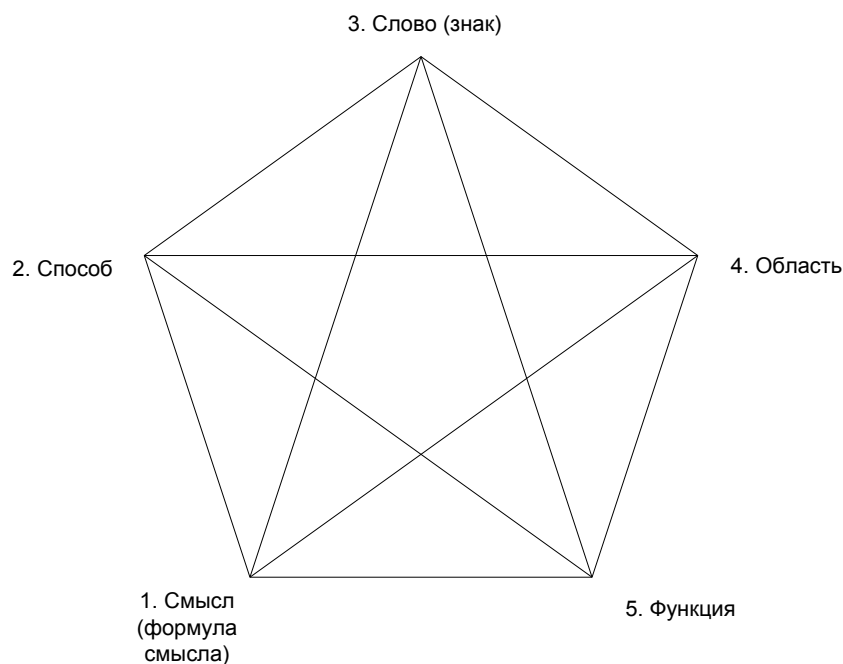


Рис. 1. Схематическое изображение фигуры знания

В качестве модели процесса тестирования выбрана самая простая модель – это последовательное предъявление вопросов в том порядке, который задан разработчиком теста.

В качестве модели обработки результатов как основная была выбрана однопараметрическая модель Раша. Это классическая модель теории IRT, которая обладает рядом важных достоинств:

- модель Раша превращает измерения, сделанные в дихотомических и порядковых шкалах в линейные измерения, в результате качественные данные анализируются с помощью количественных методов;
- поскольку мера измерения параметров модели Раша является линейной, то это позволяет использовать широкий спектр статистических процедур для анализа результатов измерений;
- оценка трудности тестовых заданий не зависит от выборки испытуемых, на которых была получена;
- оценка уровня знаний испытуемых не зависит от используемого набора тестовых заданий;
- пропуск данных для некоторых комбинаций (испытуемый – тестовое задание) не является критическим.

Для анализа качества теста и тестовых заданий реализованы как основные процедуры классической теории тестирования, так и анализ в рамках модели Раша.

Программный продукт позволяет пользователю решать следующие задачи:

1. Проведение тестирования, как индивидуального, так и массового, с высоким уровнем масштабируемости и защиты от фальсификации результатов тестирования.
2. Аналитическая обработка результатов тестирования, с использованием представления, аналогичного модели данных, позволяющая устанавливать характер зависимостей между результатами тестирования и параметрами модели тестового задания.
3. Оценка качества тестов как в соответствии с классической теорией тестирования, так и в соответствии с теорией IRT.
4. Проведение стандартизации теста и установки норм на основании интерпретации данных обработки результатов тестирования.
5. Получение объективированных оценок уровня подготовки испытуемых и уровня трудности заданий в единой интервальной шкале логитов, в соответствии с моделью Раша, численным методом с заданной точностью.
6. Вычисление отметок испытуемых на основании процента верно выполненных заданий и норм теста, в случае, когда не используется модель Раша.

1.2. Функции программного продукта

Основные функции программного продукта:

1. Поддержка ведения базы знаний/данных о тестовых заданиях (БТЗ), которая предусматривает организацию ввода новых заданий, удаления и изменения заданий.
2. Организация формирования (отбора содержания) тестов на основе БТЗ, определения порядка предъявления задания в тесте, редактирования и удаления тестов.
3. Автоматизация проведения тестирования, как индивидуального, так и массового, с высоким уровнем масштабируемости и защиты от фальсификации результатов тестирования.
4. Формирование отчетов по оценке качества тестов как в соответствии с классической теорией тестирования, так и в соответствии с теорией IRT.
5. Расчет оценок уровня подготовки испытуемых и уровня трудности заданий в единой интервальной шкале логитов, в соответствии с моделью Раша.
6. Формирование отчетов по модели Раша, содержащих итоговые и промежуточные значения основных переменных, участвующих в расчете.
7. Вычисление отметок испытуемых на основании процента верно выполненных заданий и норм теста, в случае, когда не используется модель Раша.
8. Представление тестов в виде языковой игры: кроссворда.

1.3. Требования к составу программных и технических средств

Для нормального функционирования системы на сервере должно быть установлено:

Windows 2000/XP; Microsoft Word 2000/XP; Microsoft Excel 2000/XP; Apache HTTP server; MySQL 5; PHP 5.

Для нормального функционирования рабочего места студента, преподавателя, администратора на компьютере должно быть установлено: Windows 2000/XP; Microsoft Internet Explorer версии 5.0 или выше.

Требования к оборудованию для компьютера-клиента:

Процессор: PentiumII-500 и выше;
Память: 128 Mb RAM;
Жесткий диск: 1 Гб;
Видеоадаптер: SVGA адаптер с поддержкой разрешения 1024x768;
Монитор: SVGA 1024x768x60Гц и выше;
Клавиатура: любая клавиатура, совместимая с MS Windows;
Мышь: любая мышь, совместимая с MS Windows;
Сетевая карта: на 100 Мбит или больше.

Требования к оборудованию компьютера-сервера:

Процессор: Pentium IV;
Оперативная память: 1024 Mb;
Жесткий диск: HDD 80 Gb;
Монитор: SVGA 1024x768x60Гц и выше;
Сетевая карта: на 100 Мбит или больше;
Клавиатура: любая, работа которой поддерживается средствами ОС;
Мышь: любая, работа которой поддерживается средствами ОС.

2. Архитектура системы

2.1 Модель базы данных

Для управления базами данных была выбрана СУБД MySQL, это связано с тем, что язык PHP, на котором написана система, имеет удобный и доступный интерфейс для работы с этой СУБД. Кроме того эта СУБД проста в установке и очень широко распространена.

База данных содержит 7 таблиц. Ниже перечислены все таблицы и соответствующие им атрибуты, с кратким описанием их назначения.

1. Таблица **Users** – Пользователи.

ID	Уникальный ключ
Login	Логин

Password	Пароль
Name	Имя
Surname	Фамилия
Patronymic	Отчество
Chairs	Кафедра
Years	Года обучения
Role	Статус (учитель, ученик, администратор)

2. Таблица **TestItems** – Тестовые задания.

ID	Уникальный ключ
Form	Формула смысла
Answer	Множество знаков
Type	Способ задания
DidacticUnits	Дидактическая единица
Func	Функция (ретушь, рецепт)
Test	Тест

3. Таблица **Tests** – Тесты.

ID	Уникальный ключ
Name	Имя
Items	Тестовые задания
Didact	Дидактическая единица
ItemsNum	Количество заданий

4. Таблица **TestingItems** – элементы тестирования.

ID	Уникальный ключ
User	Пользователь
Answers	Ответы на задания
Point	Общий бал

5. Таблица **Testing** – тестирование.

ID	Уникальный ключ
----	-----------------

Name	Имя
Descr	Описание
Test	Тест
Items	Элементы тестирования
TaskPoints	Балы по каждому заданию
AllPoints	Общее количество баллов
DateBegin	Дата начала
TimeBegin	Время начала
DateEnd	Дата конца
TimeEnd	Время конца

6. Таблица **Crosswords** – кроссворды.

ID	Уникальный ключ
Name	Имя
DidacticUnits	Дидактическая единица
Description	Описание
Width	Ширина в клетках
Height	Высота в клетках
Words	Элементы кроссворда

7. Таблица **CrosswordItems** – элементы кроссворда.

ID	Уникальный ключ
Crossword	Кроссворд
Question	Вопрос
Answer	Ответ
X	Координата X начала
Y	Координата Y начала
Position	Позиция (гориз. или верт.)

2.2. Структура системы

Система состоит из трех автоматизированных рабочих мест – Администратор, Преподаватель и Студент. Все АРМ используют трехзвенную архитектуру «клиент-Web-сервер-сервер базы данных», так называемый «тонкий» клиент.

Такое разделение было сделано исходя из того, что массовое тестирование студентов требует именно «тонкого» клиента, когда на рабочей станции не нужно устанавливать никакого специального программного обеспечения. При использовании «тонкого» клиента систему надо установить только на 1-2 машине – Web-сервер, сервер базы данных.

Языком разработки был выбран серверный язык программирования – PHP. Это выбор был обоснован большой популярностью этого языка, его поддерживают большинство современных Web-серверов.

Схематически физическая структура системы изображена на рисунке 2.

