МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра «Вычислительная техника»

**Лабораторная работа №1**

по дисциплине: «Человек машина интерфейс».

Выполнил:

студент гр. ИВТАПбд-21

Кондратьев П.С.

Проверила:

Валюх В. В.

г. Ульяновск, 2018

**Программный продукт Burrows-Wheeler – архиватор**

**Цели создания продукта.**

Приложение, позволяющее производить сжатие и распаковку данных при помощи преобразований Move-to-front и Барроуза-Уилера, а также кодирования Хаффмана. Приложение представляет собой exe файл проекта, который вызывает консоль Visual Studio в которой производиться последующие действия с вашим файлом на компьютере.

При запуске exe файла появляется пользовательская консоль с интерфейсом. На первом этапе работы пользователю доступно только одно действие – выбор файла для загрузки из папки «Проект» на рабочем столе. Из выбранного файла считывается байты. Выбранный файл программа кодирует по этапно (в 3 этапа). Помимо исходного файла будут генерироваться еще файлы трех кодировок, а именно кодирование и декодирование классов MTF, BWT, HUFFMAN

**Предметная область и сфера применения программного продукта**

**2.1. Классический вариант в форме спецификаций**

Основным объектом предметной области являются структуры данных map, list, vector. Перечислим свойства объектов:

– количество элементов,

– тип данных,

– адреса ячеек памяти

– приоритет

Предполагается, что с данным приложением будут работать студенты ФИСТ и преподаватели, и они должны иметь возможность решать с её помощью следующие задачи:

1. Генерирование юнит тесты

2. Проверку контрольных сумм файла

3. Многократный запуск теста с увеличением размера обрабатываемых входных данных.

4. Расчет среднего времени выполнения операций

**2.2 В форме пользовательских историй**

Пожелание пользователя приложения:

1. Как преподавателю, использующему данное приложения, мне удобно видеть результаты уровня сжатия алгоритмов в виде таблицы, в которой отражается размеры файла до и после сжатия.

2. Как пользователю приложения мне необходимо иметь возможность увидеть зависимость скорости выполнения операций от размера файлов

3. Как студенту, использующему данное приложение я хочу иметь возможность увидеть наглядно преимущества структур данных map, list, vector для быстрой работы данных алгоритмов.

**3. Целевая аудитория**

1. Преподаватели ФИСТ

Преподаватели имеют возможность использовать данное приложение для демонстрации студентам при изучении темы: «Структуры данных».

2. Студенты ФИСТ 1-2 курсов.

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «Структуры данных».

**4. Описательная модель пользователя (профиль).**

Пользователь ***студент***:

- Возраст 19-21 г.

- Род занятий: студент

- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

- Задачи, решаемые пользователем: сравнение производительности структур данных map, list, vector.

Пользователь ***преподаватель***:

- Возраст 25-50 г.

- Род занятий: преподаватель

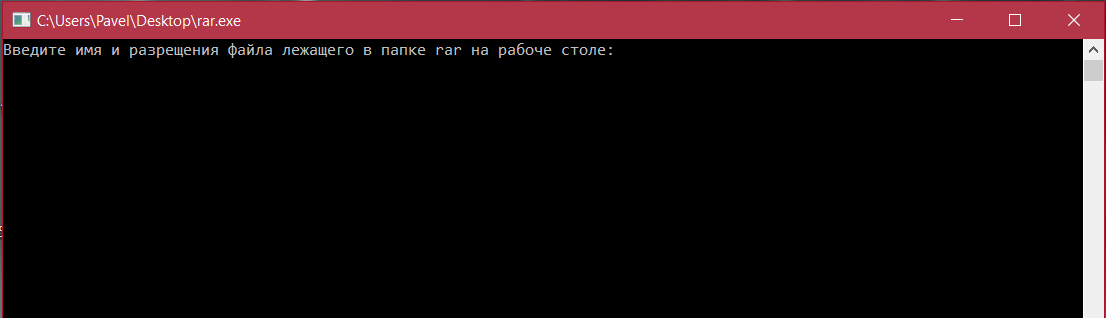
- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

- Задачи, решаемые пользователем: демонстрация на лекциях и практических занятиях сравнения быстродействия выполнения основных операций над структурами данных map, list, vector демонстрация сжатия файла с использованием этих структур, умение работать с битами.

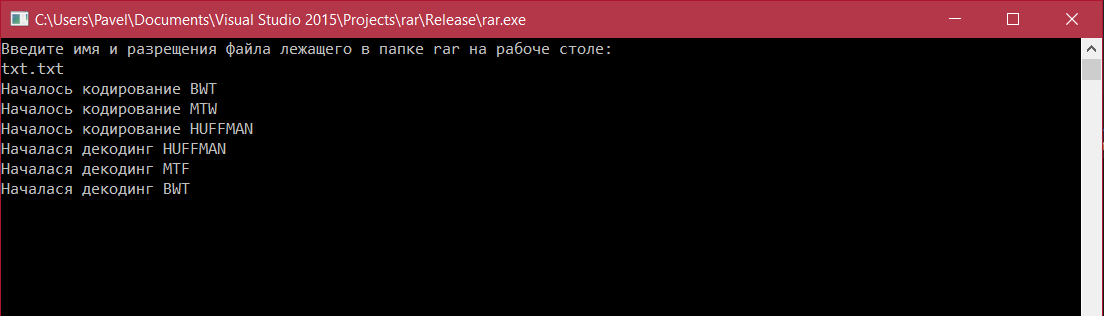
**5. Сценарии поведения пользователей на основании составленной модели.**

**5.1.** Сценарий запуска приложения пользователем студент:

1. При запуске программы (rar.exe), который находиться на рабочем столе, появляется консольное окно.



1. В котором пользователь указывает название и расширение файла который они хотя зашифровать. После чего как файл был введен, пользователь увидит сначала поэтапное шифрование после чего поэтапное дешифрование его файла.

