# Министерство образования и науки Российской Федерации

# Федеральное государственное бюджетное образовательное учреждение

# высшего профессионального образования

# «Ижевский государственный технический университет имени М. Т. Калашникова»

# Кафедра «Программное обеспечение»

Отчет по лабораторным работам

по дисциплине: «Конструирование программного обеспечения»

на тему: «Электронное пособие по теории вероятностей»

Выполнил

студент гр. Б08-191-1 Д.И.Галатов

Принял В.Г. Власов

Ижевск

2016

СОДЕРЖАНИЕ

1.ВВЕДЕНИЕ

1.1 Назначение системы

1.2 Область применения системы

1.3 Определения, акронимы, аббревиатуры

1.4 Требования к системе

1.5 Обзор системы

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

2.6 Характеристики пользователя

2.7 Допущения и зависимости

2.8 Оперативные сценарии

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема

5.2 Классы кодирования

1. ВВЕДЕНИЕ
   1. Назначение системы

Система предназначена для автоматизации интерактивного обучения школьников (студентов).

* 1. Область применения системы

Система может применяться как в рамках государственного образования, так и в целях самообучения. Система представляет собой линейный коридор, проходя по которому, обучающийся не упустит важные элементы обучения. В системе необходимо предусмотреть следующие группы пользователей: автор, учитель, ученик.

* 1. Определения, акронимы, аббревиатуры

Автор – Человек, который добавляет задания, обновляет темы и экзамен.

Учитель – пользователь, обладающий необходимыми правами для сбора статистики и контроля успеваемости учеников.

Ученик – пользователь, занимающийся по курсу.

БД – база данных.

* 1. Требования к системе

1.4.1) Проект с программой и базами данных хранятся в одной папке на локальном диске.

1.4.1.1) Для хранения данных используется база данных MS. Access.

1.4.1.2) использующийся язык программирования – C#.

1.4.2) Структура БД состоит и нескольких таблиц:

* Пользователи users: ID, keyuser, login, password, time, surname, name, otchestvo.
* с данными участника learner: ID, bday, keyslass, level, keycontrol, result, timedelation.
* с данными контролирующего teacher: ID, keyuser, keyclass.
* С названиями классов classes: Keyclass, nameclass
* С названиями названия тем topics: namethem, keyquest, level, text
* С заданиями и ответами quests: keyquest, quest, answer
* Задания контрольные-точки control: keycontrol, keyquest, namethem
* Экзамен exem: question, answerquestion, score.

1.4.3) В системе существуют 3 роли:

* Обучающийся – человек, проходящий по пути обучения программой.
* Контроль – человек(учитель, преподаватель), может зайти под своим профилем и посмотреть результаты своей обучающейся группы.
* Автор – человек, заполняющий темы, задачи, экзамен контекстом.

1.4.4.1) При входе обучающегося, загружается его прогресс пройденного пути из БД по последней контрольной точке. Продолжает путь с того места, на котором остановился в прошлую сессию.

1.4.4.2) При входе Контролирующего, загружаются данные по соответствующему классу или группе, такие как: Среднее время обучения, средний результат Экзамена. Конкретные данные по отдельному человеку: время обучения, результат Экзамена, время последнего входа. Отображение производиться посредством форм (форма выбора своего класса по ключу).

1.4.4.3) Каждый контролирующий может вести несколько классов. Эта связь производиться по ключу и 2 таблиц teacher и classes

1.4.4.4) Автор работает только с базой данных, вносит информацию или редактирует существующую через формы Access. Объекты БД закрыты.

1.4.5) Обучающийся знакомится с темой, в виде текста в приложении.

1.4.5.1) Система должна иметь возможность отрисовки графиков, таблиц конкретных педагогических примеров по настройкам:

1.4.5.1.1) Количество повторов событий

1.4.5.1.2) Уровень выпадения события

1.4.5.1.3) Точность вычисления

1.4.6) Система обучения состоит из блоков, содержащих определенные темы обучения.

Темы хранятся в таблице topics и имеют свой уровень сложности.

1.4.6.1) Блок состоит из теоретической части, практической части, контрольных точек.

Данные находятся в соответствующих таблицах.

1.4.6.2) В зависимости от выбранного уровня сложности, после темы ученик должен пройти определенной количество контрольных точек, для перехода к следующему блоку.

1.4.7) Когда обучающийся входит в приложение, для него запускается таймер. При выходе из программы, таймер останавливается, и время записывается в БД.

1.4.8) После успешного прохождения всех контрольных точек по всем темам, необходимо пройти Экзамен.

1.4.9) В зависимости от уровня сложности, выбираются задания из таблицы exam по ключу.

1.4.10) Данные Экзамена записываются в БД в виде результата, записывается время прохождения.

* 1. Обзор системы

Система представляет собой прикладное приложение с базой данных. Обучающийся выбирает уровень обучения и, далее, проходя определенное количество контрольных точек, перемещается между тематических блоков. В конце обучения он выполняет экзамен, по результатам которого можно судить об его успешности.