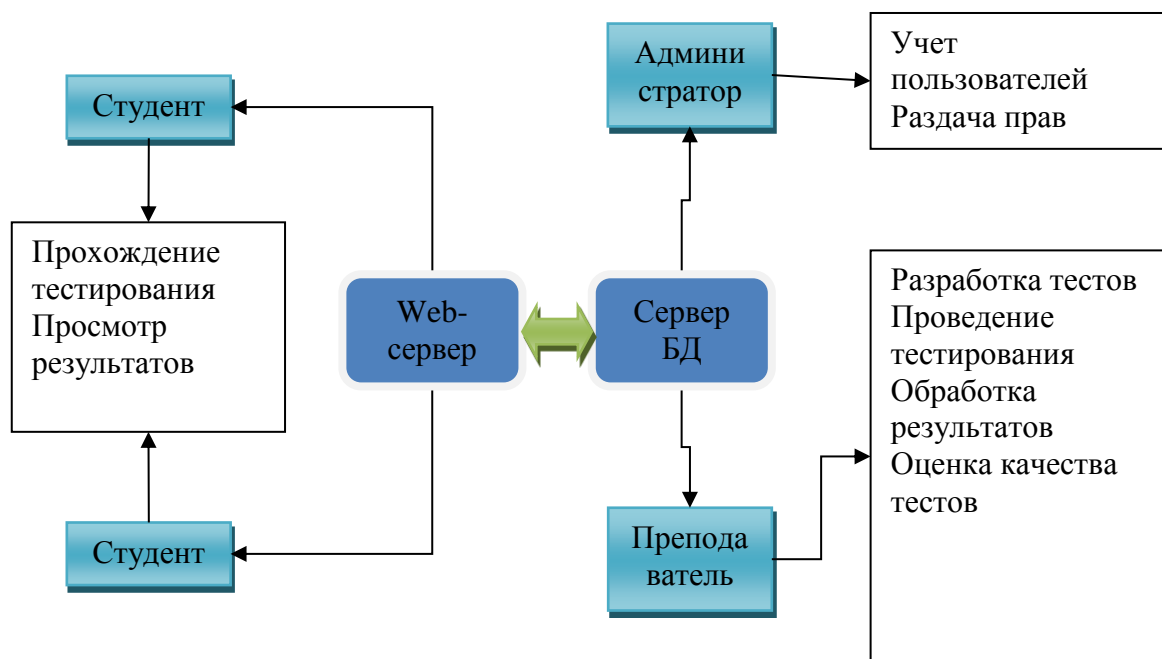
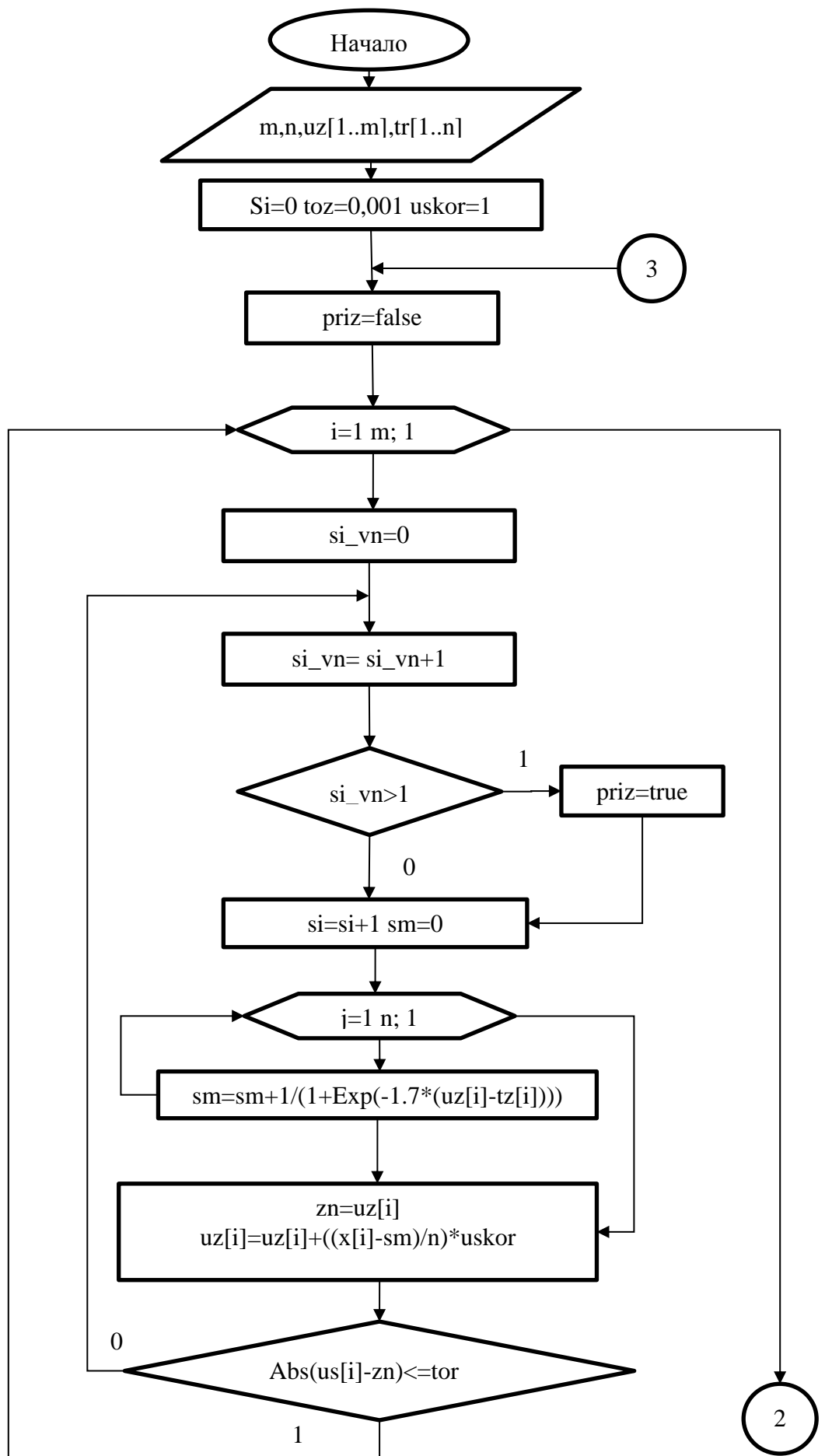


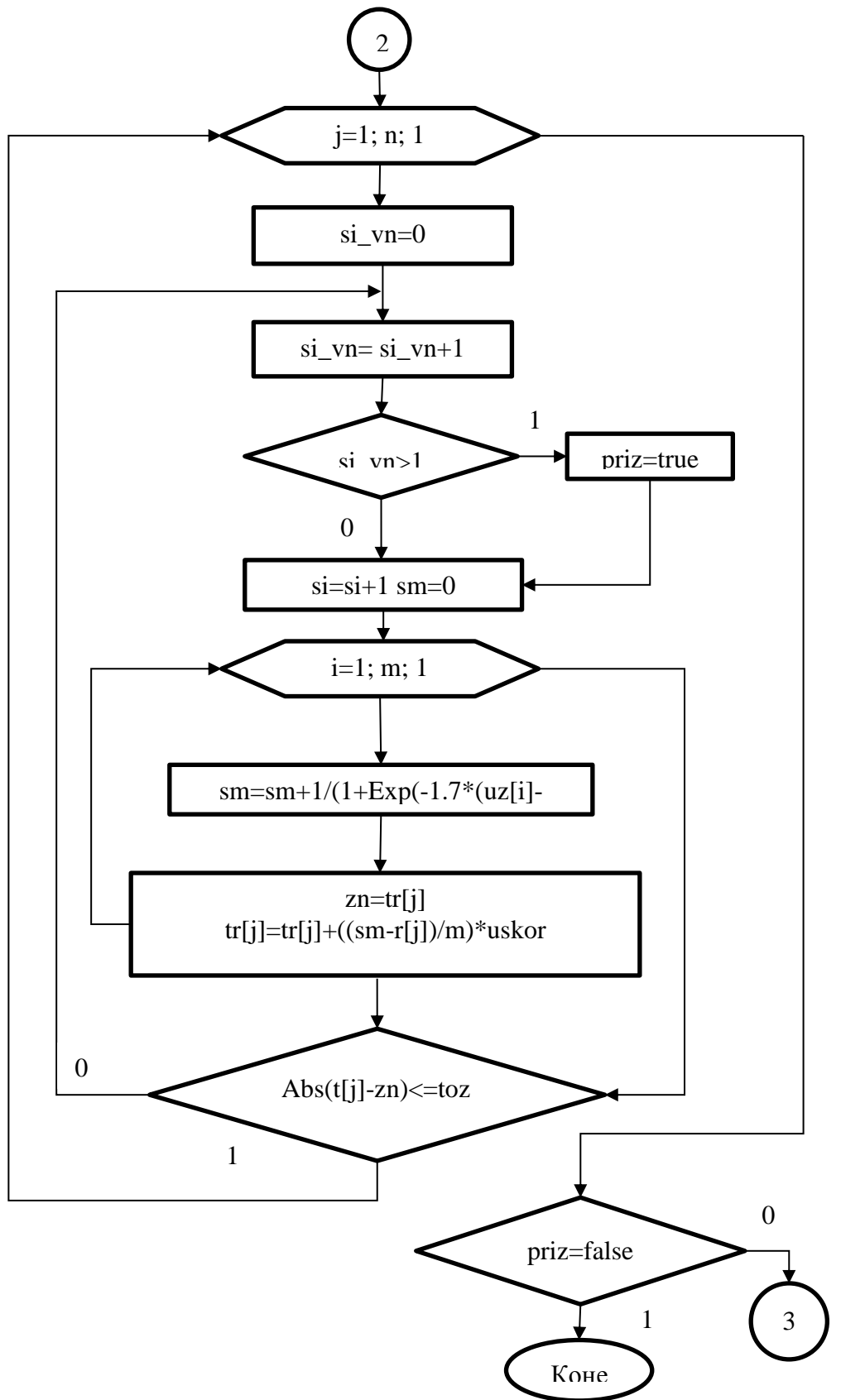
Рис 2. Структура системы

Система ориентирована в равной степени на использование Интернет и Интранет архитектуры. При тестировании через Интернет остается проблема фальсификации результатов за счет прохождения тестирования другим пользователем.

2.3. Основные алгоритмы

- алгоритм функции оценивания тестового задания;
- алгоритм проведения тестирования
- алгоритм расчета оценок уровня подготовки испытуемых и уровня трудности задания для модели Раша итерационным методом (метод Райта). рис. 3.





-
-
-

▪ Рис.3 Получение логитов из сырого бала (метод Райта)

3. Основные формы интерфейса пользователя

3.1. Автоматизированное рабочее место преподавателя

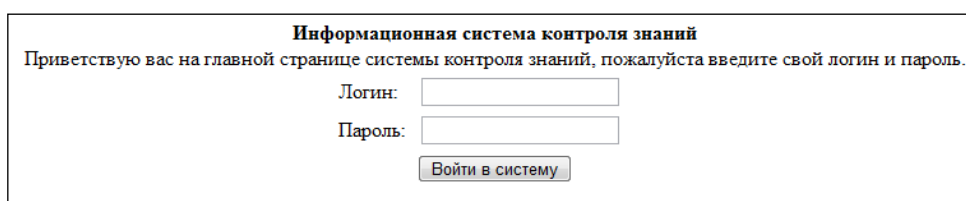
АРМ «Преподаватель» предназначено для разработки тестовых заданий и тестов, работы с базой тестовых заданий, управления процессом тестирования, обработки полученных эмпирических данных тестирования, получения различных форм отчетности.

Пользователями системы являются преподаватели, для каждого из них администратор системы должен предоставить имя и пароль пользователя для входа в систему.

Рассмотрим основные компоненты интерфейса пользователя АРМ «Преподаватель».

Вход в АРМ «Преподаватель»

Для входа в программу нужно зайти в браузере на адрес размещения системы. После этого на экране появится форма для ввода имени пользователя и пароля. Рис.4



The screenshot shows a login form titled "Информационная система контроля знаний". Below the title, it says "Приветствую вас на главной странице системы контроля знаний, пожалуйста введите свой логин и пароль." There are two input fields: "Логин:" and "Пароль:". Below the password field is a button labeled "Войти в систему".