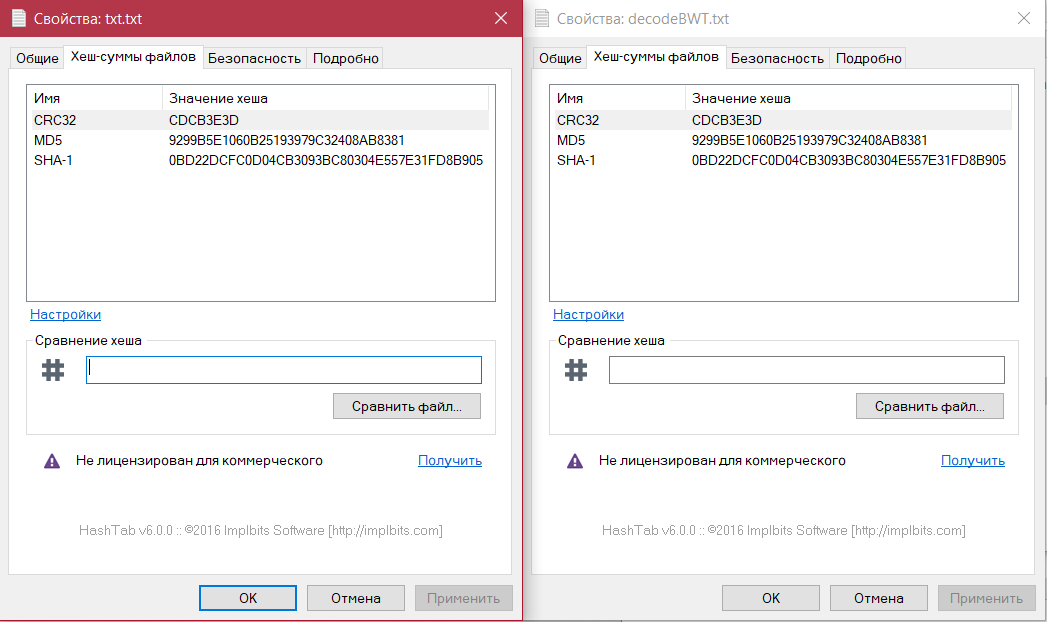


1. Результат проделанной программы можно посмотреть в папке rar, которая находиться на рабочем столе. Где было создано кодирование и декодирование каждого из алгоритмов архиватора. Файл с именем encodeHUFFMAN.bzip2 является закодированным файлом, а decodeBWT.txt - раскодированным.

**5.2** Сценарий запуска приложения пользователем преподаватель (демонстрация хеш сумм, юнит тестов и различных расширений файла):

1. Подключить файл #include "catch.hpp" в теле программы, для включения юнит тестов
2. При запуске программы (rar.exe), который находиться на рабочем столе, появляется консольное окно.
3. В котором пользователь указывает название и расширение файла который они хотя зашифровать. После чего как файл был введен, пользователь увидит сначала поэтапное шифрование после чего поэтапное дешифрование его файла.
4. Результат проделанной программы можно посмотреть в папке rar, которая находиться на рабочем столе. Где было создано кодирование и декодирование каждого из алгоритмов архиватора.

**5.3** Сценарий запуска приложения пользователем преподавателем (сравнение хэш сумм)



**5.4** Сценарий запуска приложения пользователем преподавателем (демонстрация хранения элементов структур данных)

1. Вызвать методы printTree, printable в теле main, чтобы увидеть наглядно закодированный файл в виде дерева (двоичной таблицы кодов)

**5.5** Сценарий запуска приложения пользователем преподавателем (выбор операций кодирования)

1. При запуске программы выбрать (функции кодирования): 1- кодирование методом BWT, 2 - кодирование методом MTW, 3 - кодирование методом HUFFMAN

**6. Функциональные блоки приложения и схему навигации между ними (структуру диалога)**

Отдельные функциональные блоки соответствуют работе пользователей с приложением:

- по режимам работы (функции 1,2,3): 1- выбор файла для кодирования 2- демонстрация быстродействия операций 3- демонстрация хранения элементов структур данных;

- по операциям (функции 4,5,6,7): 4- кодирование методом BWT, 5- кодирование методом MTW, 6- кодирование методом HUFFMAN;

В приложении можно выделить семь функциональных блоков:

- Режим работы;

- Выбор файла;

- кодирование методом BWT;

- кодирование методом MTW;

- кодирование методом HUFFMAN;

- Исследование хранения элементов контейнеров: map, list, vector в оперативной памяти компьютера;

- Вывод результатов

Режимы работы 1,2,3

1. Выбор файла для кодирования 4,5,6

2. Демонстрация быстродействия операций

3. Демонстрация хранения элементов структур данных

4. Кодирование BWT

5. Кодирование MTW

6. Кодирование HUFFMAN

7. Исследование хранения элементов контейнеров: map, list, vector в оперативной памяти компьютера

**Пользователь**

Вывод результатов

Рис. 1. Схема навигации

**7. Конечная концептуальная модель интерфейса, с применением UML нотации.**

Выбор файла

Пользователь

Режим работы

Студент

Преподаватель

демонстрация хранения элементов

демонстрация быстродействия операций

Выбор операции для кодирования

Кодирование MTW

Кодирование BWT

Кодирование HUFFMAN

**Программный продукт Игра "Змейка"**

**Цели создания продукта.**

Развлечение и отдых. Приложение, позволяющее сыграть в знаменитую игру - Змейка. При запуске игры требуется согласие: «Вы действительно хотите начать игру?" Если ДА, то игра запускается, иначе программа вылетает к коду.

По полю бегает змейка и поедает яблоки (зеленые кружки), и растет в длину + добавляются очки к счету. Цель игры - накормить змейку до максимальной длинны, и максимального счета.

1)Если змейка съедает кусочек своего хвоста, то она уменьшается на количество, которое съела.

2)Если змейка выходит за границы игры, то она уменьшается до длинны равной 2 клеткам и движется в противоположную сторону.