Контролирующий может авторизоваться под своим профилем и посмотреть статистику по своей обучающейся группе.

Система находиться на локальном диске. Работа производится через интерфейс пользователя.

Таким образом, учащийся сможет пройти прямой путь обучения.

.

1. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ
   1. Системный контекст
   2. Режимы и состояния системы

Система может находиться в следующих состояниях:

Приложение может иметь следующие состояния:

-Закрытое:

- В этом случае работать можно только с базой данных автор

-Открытое:

- Пользователь авторизован в системе;

- Пользователь не авторизован в системе.

* 1. Основные функциональные возможности системы.

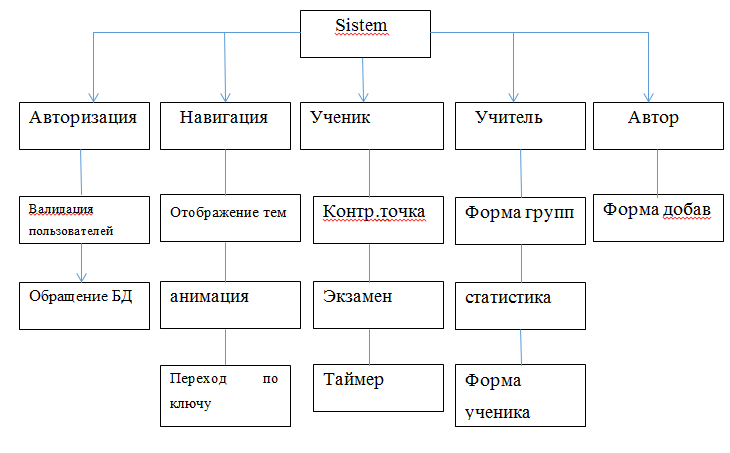


Рис.1. Классы анализа

* 1. Основные условия системы

Компьютер с системой должен иметь установленный MS Access .

Для работы приложения, необходим персональный компьютер  
процессор - двухядерный, оперативная память 2гб, жесткий диск 500Гб или больше. Компьютер средних характеристик.

* 1. Основные ограничения системы

Само приложение не имеет ограничений. Присутствуют ограничения в базе данных – для определенного типа поля – своя длинна данных. Ограничения 255 символов в текстовых полях. Для полей типа заданий и описания теоретической части поля типа MEMO.

* 1. Характеристики пользователя
* Обучающийся не обязан обладать специальными навыками для работы с системой.
* Контролирующий не обязан иметь навыки работы с системой.
* Автор заданий должен иметь минимальные навыки работы в MS Access.

2.7 Оперативные сценарии

Система предусматривает следующий сценарий. Учитель регистрируется в системе. Дети авторизуются в системе. Обучающийся заходит под своим профилем, продолжает работу с того места, на котором остановился, определенному по контрольным точкам. Еще одним сценарием является проход всех контрольных точек учеником и выполнение экзамена.

Сценарий учителя: контролирующий зашел под своим профилем. Он может выбрать свою группу, посмотреть статистику по ней или по контрктному ученику.

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

Компьютер и система не должны подвергаться вредным факторам.

3.1.3 Адаптируемость

В случае увеличения нагрузки на БД, проверяется размер свободного места на Диске.

3.1.4 Условия окружающей среды

Компьютер должен находиться в компьютерном классе с соблюдением мер безопасности.

3.2 Рабочие характеристики системы

Критическое количество запросов, обрабатываемых системой, зависит от возможностей компьютера.

3.3 Защита системы

База данных должна быть защищена паролем, а данные должны быть зашифрованы от перехвата вредоносными программами. Однако последних не должно быть, так как соблюдаются условия безопасности в компьютерном помещении, например, установленное лицензионное защитное ПО.

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

Требования по распределению функций для персонала описано в ТЗ.

3.5.2 Эксплуатационная технологичность

Техническая составляющая эксплуатационной технологичности зависит от характеристик компьютера. Программные изменения происходят в фоновом режиме и не вызывают задержек при обработке команд.

3.5.3 Надежность системы

Система защищена неправильного ввода информации, перехвата личной информации .

3.6 Стратегия и регулирование

Базовые настройки установленной системы регулируются при установке системы администратором, в зависимости от пожеланий владельца.

3.7 Устойчивость жизненного цикла системы

Для устойчивости жизненного цикла системы используется обратная связь с владельцем.