Рис.4 Форма входа в систему

После входа в систему открывается главная страница, на которой отображено текущее состояние системы. Рис.5

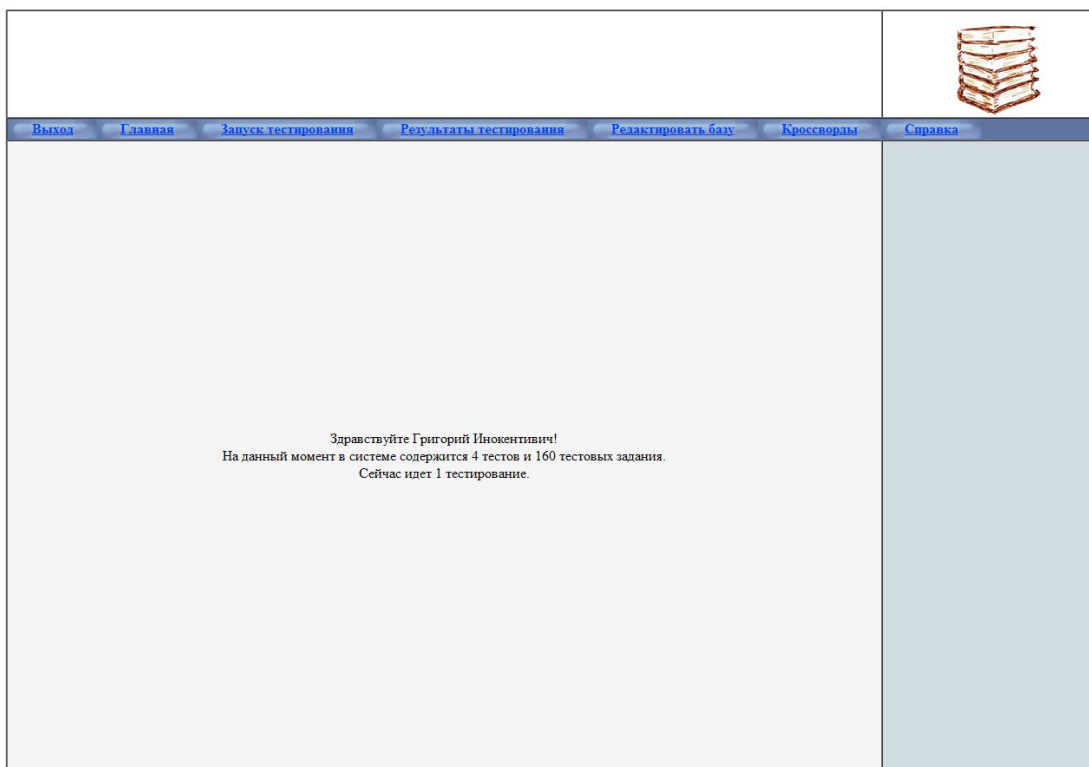


Рис. 5 Главная страница системы. АРМ «Преподаватель»


Структура главного меню АРМ «Преподаватель»

Главное меню имеет следующую структуру:

1. «Выход» – позволяет зайти в систему заново под другим логином или паролем. При завершении работы советуется выходить из системы, чтобы минимизировать риск взлома.
2. «Главная» - переводит на главную страницу системы, где отображается информация о текущем состоянии системы.
3. «Запуск тестирования» - страница, на которой можно запустить тестирование, задать его параметры: время начала и конца, тест, описание.
4. «Результаты тестирования»- формирование отчетов по результатам тестирования как в соответствии с классической теорией тестирования, так и в соответствии с теорией IRT.
5. «Редактировать базу » - редактирование базы данных тестовых заданий и тестов.
6. «Кроссворды» - редактирование, создание кроссвордов и просмотр результатов их прохождения.
7. «Справка» - документация к системе, содержащая описание основных функций программного продукта и используемых формул.


Запуск тестирования АРМ «Преподаватель»

Рассмотрим, как происходит запуск тестирования. В день тестирования преподаватель запускает свое рабочее место, открывает страницу «Запуск тестирования», выбирает в этой форме тест и указывает время, дату начала и конца тестирования, и нажимает кнопку «Запустить». После этого студенты могут подключаться к базе с помощью браузера, то есть своего рабочего места, и проходить тестирование. После окончания срока тестирования или после прохождения теста всеми студентами можно сформировать результаты тестирования. Вид страницы представлен на рис.6-7.



Выход Главная Запуск тестирования Результаты тестирования Редактировать базу Кроссворды Справка

Номер	Название	Дидактические единицы	Действия
1	Тестовый1	Основы информатики	<input type="button" value="Запустить"/>
2	Тестовый2	Основы информатики	<input type="button" value="Запустить"/>
3	Тестовый3	Основы информатики	<input type="button" value="Запустить"/>
4	Тестовый4	Основы информатики	<input type="button" value="Запустить"/>



Выход Главная Запуск тестирования Результаты тестирования Редактировать базу Кроссворды Справка

Тест выбран

Название	Тестовый1
Дидакт. единица	Основы информатики
Заданий	40

Введите параметры тестирования

Название тестирования:

Описание:

Время начала (час:минуты:секунды):

Дата начала (год-месяц-день):

Время конца (час:минуты:секунды):

Дата конца (год-месяц-день):

Рис.6-7 Запуск тестирования

Просмотр результатов тестирования

После окончания тестирования на этой странице преподавателю предоставляются результат тестирования. В него входят: получение объективированных оценок уровня подготовки испытуемых и уровня трудности заданий в единой интервальной шкале логитов, в соответствии с моделью Раша, численным методом с заданной точность, вычисление отметок испытуемых на основании процента верно выполненных заданий и норм теста, в случае, когда не используется модель Раша. Также на этой странице формируются отчеты по оценке качества тестов как в соответствии с классической теорией тестирования, так и в соответствии с теорией IRT.

Редактирование базы данных

Система имеет встроенный интерфейс для редактирования базы данных тестовых заданий и тестов. Преподаватель может создавать, изменять и удалять тест, изменять его состав и порядок предъявления тестовых заданий. Также можно изменять, создавать, удалять тестовые задания, которые привязаны к конкретному тесту. Список тестовых заданий можно отсортировать по формуле смысла, знаку, функции, способу задания или по дидактическим единицам.

Редактирование кроссвордов

Система позволяет представлять тесты в виде языковой игры. На этой странице проводится верстка кроссвордов, задается их визуальное отображение. Также здесь можно редактировать базу данных, переводить тестовые задания в элементы кроссворда. Здесь же можно просмотреть сколько кроссвордов выполнил каждый студент и сколько слов он отгадал.

Справочный материал

Эта страница содержит справочный материал по теоретической основе, используемым формулам и основным функциям данного программного продукта. Навигация между главами справки осуществляется с помощью боковой панели, а текст отображается в основной части страницы. Вид страницы представлен на рис.8.

Основные расчетные формулы

Функция IRF для однопараметрической модели Раша:

$$P_j(\theta) = \frac{e^{1,7(\theta - \beta_j)}}{1 + e^{1,7(\theta - \beta_j)}}$$

$$P_i(\beta) = \frac{e^{1,7(\theta_i - \beta)}}{1 + e^{1,7(\theta_i - \beta)}}$$

где P – вероятность правильного ответа, сигма – латентный параметр, определяющий уровень подготовки испытуемого, бета – латентный параметр, определяющий уровень трудности задания.

Начальные значения параметров:

$$\theta_i^0 = \ln \frac{P_i}{q_i}$$

$$\beta_j^0 = \ln \frac{q_j}{p_j}$$

где pi и qi – доли правильных и неправильных соответственно ответов i-го ученика на задания теста, pj и qj – доли правильных и неправильных ответов учеников на j-е задание теста. При этом число испытуемых равно N, а число заданий равно n.

Оценки параметров в единичной интервальной шкале логитов:

$$\theta = \bar{R} + \chi\theta^0$$

Рис.8 Справочный материал

3.2. Автоматизированное рабочее место студента

АРМ «Студент» предназначен для прохождения тестов и кроссвордов и просмотра результата тестирования. Каждая страница этого рабочего места имеет очень простой интерфейс.

Вход в АРМ «Студент»

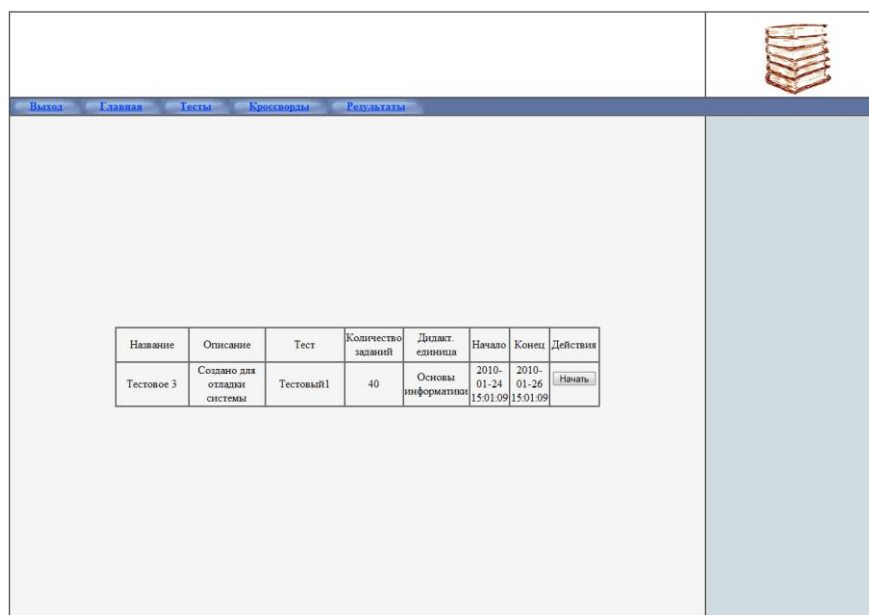
Вход производится точно так же, как и для АРМ «Преподаватель». Рис. 4.

Структура главного меню АРМ «Студент»:

1. «Выход» – позволяет зайти в систему заново под другим логином или паролем.
2. «Главная» - переводит на главную страницу системы, где отображается информация о текущем состоянии системы.
3. «Тесты» - список доступных к прохождению тестов
4. «Кроссворды» - список доступных к прохождению кроссвордов.
5. «Результаты» - страница, где отображаются результаты прохождения тестов.

Страница выбора тестирования:

Перед прохождением теста студент попадает на страницу со списком открытых тестирований, где отображается название теста, дата начала и конца тестирования, описание тестирования. Студент нажимает на кнопку «пройти» рядом с соответствующим тестированием и его направляют на страницу прохождения тестирования. Вид страницы показан на рис. 9.



Название	Описание	Тест	Количество заданий	Дидакт. единица	Начало	Конец	Действия
Тестовое 3	Создано для отладки системы	Тестовый1	40	Основы информатики	2010-01-24 15:01:09	2010-01-26 15:01:09	<input type="button" value="Начать"/>

Рис. 9. Страница выбора тестирования

Страница прохождения теста:

Студенту последовательно предъявляются вопросы тестовых заданий теста с формой для ответа. По окончании тестирования сообщается количество правильных ответов. Вид страницы представлен на рис. 10.