3)Если змейка съест испорченное яблоко (красные кружки), то счет уменьшается и змейка уменьшается на 3 клетки. (если длинна змейки меньше 4, то только уменьшается счет)

Игра заканчивается, когда у змейки останется 0 жизней. (Вылетает диалоговое окно с результатом: 1) Счет 2) Вы проиграли 3) максимальная длинна змейки) ... При нажатии ОК игра вылетает к коду.

**Предметная область и сфера применения программного продукта**

**2.1. Классический вариант в форме спецификаций**

Основным объектом предметной области является структура данных array. Перечислим свойства объектов:

– количество элементов,

– тип данных,

– адреса ячеек памяти

Предполагается, что с данным приложением будут работать студенты и преподаватели, и они должны иметь возможность улучшать данную программу:

1. Генерирование новых карт

2. Делать оптимизацию кода

3. Многократный запуск теста с увеличением размера змейки

4. Расчет среднего времени появления бонусов

**2.2 В форме пользовательских историй**

Пожелание пользователя приложения:

1. Как преподавателю, использующему данное приложения, мне удобно видеть результаты уровня в виде таблицы, в которой отражается максимальный размер змейки, жизни и счет, также таблица таймера, для бонусов.

2. Как пользователю приложения мне необходимо иметь возможность увидеть таблицу рекордов

3. Как студенту, использующему данное приложение я хочу иметь возможность дорабатывать этот проект и делать игру интереснее.

**3. Целевая аудитория**

1. Преподаватели ФИСТ

Преподаватели имеют возможность использовать данное приложение для демонстрации студентам визуализации работы WIN32API.

2. Студенты 1-2 курсов.

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «WIN32API», а также для развлечения.

**4. Описательная модель пользователя (профиль).**

Пользователь:

- Возраст любой.

- ***Примечание для студентов***: Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

Пользователь ***преподаватель***:

- Возраст 25-50 г.

- Род занятий: преподаватель

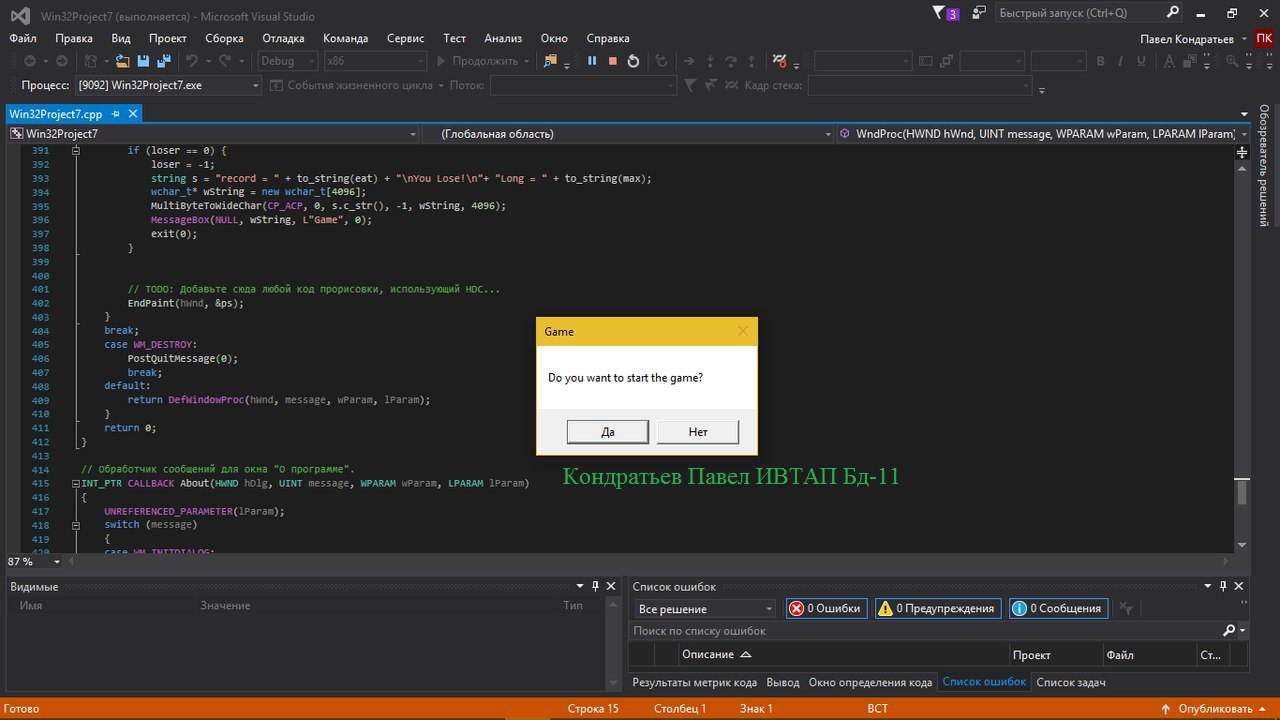
- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

- Задачи, решаемые пользователем: демонстрация на лекциях и практических занятиях сравнения быстродействия выполнения основных операций над структурой данных array демонстрация работы программного продукта, работа с WIN32API.