1. СИСТЕМНЫЕ ИНТЕРФЕЙСЫ

Sistem

IConnect

INavigation

IResult

ITeacher

IAutor

Ученик  
 Расчеттаблиц

Навигация

Авторизация

Автор

Учитель

Форма добав

Валидация пользователей

IExam

ICT

Обращение БД

IForm

Форма групп

Отображение тем

статистика

Контр.точка

анимация

Форма ученика

Экзамен

Таймер

IKey

Переход по ключу

Рис.2. Классы проектирования

Интерфейс авторизации

public interface IConnect

{

OpenDB(string p);

BasicQuery(string p, bool r);

createDB(string name, array col);

InsertInto(string table, array value);

closeDB();

}

Интерфейс навигации

public interface INavigation

{

void PrintTopics(X,Y: int);

void Row(object Row, array object[] Topics);

object View(object Row, array string[] theme);

}

Интерфейс Ученик

public interface IResult

{

viod GetSetting(int ret, int level, int might);

object Row();

void Print(object Row);

string LoadCT(string KeyUser);

void ExportCT(string NewCT);

void StartTimer();

}

Интерфейс Учитель

public interface ITeacher

{

void PrintTopics(string LoadCT);

void FormGROUP();void Stat (object group);

}

Интерфейс Автор

public interface IAutor

{

Void AddNew(string add);

}

Интерфейс Ключа

public interface IKey

{

string LoadCT(string KeyUser);

void ExportCT(string NewCT);

}

Интерфейс Форм

public interface IResult

{

void LoadForm(CT);

bool paintForm();

}

Интерфейс Экзамен

public interface IExam

{

void LoadDB(CT);

object View(object Row, array string[] theme)

DropTimer();

}

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема программы

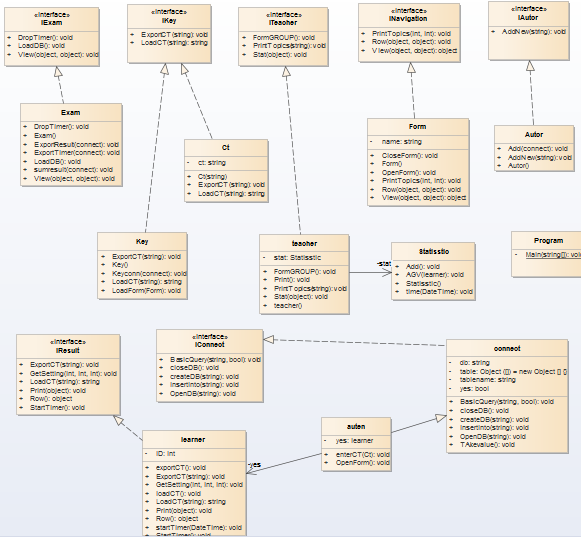


Рис.3. Классы кодирования

5.2 Классы кодирования

Классы кодирования программы соответствуют по функциональному назначению интерфейсам, по которым построены классы кодирования (см. п.4 Системные интерфейсы). Для системы так же разработаны следующие классы:

class learner : IResult

{

int ID;

public void loadCT() { }

public void exportCT() { }

public void startTimer(DateTime Time) { }

public void GetSetting(int ret, int level, int might) { }

public object Row() { return null; }

public void Print(object Row) { }

public string LoadCT(string KeyUser) { string a; a = ""; return a; }

public void ExportCT(string NewCT) { }

public void StartTimer() { }

}

class connect : IConnect

{

string db;

bool yes;

string tablename;

Object[] table = new Object [] {};

public void TAkevalue() { }

public void OpenDB(string p) { }

public void BasicQuery(string p, bool r) { }

public void createDB(string name) { }

public void InsertInto(string table) { }

public void closeDB() { }

}

class auten : connect

{

learner yes;

public void OpenForm() { }

public void enterCT(Ct i) { }

}

class Ct : IKey

{

string ct;

public Ct(string dB)

{

this.ct = dB;

}

public string LoadCT(string KeyUser) { return null; }

public void ExportCT(string NewCT) { }

}

class Exam : IExam

{

public Exam() { }

public void ExportTimer(connect db) { }

public void ExportResult(connect db) { }

public void sumresult(connect db) { }

public void LoadDB() { }

public void View(object Row, object theme) { }

public void DropTimer() { }

}

class Form : INavigation

{

string name;

public Form() { }

public void OpenForm() { }

public void CloseForm() { }

public void PrintTopics(int X, int Y) { }

public void Row(object Row, object Topics) { }

public object View(object Row, object theme) { return 0; }

}

class Animation

{

string name;

public Animation() { }

public void Play() { }

public void Stop() { }

}

class Key : IKey

{

public Key() { }

public void Keyconn(connect db) { }

public void LoadForm(Form name) { }

public string LoadCT(string KeyUser) { return null; }

public void ExportCT(string NewCT) { }

}

class Autor : IAutor

{

public Autor() { }

public void Add(connect table) { }

public void AddNew(string add) { }

}

class teacher : ITeacher

{

Statisstic stat;

public teacher() { }

public void Print() { }

public void PrintTopics(string LoadCT) { }

public void FormGROUP() { }

public void Stat(object group) { }

}

class Statisstic

{

public Statisstic() { }

public void Add() { }

public void AGV(learner yes) { }

public void time(DateTime time) { }

}