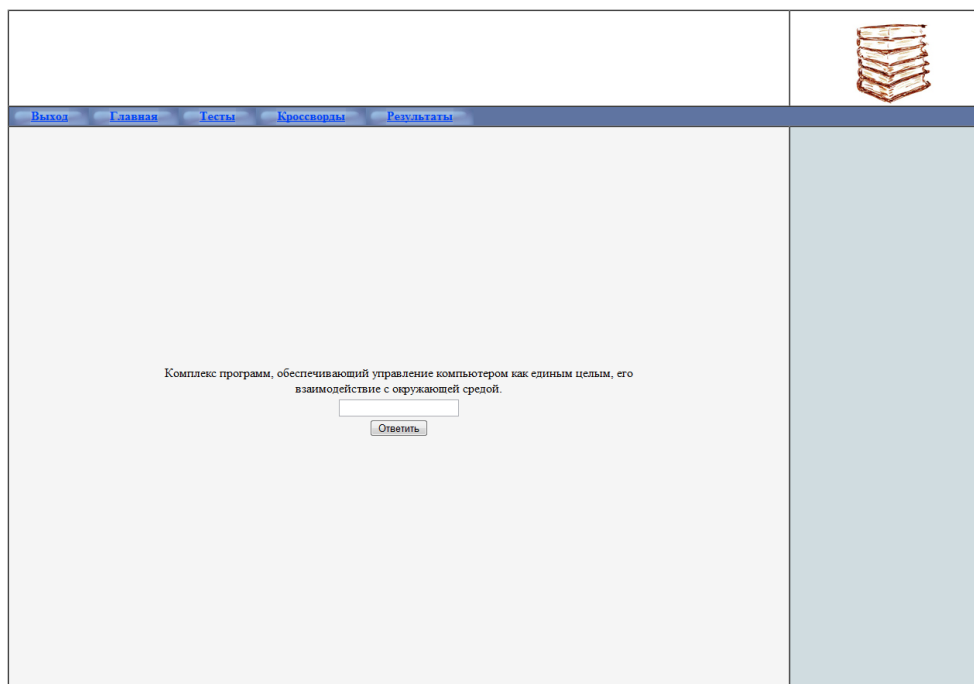


Рис.10 Прохождение теста

Страница выбора и прохождения кроссворда:

Студент из списка доступных кроссвордов выбирает нужный и нажимает на кнопку «начать». Ему предоставляется матрица кроссворда в главной части окна и список вопросов в дополнительной панели. После нажатия кнопки проверить правильные ответы помечаются зеленым цветом, а студенту предоставляется возможность скорректировать неправильные ответы.

Страница просмотра результатов:

На этой странице студенту предоставляется статистика прохождений тестов и их результаты.

3.3. Автоматизированное рабочее место администратора

АРМ «Администратор» предназначен для учета и раздачи прав пользователям.

Структура главного меню АРМ «Администратор»:

1. «Выход» – позволяет зайти в систему заново под другим логином или паролем.
2. «Главная» - переводит на главную страницу системы, где отображается информация о текущем состоянии системы.
3. «Пользователи» - страница учета и редактирования прав пользователей.

Страница учета пользователей:

В дополнительной панели отображаются отсортированные списки преподавателей и студентов и кнопки для удаления и правки. В главной части страницы отображаются

дополнительные действия: создание нового пользователя. Вид страницы показан на рис. 11.

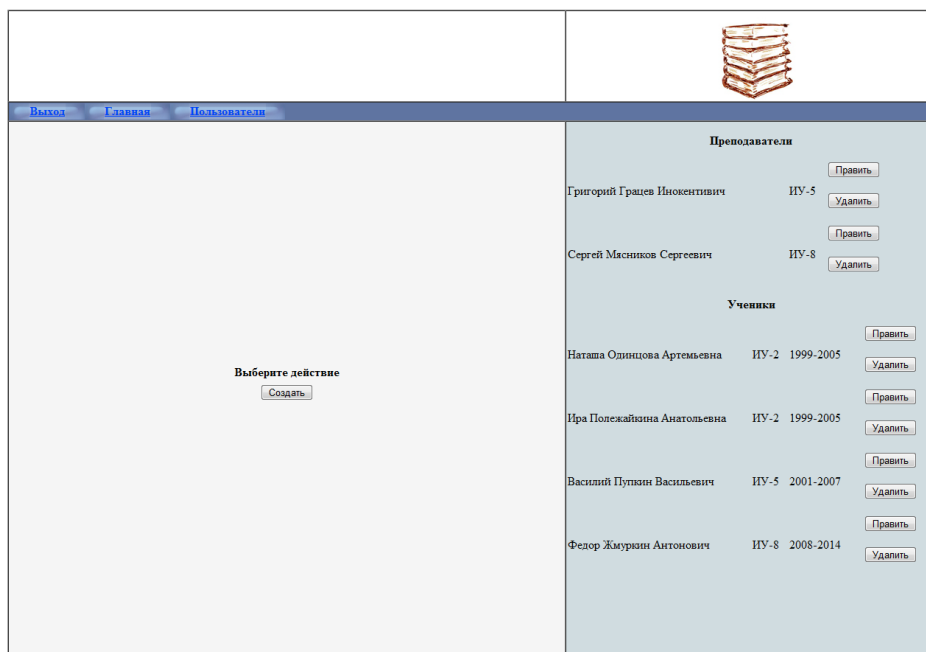


Рис 11. Страница учета пользователей

Страница создания нового пользователя:

На этой странице можно создать нового пользователя, заполнив необходимые поля: логин, пароль, статус и т.д. Вид страницы показан на рис. 12

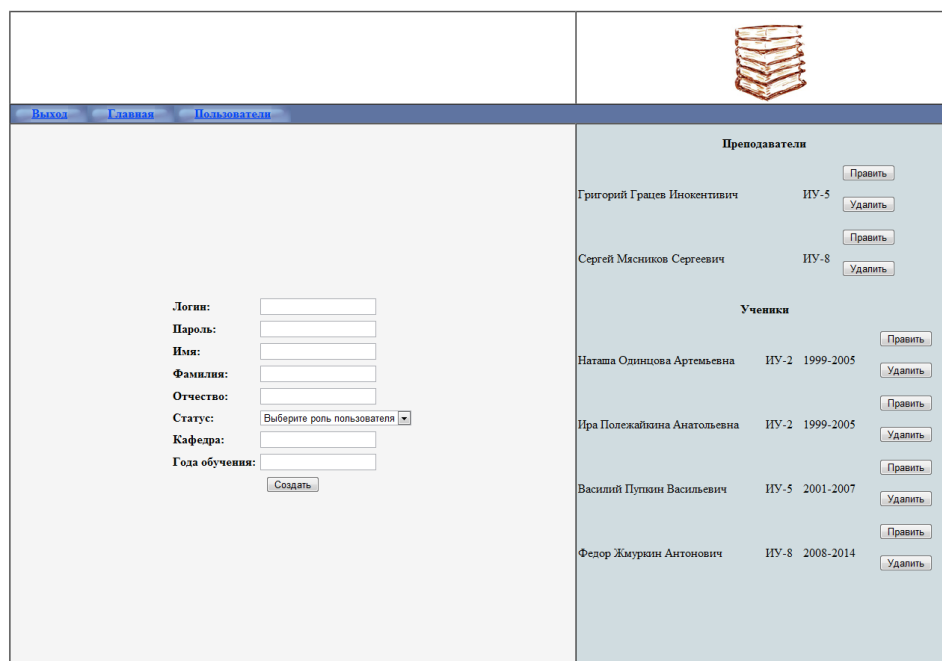


Рис. 12 Создание нового пользователя

II. ОБЗОР СПОСОБОВ И МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО КОНТРОЛЯ ЗНАНИЙ.

1. Педагогический контроль и педагогическая диагностика

1.1. Понятия и определения процесса обучения и контроля знаний

Определения даны из педагогического энциклопедического словаря под редакцией Б.М. Бим-Бада (далее ПЭС) [1].

Обучение – это совместная целенаправленная деятельность учителя и учащихся, в ходе которой осуществляется развитие личности, её образование и воспитание. Понятие обучения многозначно, а свойство обучаться принадлежит к числу фундаментальных свойств всех живых систем. Главными составляющими процесса обучения являются преподавание и учение, в единстве которых заключается основная дидактическая функция обучения.

Проверка и оценка знаний, умений и навыков учащихся – это процесс выявления и сравнения на том или ином этапе обучения результатов учебной деятельности с требованиями, задаваемыми учебными программами; выражается в форме отметки или словесного (оценочного) суждения учителя.

Педагогический контроль – наблюдения в целях надзора за воспитательным процессом, процессом обучения и выявления отклонений от заданной цели и их причин [3].

Контроль – это определенная система взаимодействия двух субъектов: деятельность одного субъекта (преподавателя), который организует деятельность другого субъекта (студента), в процессе и в результате которой выявляются результаты учебно-познавательной деятельности студента, и их соответствие требованиям, задаваемым учебными программами.

Педагогическая диагностика – это совокупность приёмов контроля и оценки, направленных на решение задач оптимизации учебного процесса, дифференциации учащихся, а также совершенствования образовательных программ и методов педагогического воздействия.

Цели обучения определяют **нормативную модель обучаемого** (какими должны быть знания, умения, навыки, качества обучаемого). **Текущая модель обучаемого** – это то, каков он есть в данный момент (его текущие знания, умения, навыки, качества) [4].

Педагогический тест - система заданий возрастающей трудности, специфической формы, позволяющей качественно и эффективно измерить уровень и оценить структуру подготовленности учащихся.

1.2. Функции педагогического контроля

Контролирующая функция. Контролирующая функция состоит в выявлении состояния знаний и умений учащихся, уровня их умственного развития, в изучении степени усвоения приемов познавательной деятельности, навыков рационального учебного труда [2].

Диагностическая функция. Сущность диагностической функции контроля – в получении информации об ошибках, недочетах и пробелах в знаниях и умениях учащихся и порождающих их причинах затруднений учащихся в овладении учебным материалом, о числе, характере ошибок. Результаты диагностических проверок помогают выбрать наиболее интенсивную методику обучения, а также уточнить направление дальнейшего совершенствования содержания методов и средств обучения.

Обучающая функция связана с актуализацией приобретенных знаний, умений и навыков, осуществляется в ходе повторения, уточнения и обобщения, имеющих место в ходе процедур контроля, кроме того, реализация данной функции способствует углублению, расширению и прочному усвоению знаний, умений и навыков, установлению меж- и внутрипредметных связей [2].

Прогностическая функция проверки служит получению опережающей информации об учебно-воспитательном процессе. В результате проверки получают основания для прогноза о ходе определенного отрезка учебного процесса: достаточно ли сформированы конкретные знания, умения и навыки для усвоения последующей порции учебного материала (раздела, темы).

Ориентирующая функция предполагает получение объективной информации для дальнейшего планирования учебного процесса с учетом возможностей учеников.

Развивающая функция контроля состоит в стимулировании познавательной активности учащихся, в развитии их творческих способностей. Контроль обладает исключительными возможностями в развитии учащихся. В процессе контроля развиваются речь, память, внимание, воображение, воля и мышление школьников. Контроль оказывает большое влияние на развитие и проявление таких качеств личности, как способности, склонности, интересы, потребности.

Воспитывающая функция контроля состоит в воспитании у учащихся ответственного отношения к учению, дисциплины, аккуратности, честности. Проверка побуждает более серьезно и регулярно контролировать себя при

выполнении заданий. Она является условием воспитания твердой воли, настойчивости, привычки к регулярному труду [2].