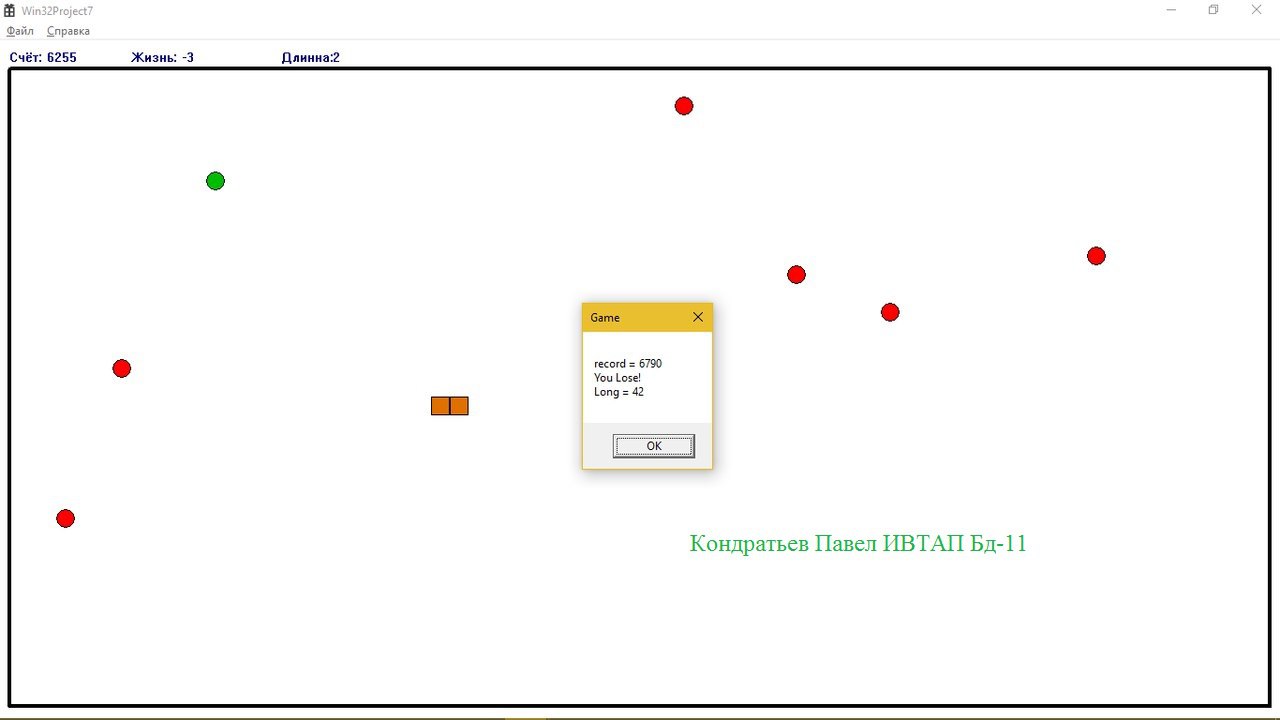
**5. Сценарии поведения пользователей на основании составленной модели.**

**5.1.** Сценарий запуска приложения пользователем студент:

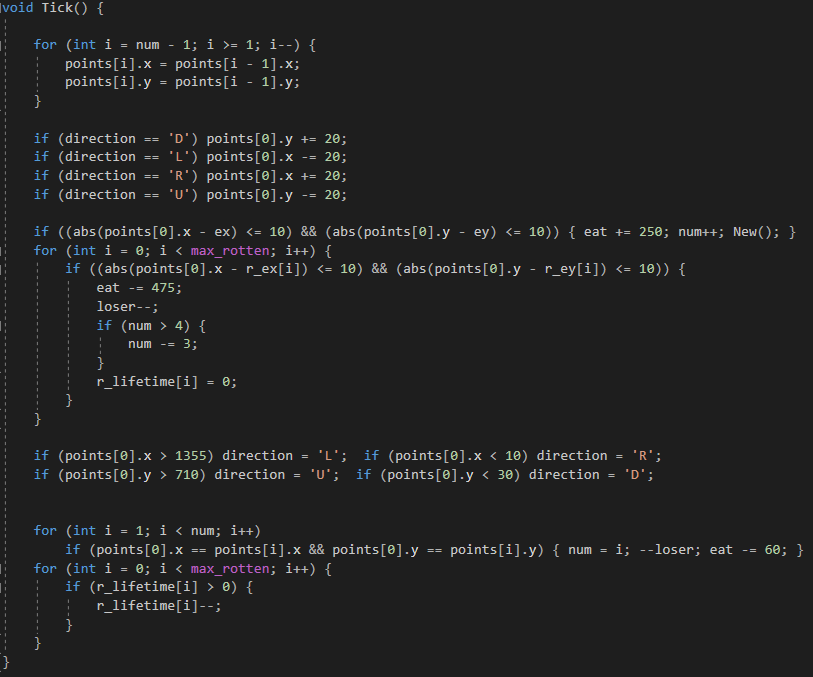
1. При запуске игры требуется согласие: «Вы действительно хотите начать игру?" Если ДА, то игра запускается, иначе программа вылетает к коду.



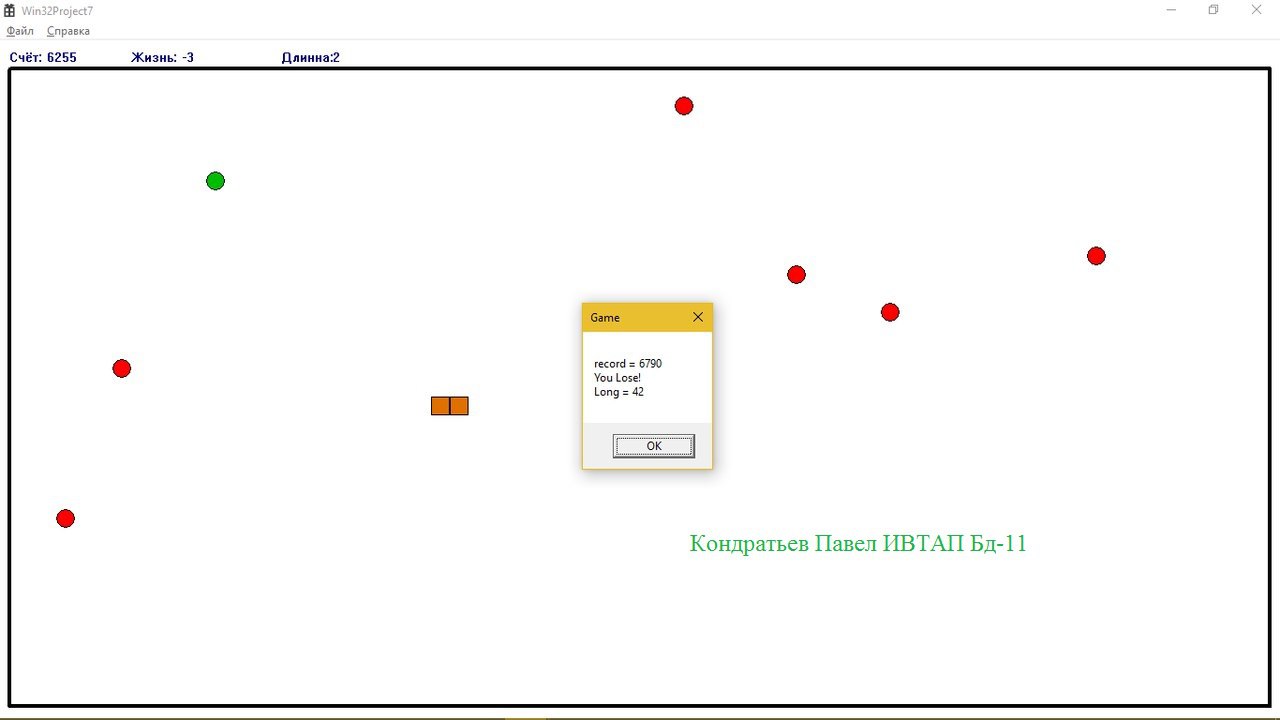
1. По полю бегает змейка и поедает яблоки (зеленые кружки), и растет в длину + добавляются очки к счету. Цель игры - накормить змейку до максимальной длинны, и максимального счета.
2. Игра заканчивается, когда у змейки останется 0 жизней. (Вылетает диалоговое окно с результатом: 1) Счет 2) Вы проиграли 3) максимальная длинна змейки) ... При нажатии ОК игра вылетает к коду.



**5.2** Сценарий запуска приложения пользователем преподаватель (демонстрация движение змейки, и изменение ее свойств в главном методе Tick()):



1. При запуске игры требуется согласие: «Вы действительно хотите начать игру?" Если ДА, то игра запускается, иначе программа вылетает к коду.
2. По полю бегает змейка и поедает яблоки (зеленые кружки), и растет в длину + добавляются очки к счету. Цель игры - накормить змейку до максимальной длинны, и максимального счета.
3. Игра заканчивается, когда у змейки останется 0 жизней. (Вылетает диалоговое окно с результатом: 1) Счет 2) Вы проиграли 3) максимальная длинна змейки) ... При нажатии ОК игра вылетает к коду.



**5.3** Сценарий запуска приложения пользователем преподаватель (изменение начальных значения)

1. Перед запуском задать начальные параметры змейки в файле source.cpp (параметры: direction, points, num, sum, record, heart, eat)

**5.4** Сценарий запуска приложения пользователем преподаватель (убрать ограничения змейки)

1. Перед запуском программы, в методе Tick(), закомментировать условие на съедание змейки самой себя

if (points[0].x > 1355) direction = 'L'; if (points[0].x < 10) direction = 'R';

if (points[0].y > 710) direction = 'U'; if (points[0].y < 30) direction = 'D';

for (int i = 1; i < num; i++)

if (points[0].x == points[i].x && points[0].y == points[i].y) { num = i; --loser; eat -= 60; }

for (int i = 0; i < max\_rotten; i++) {

if (r\_lifetime[i] > 0) {

r\_lifetime[i]--;

}

}

**5.5** Сценарий запуска приложения пользователем преподаватель (работа с таймерами)

1. Перед запуском программы в методах Newf, New, srand(time(0)), WM\_TIMER, есть возможность настроить таймера у яблок (бонусов игры), изенить их скорость появления и зависимость появления друг от друга.

**6. Функциональные блоки приложения и схему навигации между ними (структуру диалога)**

Отдельные функциональные блоки соответствуют работе пользователей с приложением:

- по режимам работы (функции 1,2): 1- выбор свойств для змейки 2- Демонстрация графической игры;

- по операциям (функции 4,5): 4- Исходный код (Исследование хранения элементов контейнера array), 5- WIN32API;

В приложении можно выделить пять функциональных блоков:

- Режим работы;

- Выбор свойств для змейки;

- Исходный код для улучшения методов (Исследование хранения элементов контейнера array);

- WIN32API;

- Вывод результатов

Режимы работы 1,2

1. Выбор свойств для змейки

2. Демонстрация графической игры

4. Исходный код (Исследование хранения элементов контейнера array)

5. WIN32API

**Пользователь**

Вывод результатов

Рис. 1. Схема навигации

**7. Конечная концептуальная модель интерфейса, с применением UML нотации.**

Демонстрация графической игры

Пользователь

Режим работы

Студент

Преподаватель

Демонстрация графической игры

Исходный код (Исследование хранения элементов контейнера array)

Выбор свойств для запуска (змейка)

WIN32API

Исходный код (Исследование хранения элементов контейнера array)

**Программный продукт Телефонный справочник**

**Цели создания продукта.**

Приложение, позволяющее производить добавление, удаление и поиск пользователя. Программа должна быть реализована с собственной панелью MyPanel, на которой выполняются действия над содержанием двусвязного списка, который реализуется через стандартные функции ArrayList. Программа должна при запуске выводить меню (в MyPanel) следующего содержания (К каждому пункту меню реализовать соответствующую функцию его названию):

1.Показать справочник

2.Добавить запись

3.Найти по номеру

4.Удалить по номеру

5.Очистить справочник

6.Выход

**Предметная область и сфера применения программного продукта**

**2.1. Классический вариант в форме спецификаций**

Основным объектом предметной области является структура данных list. Перечислим свойства объекта:

– количество элементов,

– тип данных,

– адреса ячеек памяти

– ссылки

**2.2 В форме пользовательских историй**

Пожелание пользователя приложения:

1. Как преподавателю, использующему данное приложения, мне удобно видеть результаты добавления пользователя.

2. Как пользователю приложения мне необходимо иметь возможность найти пользователя

3. Как студенту, использующему данное приложение я хочу иметь возможность дорабатывать этот проект и добавить изменение данных.

**3. Целевая аудитория**

1. Преподаватели ФИСТ

Преподаватели имеют возможность использовать данное приложение для демонстрации студентам визуализации работы языка Java с использованием собственной панелью MyPanel.

2. Студенты 1 курса.