1.3. Средства контроля

Безмашинные средства контроля [2]

Проверка домашнего задания. Роль домашних заданий практически обесценивается, если не налажена их проверка. Преподаватели практикуют разные формы учета. Это и устный опрос у доски или с места по домашнему заданию, и короткая письменная работа, но, прежде всего это непосредственная проверка задания в тетрадях – фронтальная при обходе класса в начале и более основательная, выборочная во внеурочное время.

Диктант. Диктант может заменить опрос по теме, заданной для повторения. Его продолжительность обычно 10-20 минут. Он представляет собой систему вопросов, связанных между собой.

Организация самостоятельных работ. При изучении важно, чтобы учащиеся не только знали теоретический материал, но и умели применять его к решению задач и упражнений, обладали бы рядом навыков. Эти умения и навыки могут быть по настоящему проверены только в письменной работе. Обычно самостоятельные работы проводятся после коллективного решения задач новой темы и предшествуют контрольной работе по этой теме.

Организация контрольных работ. Обычно проходит в письменном виде и на уроке и этим отличается от домашней работы. В ходе контрольной работы ученики обычно не имеют права пользоваться учебниками, конспектами и т. п. После серии контрольных работ и ответов на уроке, в конце учебного года или по семестрам назначается экзамен и зачёт.

Машинные средства контроля

Для контроля знаний учащихся используют персональный компьютер. Один из возможных вариантов проверки знаний с использованием компьютера – это компьютерное тестирование.

1.4. Методы педагогического контроля

Устная проверка. Среди целевых установок проверки можно выделить следующие: проверить выполнение домашнего задания, выявить подготовленность учащихся к изучению нового материала, проверить степень понимания и усвоения новых знаний.

Письменно–графическая работа. Вторым широко применяемым методом контроля в обучении является проверка письменно-графических работ. Этот метод имеет свои

качественные особенности: большая объективность по сравнению с устной проверкой, охват нужного числа проверяемых, экономия времени. Применение письменных работ используется для: проверки знания теоретического материала; умения применять его к решению задач; контроля сформированных навыков.

Проверка практических и лабораторных работ. С помощью этого метода получают данные об умении учащихся применять полученные знания при решении практических задач, пользоваться различными таблицами, формулами, чертежными и измерительными инструментами, приборами.

2. Математические модели контроля знаний

2.1. История развития математической теории

После ранних работ Г.Эббингауза и Э.Торндайка эмпирические данные, полученные при обучении, стали выражать в виде уравнений, которые не опирались на какую-то теорию, а лишь интерпретировали имеющиеся данные [5,6]. Наиболее распространенным методом описания результатов эксперимента обучения возникла "кривая обучения", то есть график, который изображает, как в условиях заданного эксперимента от пробы к пробе изменяются характеристики обучаемого или группы обучаемых. В качестве "функции обучения" предлагались разнообразнейшие аналитические функции, в том числе гипербола, функция экспоненциального роста, арктангенс. Однако ни одна из этих функций не была получена на основе фундаментальной теории обучения.

О. Щукаревым было выведено следующее уравнение:

$y = a - be^{-cn}$, где: y – усвоение, обусловленное числом правильных воспроизведений (успехов) за единицу времени; n – число испытаний за единицу времени; a – граница усвоения при $n \rightarrow \infty$; b и c – константы.

Т.Робертсоном было предложено уравнение вида:

$y = \frac{be^{An}}{c + e^{An}}$, где: y – усвоения, n – число испытаний за единицу времени; $A = ab$, a , c ,

– константы (параметры обучаемого), b – граница усвоения при $n \rightarrow \infty$.

Л.Терстоун предложил следующую формулу (так называемый гиперболический закон обучения):

$y = \frac{a(n+c)}{(n+c)+b}$, где: y – усвоения, n – число испытаний, a и c – константы, b –

скорость обучения.

Дальнейшее развитие применения математических методов к исследованию проблем обучения связано с работами К.Халла [5,6]. Он ввел переменную, которую назвал "силой навыка". Она выражается формулой:

$H_{RS} = M(1 - e^{-bn})$, где: H_{RS} – «сила навыка», или ассоциативная переменная, которая связывает стимул и реакцию; M – асимптотическое значение «силы навыка»; b – параметр, который выражает скорость обучения; n – число учебных попыток за единицу времени.

В 40-50-е гг. 20-го столетия в психологии начало складываться представление об обучении как о стохастическом процессе. Г. Бушем и Ф. Мостеллером, были сформулированы стохастические модели обученности. Для теории обучения особенно важным было сосредоточение внимания на вероятностях событий, последовательных эффектах и представлении данных в виде цепей Маркова.

Эта модель предполагает, что обучение есть бесконечный процесс, уровень обученности постоянно возрастает, но состояние полного обучения не наступает никогда.

Предполагают, что p_t есть вероятность правильного ответа при испытании t ($t = 0, 1, 2, \dots$). Тогда $q_t = 1 - p_t$ есть вероятность неправильного ответа при испытании t . В линейной модели предполагается, что с каждым новым испытанием вероятность q_t убывает, причем коэффициент убывания остается постоянным для всех испытаний. Это означает, что существует константа α , $0 < \alpha < 1$ такая, что $q_{i+1} = \alpha q_i$. Для вероятностей p_i имеем: $p_{i+1} = 1 - \alpha(1 - p_i) = \alpha p_i + (1 - \alpha)$.

Постоянная α называется параметром модели, оценка этой величины должна быть получена на основе экспериментальных данных. С помощью этой величины можно сделать прогноз вероятности ошибки или выдачи правильного ответа в опыте t : $q_t = \alpha^t q_0$ или $p_t = 1 - \alpha^t q_0$.

Постоянная q_0 – второй параметр модели, её значение необходимо также оценить на основе экспериментальных данных. Построив зависимость p_t от t , получим так называемую кривую обучения.

2.2. Модели оценки уровня знаний

Простая модель. Данная модель является самой простой и самой распространенной. Ответ студента на каждое задание оценивается по двухбалльной или многобалльной шкале. Оценка выставляется путем вычисления значения R :

$$R = \frac{\sum_{i=1}^k R_i}{n}, \text{ где } R_i \text{ – правильный ответ обучаемого на } i\text{-е задание; } k \text{ – количество}$$

правильных ответов из n предложенных ($k \leq n$), которое затем обычно округляется по правилам математики.

Окончательная оценка, как правило, определяется по формуле:

$$I = \begin{cases} 1, R \leq c_1, \\ 2, c_1 < R \leq c_2, \\ \dots \\ M, R > c_{M-1}. \end{cases}$$

Здесь: I – окончательная оценка, $\{c_1, c_2, \dots, c_M\}$ – вектор граничных значений, M – максимально возможная оценка (например, при пятибалльной шкале $M = 5$).

К достоинствам данной модели следует отнести простоту ее реализации. Недостатком модели является ее зависимость от единственного параметра (количества правильных ответов), т.е. она не учитывает не полностью точные ответы и характеристики заданий.

Модели современной теории тестов. Под современной теорией тестов здесь понимается существующая на Западе Item Response Theory (IRT), предназначенная для оценки латентных параметров испытуемых и параметров заданий теста посредством применения математико-статистических моделей измерения [3].

В отличие от классической теории, где индивидуальный балл тестируемого рассматривается как постоянное число, в IRT латентный параметр трактуется как некоторая переменная. Начальное значение параметра получается непосредственно из эмпирических данных тестирования.

В рамках основного предположения IRT устанавливается связь между латентными параметрами испытуемых и наблюдаемыми результатами выполнения теста. При установлении связи важно понимать, что первопричиной являются латентные параметры

Элементы первого множества – это значения латентного параметра, определяющего уровень подготовки N испытуемых θ_i ($i = 1, 2, \dots, N$). Второе множество образуют значения латентного параметра β_j ($j = 1, 2, \dots, n$), равные трудностям n заданий теста.

Однако на практике всегда ставится обратная задача: по ответам испытуемых на задания теста оценить значения латентных параметров θ и β . Для ее решения нужно ответить по меньшей мере на два вопроса.

Первый связан с выбором вида соотношения между латентными параметрами θ и β . Идея установления соотношения принадлежит датскому математику Г. Рашу, который

предложил ввести его в виде разности $(\theta - \beta)$, предполагая, что параметры θ и β оцениваются в одной и той же шкале.

Ответ на второй вопрос, который является центральным в IRT, связан с выбором математической модели для описания рассматриваемой связи между латентными параметрами и наблюдаемыми результатами выполнения теста.

В частности, можно рассматривать условную вероятность правильного выполнения i -м испытуемым с уровнем подготовки θ_i различных по трудности заданий теста, считая θ_i параметром i -го ученика, а β – независимой переменной. В этом случае условная вероятность будет функцией латентной переменной β :

$$P_i\{x_{ij} = 1 | \theta_i\} = f(\theta_i - \beta), i = 1, 2, \dots, N.$$

Здесь $x_{ij} = 1$, если ответ i -го испытуемого на j -е задание верный; $x_{ij} = 0$, если ответ i -го испытуемого на j -е задание неверный.

Аналогично вводится условная вероятность правильного выполнения j -го задания с трудностью β_j различными испытуемыми группы. Здесь независимой переменной является θ , а β_j – параметр, определяющий трудность j -го задания теста:

$$P_j\{x_{ij} = 1 | \beta_j\} = \varphi(\theta - \beta_j), j = 1, 2, \dots, n.$$

В теории IRT функции $f(\beta)$ и $\varphi(\theta)$ получили название Item Response Functions (IRF). Специальное название имеют и их графики. График функции P_j – это характеристическая кривая j -го задания (ICC), а график функции P_i – индивидуальная кривая i -го испытуемого (PCC).

При выборе вида функций P_i и P_j учитываются обстоятельства как эмпирического, так и математического характера. В предположении нормального распределения значений латентных переменных θ и β таких функций предлагаются две. Одна из них, обычно обозначаемая $\psi(x)$, относится к семейству логистических кривых, другая $\Phi(x)$, является интегральной функцией нормированного нормального распределения. Поскольку для одних и тех же значений x ординаты точек графиков функций $\Phi(x)$ и $\psi(1,7x)$ отличаются друг от друга достаточно мало, то в том, что их две, нет ни ошибки, ни противоречия.

Наиболее сильный аргумент в пользу логистической функции связан не с качеством измерений, а с относительной простотой ее аналитического задания. Поэтому в практических приложениях предпочтение обычно отдают функции $\psi(1,7x)$.

Число параметров, входящих в аналитическое задание функций, является основанием для подразделения семейства IRF на классы. Среди логистических функций различают:

- однопараметрическую модель Г. Раша

$$P_j(\theta) = \frac{e^{1,7(\theta-\beta_j)}}{1 + e^{1,7(\theta-\beta_j)}}, \quad P_i(\beta) = \frac{e^{1,7(\theta_i-\beta)}}{1 + e^{1,7(\theta_i-\beta)}}, \text{ где } \theta \text{ и } \beta - \text{ независимые переменные для}$$

первой и второй функций соответственно.