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «Списки на java».

**4. Описательная модель пользователя (профиль).**

- Возраст 19-21 год.

- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

Пользователь ***преподаватель***:

- Возраст 25-50 г.

- Род занятий: преподаватель

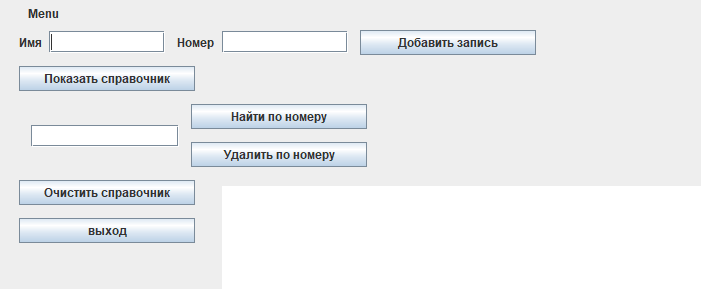
- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования MS Visual Studio

- Задачи, решаемые пользователем: демонстрация на лекциях и практических занятиях сравнения быстродействия выполнения основных операций над структурой данных list демонстрация работы программного продукта, работа со списки на java.

**5. Сценарии поведения пользователей на основании составленной модели.**

**5.1.** Сценарий запуска приложения пользователем студент:

1. Запустить проект в Visual Studio
2. Появиться окно MyPanel



**5.2** Сценарий запуска приложения пользователем студент (Button - Показать справочник)

1) Запустить проект в Visual Studio

2) Метод Button2\_click отвечает за показ справочника (коллекции листа), выполнен в классическом стиле classic form

**5.3** Сценарий запуска приложения пользователем студент (выбор свойств коллекции)

1) Изменение перед запуском программы свойств коллекции в классе list (добавление, изменение, удаление) полей этого класса.

**5.4** Сценарий запуска приложения пользователем студент (выход из телефонной книги)

1) Использовать кнопку Выход на форме MyPanel

2) Использовать x на форме MyPanel

**5.5** Сценарий запуска приложения пользователем студент (удаление пользователя)

1) Запустить проект в Visual Studio

2) Метод Button4\_click отвечает за удаление пользователя из справочника (коллекции листа), выполнен в классическом стиле classic form (удаление производиться в коллекции list с последующими действиями после удаления, т.е. перекидывание ссылок на последующий и предыдущий участок памяти с данных)

**6. Функциональные блоки приложения и схему навигации между ними (структуру диалога)**

Отдельные функциональные блоки соответствуют работе пользователей с приложением:

- по режимам работы (функции 1,2): 1-2 выбор свойств коллекции, Демонстрация формы телефонной книги;

- по операциям (функции 4,5): 4- Исходный код (Исследование хранения элементов контейнера list), 5- MyPanel;

В приложении можно выделить пять функциональных блоков:

- Режим работы;

- Выбор свойств коллекции;

- Исходный код для улучшения методов (Исследование хранения элементов контейнера list);

- MyPanel;

- Вывод результатов

Режим работы

1-2. Демонстрация формы телефонной книги

4. Исходный код (Исследование хранения элементов контейнера list)

5. MyPanel

**Пользователь**

Вывод результатов

Рис. 1. Схема навигации

**7. Конечная концептуальная модель интерфейса, с применением UML нотации.**

Пользователь

Режим работы

Студент

Преподаватель

Демонстрация графической игры

Выбор свойств для запуска (змейка)

MyPanel

Исходный код (Исследование хранения элементов контейнера list)

**Программный продукт сайт – Musfe**

**Цели создания продукта.**

Приложение, позволяющее работать с пользователями (обратный ответ) с использованием баз данных(sql). Сайт представляет собой главную страницу (регистрация) в который нужно пройти 2 этапа регистрации с загрузкой фотографии на сервер. После производиться переадресация на страницу пользователя (админа) где можно будет либо оставить ответ (письмо админу), либо заняться настройкой пользователей (для админа).

**Предметная область и сфера применения программного продукта**

**2.1. Классический вариант в форме спецификаций**

Основным объектом предметной области являются языки программирования php, javascript, также mysql, phpmyadmin.

Предполагается, что с данным приложением будут работать студенты ФИСТ и преподаватели, и они должны иметь возможность решать с её помощью следующие задачи:

1. базовые знания php, javascript, mysql

2. Проверка защиты пользователя

3. Оптимизация сложных sql запросов

4. Расчет среднего времени выполнения запросов

5. Любовь к комментариям в коде и желание разбираться в сложных по архитектуре системах

**2.2 В форме пользовательских историй**

Пожелание пользователя приложения:

1. Как преподавателю, использующему данное приложения, мне удобно видеть оптимизированный и развитый готовый программный продукт с таблицей пользователей сайта, массивов ошибок, exceptions, если другой пользователь хочет зайти на чужой аккаунт.

2. Как пользователю приложения мне необходимо иметь возможность изменить свою фотографию, логин, пароль

3. Как студенту, использующему данное приложение я хочу научиться базовым знания php, javascript, mysql, для того чтобы их применить для бедующего проекта.

**3. Целевая аудитория**

1. Преподаватели ФИСТ

Преподаватели имеют возможность использовать данное приложение для базового курса php.

2. Студенты ФИСТ

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «web программирование».

**4. Описательная модель пользователя (профиль).**

Пользователь ***студент***:

- Возраст 19-24 г.

- Род занятий: студент

- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в разной среде программирования

- Задачи, решаемые пользователем: работа с СУБД и сайтостроение.

Пользователь ***преподаватель***:

- Возраст 25-50 г.