- двухпараметрическую модель А. Бирнбаума

$$P_j(\theta) = \frac{e^{1,7a_j(\theta-\beta_j)}}{1 + e^{1,7a_j(\theta-\beta_j)}}, \quad P_i(\beta) = \frac{e^{1,7a_i(\theta_i-\beta)}}{1 + e^{1,7a_i(\theta_i-\beta)}}.$$

Кроме прежних обозначений в этой модели появляются параметры a_i и a_j . Параметр a_j был введен А. Бирнбаумом для характеристики дифференцирующей способности задания при измерении различных значений θ ; параметр a_i указывает на меру структурированности знаний ученика.

- трехпараметрическую модель А. Бирнбаума

$$P_j\{x_{ij} = 1 | \beta_j\} = c_j + (1 - c_j) \frac{e^{1,7a_j(\theta-\beta_j)}}{1 + e^{1,7a_j(\theta-\beta_j)}}, \text{ где } c_j \text{ является третьим параметром}$$

модели, характеризующим вероятность правильного ответа на задание j в том случае, если этот ответ угадан, а не основан на знаниях ученика.

В каждой из представленных моделей параметры θ и β выражаются как шкалированные показатели единой для всех моделей шкалы логитов. При отношении двух величин, равном e , их различие составит 1 логит. Начальные значения параметров θ и β находят по формулам:

$$\theta_i^0 = \ln \frac{p_i}{q_i}, \quad \beta_j^0 = \ln \frac{q_j}{p_j}, \text{ где } p_i \text{ и } q_i - \text{ доли правильных и неправильных}$$

соответственно ответов i -го ученика на задания теста; p_j и q_j – доли правильных и неправильных ответов учеников на j -е задание теста.

Затем начальные значения параметров переводятся в единую интервальную шкалу стандартных оценок. Стандартизация достигается с помощью ряда специальных преобразований, в ходе которых вычисляются среднее значение $\bar{\theta}$ для множества θ_i^0 ($i = 1, 2, \dots, N$), среднее значение $\bar{\beta}$ для множества β_j^0 ($j = 1, 2, \dots, n$), дисперсии по этим множествам и поправочные коэффициенты X и Y . Оценки параметров θ и β в единой интервальной шкале находятся по формулам:

$$\theta_i = \bar{\theta} + X\theta_i^0, \quad \beta_j = \bar{\beta} + Y\beta_j^0.$$

Роль последних двух формул в развитии современной теории тестов трудно переоценить, хотя на первый взгляд они имеют узкую практическую направленность. Эти формулы позволяют преодолеть ряд существенных недостатков классической теории

тестов, поскольку с их помощью можно получить объективные оценки параметров испытуемых и заданий, не зависящие друг от друга и выраженные в единой интервальной шкале.

III. МЕТОДИКА ИНТЕЛЛЕКТУАЛЬНОГО КОНТРОЛЯ ЗНАНИЙ НА ОСНОВЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ВЕРБАЛЬНЫХ ЯЗЫКОВЫХ ЗНАНИЙ

1. Исследование языкового сознания

Предлагаемая методика интеллектуального контроля знаний использует оригинальную информационную модель базы тестовых заданий (и самого тестового задания), которая основана на предложенной Ю.Н. Карауловым модели представления вербальных языковых знаний [7].

В последнее время всесторонне обсуждается проблема взаимосвязи культуры, языка и сознания: проводятся исследования языковой картины мира у носителей определённого языка, создаются ассоциативные словари разных языков. Бесспорно – языки культуры многообразны и имеют свои особенности, они непередаваемы адекватно ни на один другой язык.

Культурно-этнический компонент, отражающий так называемую “языковую картину мира” его носителей как факт обыденного сознания, воспринимается пофрагментально в лексических единицах языка, однако, сам язык непосредственно этот мир не отражает. Он отражает лишь способ представления этого мира, и поэтому выражение “языковая картина мира” в достаточной мере условно: образ мира, воссоздаваемый по данным одной лишь языковой семантики, скорее схематичен, поскольку его фактура сплетается преимущественно из отличительных признаков, положенных в основу категоризации и номинизации предметов, явлений и их свойств, и для адекватности языковой образ мира корректируется эмпирическими знаниями о действительности, общими для пользователей определённого естественного языка.

Каждый язык по-своему членит мир, т.е. имеет свой способ его концептуализации. Значит, каждый язык имеет особую картину мира, и языковая личность обязана организовать содержание высказывания в соответствии с этой картиной. И в этом проявляется специфически человеческое восприятие мира, зафиксированное в языке.

Языковая картина мира не стоит в ряду со специальными картинами мира (химической, физической и др.), она им предшествует и формирует их, потому что человек способен понимать мир и самого себя благодаря языку, в котором закрепляется общественно-исторический опыт – как общечеловеческий, так и национальный. Ю.Д. Апресян подчёркивал донаучный характер языковой картины мира, называя её наивной картиной. Языковая картина мира как бы дополняет объективные значения о реальности, часто искажая их.

2. Ассоциации и ассоциативный тезаурус

Ассоциативный вербальный эксперимент впервые был использован для исследования речевого поведения и описан в 1901 году немецкими исследователями А.Тумбом и К.Марбе. Сначала он использовался для изучения психических процессов (генерализация, воспроизведение и др.), а затем стал все активнее применяться для исследования семантики [8].

В основе классификаций вербальных ассоциаций могут лежать логические отношения между обозначаемыми этими словами понятиями, субординатные (вербальный стимул выражает более общее понятие, чем вербальная реакция, например: посуда – кувшин), координированные (вербальный стимул и вербальная реакция являются частными случаями одной и той же категории, например: кувшин – кружка). Различаются вербальные ассоциации синтагматические (типа век – двадцатый, вера – истинная) и парадигматические (типа век – время, вера – надежда).

Значительный экспериментальный материал показывает, какую важную роль играют процессы образования простых ассоциаций в различных психических процессах, а именно: в припоминании, в переносе, в опосредованных обобщениях, в ассоциативных обобщениях и в восприятии.

Наибольший интерес для исследователей представляют так называемые ассоциативные тезаурусы.

Ассоциативный тезаурус современного русского языка построен по результатам массового ассоциативного эксперимента с носителями русского языка, проводившегося с 1986 по 1996 годы [9]. Ассоциативный тезаурус состоит из трех частей, а каждая из частей состоит в свою очередь из прямого и обратного словарей. Все три части связаны между собой, так как отражают результаты последовательных этапов массового ассоциативного эксперимента. Слова-реакции, полученные в ходе первого этапа анкетирования (1986-1991), использовались в качестве слов-стимулов при повторении эксперимента с иным контингентом носителей русского языка, а прирост новых слов среди реакций второго этапа образовал список стимулов завершающего третьего этапа эксперимента.

Представленный в виде многомерной ассоциативной сети ассоциативный тезаурус дает четкое представление об устройстве и функционировании языкового сознания «усредненного» носителя того или иного языка и тем самым о его образе мира. Использование материалов ассоциативного тезауруса позволяет наметить новые пути в исследовании механизмов речевого воздействия и поведения, а также в изучении семантических законов в языке в целом, принципов соотношения семантики и синтаксиса в речи и языке, закономерностей социализации индивидуальных семантических изменений и установлении новых типов ассоциативных связей.

3. Понятие фигуры знания

Ранее уже говорилось о том, что Ю.Н. Караулов вводит понятие фигуры знания как элементарной когнитивной единицы, множество которых на самом нижнем уровне языкового сознания образует образ мира [7]. Фигура знания является, следовательно, элементарной единицей в модели представления вербальных языковых знаний.

Согласно Ю.Н. Караулову, фигура знания является пятикомпонентной структурой. Рассмотрим ход рассуждений, обосновывающих это заключение.

Единица знаний о мире включает не только смысл знака, но и сам знак: отдельно взятый смысл еще не составляет знания, знание двусоставно и с необходимостью подразумевает соединение как минимум двух компонентов – знака и смысла. Например, каждый из следующих смыслов – «в африканском поверье: мертвец, выполняющий волю того, кто вызвал его дух», «окончательно деградировавшая ткань», «широко распространенный способ печати» – становится элементом знаний о мире, соединяясь в языке с соответствующим другим компонентом, а именно, со словом (знаком): ЗОМБИ, ТРЯПКА, ОФСЕТ.

Сразу можно отметить, что приведенные способы задания смысла элементов знания – дефиниция, дескрипция, метафора – произвольны по отношению к знакам, один и тот же знак может быть задан многими различными способами.

Ю.Н. Караулов выделяет два режима работы когнайзера – активный и пассивный. В активном режиме исходным является словесный **знак**, и через его **смысл** когнайзер приходит к **способу** обозначения: Знак → Смысл → Способ.

В пассивном режиме заданным оказывается **смысл**, с неизбежностью уже оформленный тем или иным **способом**, а целью поиска становится **знак**: Смысл → Способ → Знак.

Три этих составляющих, три компонента элементарной единицы знания создают ее фундамент. Содержание будет исчерпано целиком, если к трем названным основным компонентам мы добавим параметры, характеризующие экстенционал элементарной единицы знания.

С точки зрения относительной ценности единиц знания Ю.Н. Караулов выделяет два их типа – **знание-рецепт** (несет важное, основное знание) и **знание-ретушь** (сообщает второстепенные, сведения).

Таким образом, можно завершить цепочку параметров, характеризующих элементарную единицу знания, добавив к ее интенциональной части экстенциональные показатели – когнитивную **область** и **функцию** [2]. Для пассивного режима работы когнайзера цепочка выглядит таким образом:

Смысл → Способ → Знак → Область → Функция,

В активном режиме третья позиция меняется местами с первой:

Знак → Способ → Смысл → Область → Функция.

Цепочечное представление элемента знания хорошо подчеркивает его процессную природу: это не застывшая, жесткая конструкция, а процесс, постоянно текущее состояние. Недостаток цепочечного представления в том, что сам процесс предстает в этом случае в одном измерении, тогда как на самом деле связи между узлами не только обратимы, но и могут быть дистантными в этой цепочке, т.е. смысл может быть связан непосредственно с областью, а способ, например, с функцией. Схематически фигура знания изображена на рис.1.

Таким образом, Ю.Н. Караулов делает вывод о существовании в языковом сознании своеобразных «фигур знания», соответствующих элементарным когнитивным единицам. Фигура знания представляет собой постоянно текущий динамический процесс взаимодействия пяти охарактеризованных выше параметров.

4. Информационная модель базы тестовых заданий

В теории педагогического тестирования известна открытая форма тестового задания как одна из четырех основных форм. Разновидностью этой формы является так называемое задание на дополнение с ограниченным ответом, когда ученик дописывает пропущенное слово, формулу, символ или число на месте прочерка. Задания на дополнения кажутся ученикам более трудными, так как в них исключается догадка. Но именно это свойство делает задания на дополнение исключительно привлекательными для педагогов, особенно для тех, кто привык в своей работе опираться на традиционные средства контроля и не доверяет тестам [2].