- Род занятий: преподаватель

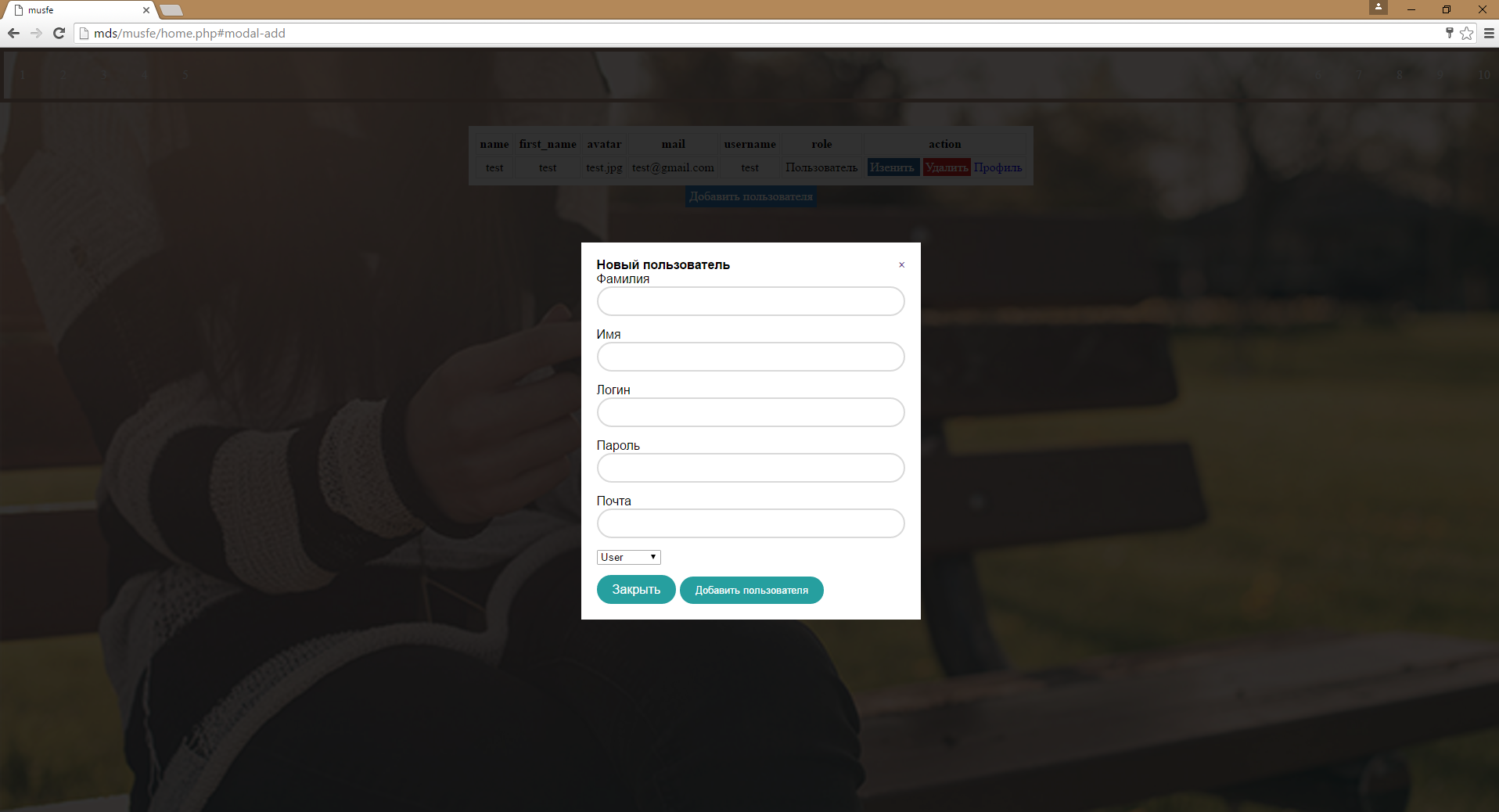
- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в разной среде программирования

-знание продвинутого курса php.

**5. Сценарии поведения пользователей на основании составленной модели.**

* 1. Запустить локальный сервер (Openserver)
  2. Произвести регистрацию, пройдя все 2 этапа
  3. Войти в профиль (admin)
  4. В профиле будет доступно изменение, добавление, удаление пользователя (для каждого действия реализовано свое модальное окно)
  5. Использование phpmyadmin





**6. Функциональные блоки приложения и схему навигации между ними (структуру диалога)**

Отдельные функциональные блоки соответствуют работе пользователей с приложением:

- по режимам работы (функции 1,2): 1- вход под ролью admin 2- вход под ролью user

- по операциям (функции 3,4): 3- добавление, удаление, изменение пользователя, 4- изменение фото, логина, пароля;

В приложении можно выделить пять функциональных блоков:

- Режим работы;

- Выбор входа;

- добавление, удаление, изменение пользователя;

- изменение фото, логина, пароля;

- Вывод результатов

Режимы работы 1,2

1. вход под ролью admin

2. вход под ролью user

3. добавление, удаление, изменение пользователя

4. изменение фото, логина, пароля

**Пользователь**

Вывод результатов

Рис. 1. Схема навигации

**7. Конечная концептуальная модель интерфейса, с применением UML нотации.**

Пользователь

Режим работы

Студент

Преподаватель

вход под ролью user

вход под ролью admin

Выбор входа

изменение фото, логина, пароля

добавление, удаление, изменение пользователя

**Программный продукт Работа с коллекциями с использованием WPF**

**Цели создания продукта.**

Приложение, позволяющее работать с коллекцией, в котором реализовано визуальное окно для работы с коллекцией. Приложение также имеет анимацию и имеет возможность производить действия с коллекцией (добавление, удаление, изменение)

**Предметная область и сфера применения программного продукта**

**2.1. Классический вариант в форме спецификаций**

Основным объектом предметной области являются язык программирования c# (реализация самой коллекции и работы с ней) и DoubleAnimation (анимация для модальных окон).

Предполагается, что с данным приложением будут работать студенты ФИСТ и преподаватели, и они должны иметь возможность решать с её помощью следующие задачи:

1. базовые знания с#

2. знание библиотеки DoubleAnimation

3. работа с коллекцией (list)

**2.2 В форме пользовательских историй**

Пожелание пользователя приложения:

1. Как преподавателю, использующему данное приложения, мне удобно видеть оптимизированную анимацию окон и правильную работу с коллекцией.