Есть и недостатки, связанные с трудностями, возникающими при оценке ответа учеников. Дописывая ответ на месте прочерка, ученик может выбрать синонимы пропущенного запланированного разработчиком слова или изменить порядок следования элементов в пропущенной формуле. Рекомендуется, чтобы ответ на задание был достаточно кратким, по крайней мере, не превышающим двух-трех слов.

Очевидно, что входящее в состав фигуры знания слово (знак), как раз, может являться ответом на тестовое задание, формулировкой которого будет формула смысла. Такое тестовое задание в открытой форме является идеальным в том случае, если оно действительно подразумевает только один ответ, то есть одно слово (знак), а остальные слова, данные в качестве ответа, будут являться неверными.

Поэтому применяют подход, когда вместо одного слова (знака) при описании тестового задания указывается множество знаков, являющихся верными. Этот подход выбран и в данной методике.

Функция оценивания задания является дихотомической, то есть за каждое задание испытуемый получает 1 балл, если его ответ входит во множество правильных ответов, либо 0 баллов в противном случае. В качестве основной модели обработки результатов и оценивания выбрана модель Раша, для которой использование дихотомических тестовых заданий является предпочтительным [2].

Тем не менее, у способа задания смысла как компонента тестового задания есть и недостаток. Он заключается в том, что способ для фигуры знания – это лингвистическое понятие, и кроме тривиальных способов, таких как дефиниция, антоним, синоним, существуют и достаточно специфические. Поэтому целесообразно, исходя из выделенных Ю.Н.Карауловым лингвистических способов задания смысла, ввести новые «педагогические» способы, которые бы описывали возможные виды тестовых заданий. Некоторые способы из лингвистических можно взять в чистом виде, например, дефиницию, дескрипцию, метафору, антоним, синоним.

Функция, входящая в состав фигуры знания, также может являться и компонентом тестового задания, и представлять собой дополнительную субъективную характеристику тестового задания, определяемую разработчиком теста.

Очевидно, что построение моделей учебных курсов, выделение так называемых «учебных элементов», дидактических единиц, установление их иерархии является субъективным.

Тестовое задание, следовательно, состоит из следующих компонентов: формула смысла (формулировка задания); Множество знаков (множество правильных ответов); способ задания смысла; Множество дидактических единиц, входящих в разные тематические разбиения; функция.

5.Обработка результатов контроля

Обработку результатов тестирования в данной методике можно разделить на три основных класса:

1. Аналитическая обработка результатов в соответствии с информационной моделью базы тестовых заданий.
2. Оценка качества теста, которая включает:
 - оценку качества теста в соответствии с классической теорией тестирования;
 - оценку качества теста в соответствии с теорией IRT по модели Раша.
3. Подсчет тестовых баллов испытуемых, для двух различных случаев:

- применение простой модели оценивания, где окончательный балл (отметка) вычисляется на основе процента верно выполненных заданий теста;

- применение модели Раша, когда при обработке результатов вычисляются параметры уровней подготовки испытуемых и параметры сложности заданий теста в единой интервальной шкале логитов.

Оценка качества теста в соответствии с классической теорией тестирования включает в себя:

- 1) Получение распределения индивидуальных баллов в аналитическом и графическом виде.

- 2) Получение частотного распределения индивидуальных баллов, в аналитическом и графическом виде, при этом возможно использовать как полигон частот, так и гистограмму.

- 3) Расчет выборочного среднего и исправленной выборочной дисперсии.

- 4) Расчет корреляционной матрицы результатов тестирования

- 5) Расчет точечно-бисериального коэффициента корреляции $(r_{pbis})_j$ задания теста с суммой баллов (для всех заданий теста).

- 6) Расчет коэффициента надежности теста по корреляции.

- 7) Расчет коэффициента надежности теста по методу Кьюдера-Ричардсона.

- 8) Расчет стандартной ошибки измерения для теста S_E .

Анализ частотного распределения индивидуальных баллов и значение среднего арифметического и дисперсии позволяет оценить меры центральной тенденции совокупности результатов, полученные при выполнении теста, близость распределения баллов к нормальному распределению.

Корреляционная матрица результатов тестирования является классическим средством для анализа качества теста, его системообразующих свойств.

Точечно-бисериальный коэффициент корреляции позволяет оценить валидность и дискриминативность заданий теста.

Коэффициенты надежности и стандартная ошибка измерения являются интегральными показателями для оценки качества теста.

Оценка качества теста в соответствии с современной теорией тестирования IRT и моделью Раша включает в себя:

- 1) Получение аналитического и графического представления следующих функций:

- индивидуальные кривые испытуемых P_i (функция $\varphi(\theta)$);

- характеристические кривые заданий теста P_j (функция $f(\beta)$);

- информационные функции заданий теста $I_j(\theta)$;
- информационная функция теста $I(\theta)$;

Анализ качества теста в соответствии с моделью Раша дает много нового по сравнению с классической теорией тестирования. На основании визуального анализа этих функций можно выяснить, насколько группа гомогенна по уровню подготовки, для каких интервалов уровней подготовки испытуемых отсутствуют задания с соответствующим уровнем трудности, насколько распределения значений латентных переменных близки к нормальному, какие задания необходимо добавить в тест, какие задания лишние, какие задания дают максимальную ошибку измерения уровня подготовки, можно ли разбить исходный тест на два различных теста и многое другое [2].

IV. ТЕКСТЫ ОСНОВНЫХ ПРОГРАММ.

Листинг 1. Модуль авторизации пользователя.

```
<?php
    if (!isset($_POST['enter'])) {
?>
    <table align="center" width="20%" >
        <FORM method="POST">
            <tr>
                <td>
                    Логин:
                </td>
                <td>
                    <input type="text" name="login"><br>
                </td>
            </tr>
            <tr>
                <td>
                    Пароль:
                </td>
                <td>
                    <input type="password" name="password"><br>
                </td>
            </tr>
            <tr>
                <td COLSPAN="2" align="center">
                    <input type="submit" name="enter" value="Войти в систему">
                </td>
            </tr>
        </FORM>
    </table>
<?php

    } else {
        if ($_POST['login']!=" and $_POST['password']!=") {
```

```

// Сохраняем нужные данные и защищаемся от инъекций
$safe_name=mysql_escape_string($_POST['login']);
$safe_pass=md5(mysql_escape_string($_POST['password']));
    // Подключение к базе данных
    require_once("Connect.php");
// запрос по таблице USERS
$sql="SELECT login,password,name,surname,patronymic,role FROM users WHERE
login=".$safe_name." and password=".$safe_pass.""";
$result = mysql_query($sql);
// Если записи нету, то сообщаем о неверном пароле
if (!mysql_num_rows($result)) die("<center><b>Неверный логин или пароль!</b> <a
href='index.php'>Назад</a>.</center> <meta http-equiv='refresh'; content='3; url=index.php'>
");
else {
    //Получаем результат запроса
    $line=mysql_fetch_row($result);
    //Записываем факт авторизации в сессию
    $_SESSION['authorized']=true;
    //Сохраняем имя пользователя
    $_SESSION['login']=$_POST['login'];
    //Сохраняем роль пользователя
    $_SESSION['role']=$line[5];
    //Сохраняем остальные данные
    $_SESSION['name']=$line[2];
    $_SESSION['surname']=$line[3];
    $_SESSION['patronymic']=$line[4];
    //Выводим, что авторизация пройдена
    echo "<center><b>Авторизация прошла успешно!</b> <a href='main.php'>На
главную страницу</a></center>";
    echo "<meta http-equiv='refresh'; content='0; url='main.php'>";
}
} else {
    //Если пользователь не ввел данные, отказываем в доступе
    die("<center><b>Неверный логин или пароль!</b> <a
href='index.php'>Назад</a>.</center> <meta http-equiv='refresh'; content='3;

```

```
url=index.php'>");
    }
}

?>
```

Листинг 2. Страница проведения тестирования

```
<?php
    if (isset($_POST['Result'])) {
        include("Connect.php");
        $sql="SELECT Point FROM testingitems WHERE id = '".$_POST['AID']."'";
        $result = mysql_query($sql) or die(mysql_error());
        $line = mysql_fetch_row($result);
        echo "<b>Число правильных ответов: ".$line[0]."</b>";
        die;
    }
// Основное окно
if (!isset($_POST['OK'])) {
    require_once("Connect.php");
    // Создание тестирования
    $sql="INSERT INTO testingitems SET User =
".$_SESSION['login']." ,Answers=", Point='0'";
    $result = mysql_query($sql) or die(mysql_error());
    $id = mysql_insert_id();
    $sql="SELECT * FROM testingitems WHERE id = ".$id."";
    $result = mysql_query($sql) or die(mysql_error());
    $line = mysql_fetch_row($result);
    $sql="UPDATE testing SET Items=CONCAT(Items,'".$line[0].";)";
    $result = mysql_query($sql) or die(mysql_error());

    // запрос по таблице tests
    $sql="SELECT * FROM testitems WHERE Test='".$_GET['ID']";
    $result = mysql_query($sql);
```

```

$line=mysql_fetch_row($result);

// Оформление таблицы с вопросом
echo "<table width='70%' cellspacing='0'>";
echo "<tr>";
echo "<td width='10%' align='center'>";
// Вывод вопроса
echo $line[1];
echo "</td>";
echo "</tr>";
echo "<td width='30%' align='center'>";
    echo "<form method='POST'>";
    echo "<input type='hidden' name='Test' value=\".$_GET['ID'].\">";
    // Сохранение номера следующего вопроса
    echo "<input type='hidden' name='ID' value='2'>";
    // Сохранение ключа тестирования
        echo "<input type='hidden' name='AID' value=\".$id.\">";
    echo "<input type='text' name='ANSWER'><BR> ";
    echo "<input type='submit' name='OK' value='Ответить'>";
    echo "</form>";
echo "</td>";
echo "</tr>";
echo "</table>";

} else {
if (isset($_POST['OK'])) {
    require_once("Connect.php");
    //Проверка верности ответа

    $Last = $_POST['ID']-1;
    $sql="SELECT * FROM testitems WHERE Test=\".$_POST['Test'].\" LIMIT
".$Last.", 1";
    $result = mysql_query($sql);
    $line=mysql_fetch_row($result);

```

```

$answer = $line[2];
if (strpos($answer, ';') == 0) {
    $IsAT = ($_POST['ANSWER']==$answer);
} else {
    while (strpos($answer, ';') != 0) {
        $IsAT = ($_POST['ANSWER']==substr($answer, 0, strpos($answer, ';')));
        if ($IsAT) {
            $answer = "";
        } else {
            $answer = substr($answer, strpos($answer, ';')+1);
        }
    }
    $IsAT = ($_POST['ANSWER']==$answer);
}

// Запись результата в базу
$testing = $_POST['AID'];
$sql="SELECT * FROM testingitems WHERE id='".$_.$testing.'"";
$result = mysql_query($sql) or die(mysql_error());
$line=mysql_fetch_row($result);
$answer = $line[2];
if ($IsAT) {
    $IsAT = '1;';
} else {
    $IsAT = '0;';
}
$answer = $answer.$IsAT;
$points = $line[3]+$IsAT;
$sql="UPDATE testingitems SET Answers='".$_.$answer."', Point='".$_.$points.'"
WHERE id='".$_.$testing.'"";
$result = mysql_query($sql) or die(mysql_error());

// Показ следующего вопроса
$sql="SELECT * FROM testitems WHERE Test='".$_.$_POST['Test']."' LIMIT
".$_.$_POST['ID'].", 1";