2. Как пользователю приложения мне необходимо иметь возможность видеть состояние коллекции

3. Как студенту, использующему данное приложение уметь изменять анимацию.

**3. Целевая аудитория**

1. Преподаватели ФИСТ

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «WPF».

2. Студенты ФИСТ

Студенты имеют возможность использовать данное приложения в целях самостоятельного изучения темы: «основы WFP».

**4. Описательная модель пользователя (профиль).**

Пользователь ***студент***:

- Возраст 19-21 г.

- Род занятий: студент

- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в среде программирования visual studio C#

- Задачи, решаемые пользователем: работа с коллекциями, анимация.

Пользователь ***преподаватель***:

- Возраст 25-50 г.

- Род занятий: преподаватель

- Уровень компьютерной грамотности: знание основ программирования, уверенная работа в разной среде программирования

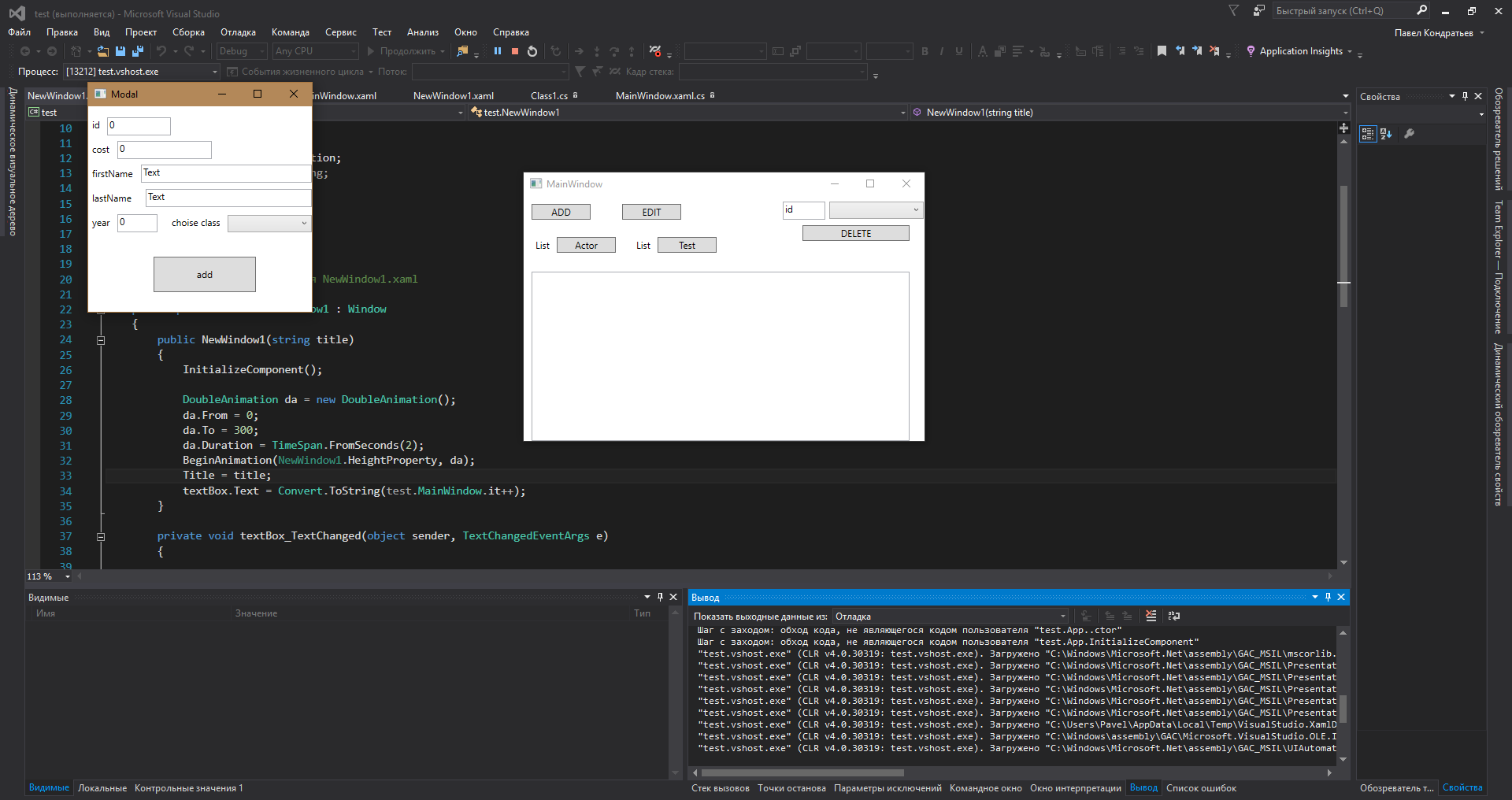
-знание продвинутого курса С#.

**5. Сценарии поведения пользователей на основании составленной модели.**

5.1 Производим запуск проекта в visual studio

5.2 Появляется главное окно (надстройка-действия коллекции)

* 1. Анимация окна добавления
  2. Добавление новой записи с автоинкрементном для пользователя
  3. Выбор действия над коллекцией



**6. Функциональные блоки приложения и схему навигации между ними (структуру диалога)**

Отдельные функциональные блоки соответствуют работе пользователей с приложением:

- по режимам работы (функции 1,2): 1- 2 выбор действия над коллекцией, анимация формы

- по операциям (функции 3,4): 3- добавление, удаление, изменение пользователя;

В приложении можно выделить пять функциональных блоков:

- Режим работы;

- Выбор действия;

- добавление, удаление, изменение пользователя коллекции;

- изменение анимации;

- Вывод результатов

Режим работы

1-2. выбор действия над коллекцией, анимация формы

3. добавление, удаление, изменение, поиск

**Пользователь**

Вывод результатов

Рис. 1. Схема навигации

**7. Конечная концептуальная модель интерфейса, с применением UML нотации.**

Пользователь

Режим работы

Студент

Преподаватель

выбор действия над коллекцией

Выбор действия

добавление, удаление, изменение, поиск