```

```

$result = mysql_query($sql);

if (!mysql_num_rows($result)) { //Конец теста
    echo "<table width='70%' cellspacing='0'>";
    echo "<tr>";
    echo "<td width='10%' align='center'>";
    echo "Обработать результаты";
    echo "</td>";
    echo "</tr>";
    echo "<td width='30%' align='center'>";
        echo "<form method='POST'>";
        echo "<input type='hidden' name='Test' value=\"$_POST['Test'].\">";
        echo "<input type='hidden' name='AID' value=\"$.testing.\">";
        echo "<input type='submit' name='Result' value='Ответить'>";
        echo "</form>";
        echo "</td>";
        echo "</tr>";
        echo "</table>";
    } else { //Получаем результат запроса
    echo "<table width='70%' cellspacing='0'>";
    $line=mysql_fetch_row($result);
    echo "<tr>";
    echo "<td width='10%' align='center'>";
    // Показ вопроса
    echo $line[1];
    echo "</td>";
    echo "</tr>";
    echo "<td width='30%' align='center'>";
        echo "<form method='POST'>";
        $NextItem = $_POST['ID']+1;
        // Сохранение номера следующего вопроса
        echo "<input type='hidden' name='ID' value=\"$.NextItem.\">";
        // Сохранение ключа теста
        echo "<input type='hidden' name='Test' value=\"$_POST['Test'].\">";
        // Сохранение ключа тестирования

```

```

        echo "<input type='hidden' name='AID' value='".$testing.">";
        echo "<input type='text' name='ANSWER'><BR> ";
        echo "<input type='submit' name='OK' value='Ответить'>";
        echo "</form>";

    echo "</td>";
    echo "</tr>";
    echo "</table>";
    }
    }
}
?>

```

Листинг 3. Страница управления учетными записями.

```

<?php
// Основное окно
if (isset($_POST['Edit'])) { // Нажата кнопка "Редактировать"
    // Подключение к базе данных
    include("Connect.php");
    $sql="SELECT * FROM users WHERE id ='".$_POST["UID"]."";
    $result = mysql_query($sql) or die(mysql_error());
    $line = mysql_fetch_row($result);
    // Вывод формы для редактирования
    echo "<table>
        <tr>
            <td align='left'>
                <b>Логин:</b>
            </td>
            <td>
                <form method='POST'>
                    <input type='text' name='Login' value='".$line[1]."'>
                </td>
        </tr>
        <tr>
            <td align='left'>
                <b>Пароль:</b>

```



```
</td>
<td>
    <input type='password' name='Password'>
</td>
</tr>
<tr>
<td align='left'>
    <b>Имя:</b>
</td>
<td>
    <input type='text' name='Name' value='".$line[3]."'>
</td>
</tr>
<tr>
<td align='left'>
    <b>Фамилия:</b>
</td>
<td>
    <input type='text' name='Surname' value='".$line[4]."'>
</td>
</tr>
<tr>
<td align='left'>
    <b>Отчество:</b>
</td>
<td>
    <input type='text' name='Patron' value='".$line[5]."'>
</td>
</tr>
<tr>
<td align='left'>
    <b>Статус:</b>
</td>
<td>
    <p><select name='Role'>
```

```

        <option selected>Выберите роль пользователя</option>
        <option value='teacher'>Учитель</option>
        <option value='student'>Пользователь</option>
    </select></p>
</td>
</tr>
<tr>
<td align='left'>
    <b>Кафедра:</b>
</td>
<td>
    <input type='text' name='Chairs' value='".$line[6]."'>
</td>
</tr>
<tr>
<td align='left'>
    <b>Года обучения:</b>
</td>
<td>
    <input type='text' name='Years' value='".$line[7]."'>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<center>
    <input type='submit' name='EditEnd' value='Сохранить'>
    <input type='hidden' name='UID' value='".$_POST["UID"]."'>
</form>
</center>
</td>
</tr>
</table>";
} if (isset($_POST['EditEnd'])) { // Завершение редактирования
// Подключение к базе данных
include("Connect.php");

```

```

// Хеширование пароля
if (isset($_POST['Password'])) {
    $password= "password=MD5("".$_POST["Password"].").";
} else {
    $password= "";
}

// Составление и отправка запроса
$sql="UPDATE users SET login = '".$_POST["Login"]."', ".$password." name=
".$_POST["Name"]."', surname='".$_POST["Surname"]."', patronymic='".$_POST["Patron"]."',
chairs='".$_POST["Chairs"]."',    years='".$_POST["Years"]."',    role='".$_POST["Role"]."'
WHERE id='".$_POST["UID"]."'";

$result = mysql_query($sql) or die(mysql_error());
echo "<b>Изменения успешно сохранены</b>";
} elseif (isset($_POST['Delete'])) { // Нажата кнопка удаления
    // Вывод предупреждения
    echo "<table>
        <tr>
            <td><b>Вы точно хотите удалить пользователя?</b></td>
            <td>
                <form method='Post'>
                    <input type='submit' name='DeleteEnd' value='Да'>
                    <input type='submit' name='DeleteCancel' value='Нет'>
                    <input type='hidden' name='UID' value='".$_POST["UID"]."'>
                </form>
            </td>
        </tr>
    </table>";
} elseif (isset($_POST['DeleteEnd'])) { // Завершение удаления
    // Подключение к базе данных, удаление записи
    include("Connect.php");
    $sql="DELETE FROM users WHERE id =".$_POST["UID"]."'";
    $result = mysql_query($sql) or die(mysql_error());
    echo "<b>Пользователь успешно удален</b>";
} elseif (isset($_POST['Create'])) { // Нажата кнопка создания нового
пользователя

```

```
// Вывод формы для заполнения
echo "<table>
  <tr>
    <td align='left'>
      <b>Логин:</b>
    </td>
    <td>
      <form method='POST'>
        <input type='text' name='Login'>
      </td>
  </tr>
  <tr>
    <td align='left'>
      <b>Пароль:</b>
    </td>
    <td>
      <input type='password' name='Password'>
    </td>
  </tr>
  <tr>
    <td align='left'>
      <b>Имя:</b>
    </td>
    <td>
      <input type='text' name='Name'>
    </td>
  </tr>
  <tr>
    <td align='left'>
      <b>Фамилия:</b>
    </td>
    <td>
      <input type='text' name='Surname'>
    </td>
  </tr>
</table>
```

```
<tr>
  <td align='left'>
    <b>Отчество:</b>
  </td>
  <td>
    <input type='text' name='Patron'>
  </td>
</tr>
<tr>
  <td align='left'>
    <b>Статус:</b>
  </td>
  <td>
    <p><select name='Role'>
      <option selected>Выберите роль пользователя</option>
      <option value='teacher'>Учитель</option>
      <option value='student'>Пользователь</option>
    </select></p>
  </td>
</tr>
<tr>
  <td align='left'>
    <b>Кафедра:</b>
  </td>
  <td>
    <input type='text' name='Chairs'>
  </td>
</tr>
<tr>
  <td align='left'>
    <b>Года обучения:</b>
  </td>
  <td>
    <input type='text' name='Years'>
  </td>
</tr>
```

```

        </tr>
    <tr>
        <td colspan='2' align='center'>
            <center>
                <input type='submit' name='CreateEnd' value='Создать'>
            </form>
        </center>
    </td>
</tr>
</table>";
} elseif (isset($_POST['CreateEnd'])) { // Завершение создания пользователя
    // Подключение к базе данных и вставка записи.
    include("Connect.php");
    $sql="INSERT INTO users SET login='".$_POST["Login"]."', password=MD5(
".$_POST["Password"].' '), name='".$_POST["Name"]."', surname='".$_POST["Surname"]."',
patronymic='".$_POST["Patron"]."', chairs='".$_POST["Chairs"]."',
years='".$_POST["Years"]."', role='".$_POST["Role"].'";';
    $result = mysql_query($sql) or die(mysql_error());
    echo "<b>Пользователь успешно зарегистрирован</b>";
} else { // Вывод таблицы с возможными действиями
    echo "<table>
    <tr>
        <td align='center'><b>Выберите действие</b></td>
    </tr>
    <tr>
        <td align='center'>
            <form method='POST'>
                <input type='submit' name='Create' value='Создать'>
            </form>
        </td>
    </tr>
</table>";
}
?>
<?php // Боковая панель

```

```

// Таблица пользователей
require_once("Connect.php");
// Преподаватели
$sql = "SELECT * FROM users WHERE role='teacher' ORDER BY chairs";
$result = mysql_query($sql) or die(mysql_error());
$i=0;
echo "<table width='100%' valign='top' align='center' cellspacing='0'>
<tr>
    <td align='center' height='50' width='100%' colspan='3'>
        <b>Преподаватели</b>
    </td>
</tr>";
// Вывод информации о каждом преподавателе
while (mysql_num_rows($result) > $i) {
    $line = mysql_fetch_row($result);
    echo "<tr>
<td height='50' width='60%'>
    ".$line[3]." ".$line[4]." ".$line[5]."
</td>
<td height='50' width='10%'>
    ".$line[6]."
</td>
<td>
    <form method='POST'>
        <input type='submit' name='Edit' value='Править'>
        <input type='hidden' name='UID' value='".$line[0]."'>
    </form>
    <form method='POST'>
        <input type='submit' name='Delete' value='Удалить'>
        <input type='hidden' name='UID' value='".$line[0]."'>
    </form>
    </td>
</tr>";
    $i++;
}

```

```

// Студенты
$sql = "SELECT * FROM users WHERE role='student' ORDER BY chairs";
$result = mysql_query($sql) or die(mysql_error());
$i=0;
echo "<table width='100%' align='center' cellspacing='0'>
<tr>
  <td align='center' height='50' width='100%' colspan='4'>
    <b>Студенты</b>
  </td>
</tr>";
// Вывод информации о каждом студенте
while (mysql_num_rows($result) > $i) {
  $line = mysql_fetch_row($result);
  echo "<tr>
  <td height='50' width='50%'>
    ".$line[3]." ".$line[4]." ".$line[5]."
  </td>
  <td height='50' width='10%'>
    ".$line[6]."
  </td>
  <td height='50' width='20%'>
    ".$line[7]."
  </td>
  <td>
    <form method='POST'>
      <input type='submit' name='Edit' value='Править'>
      <input type='hidden' name='UID' value='".$line[0]."'>
    </form>
    <form method='POST'>
      <input type='submit' name='Delete' value='Удалить'>
      <input type='hidden' name='UID' value='".$line[0]."'>
    </form>
  </td>
</tr>";
  $i++;
}

```



```
}  
echo "</table>";  
?>
```