# РАБОЧИЙ ПРОЕКТ

## Общие сведения о работе программы

В результате выполнения курсовой работы была создана программа, позволяющая проводить автоматические тестирования на составление булевых выражений. В программе реализованы следующие алгоритмы:

1. генерации логических выражений;
2. генерация графической области, соответствующей логическому выражению;
3. сравнения введенного пользователем и сгенерированного выражения.

Программа разработана на языке программирования высокого уровня *C#* с использованием средств .NET Framework версии 3,5 и системы Windows Presentation Foundation (WPF) в интегрированной среде *Microsoft Visual Studio 2013* и работает под управлением операционных систем семейства Windows.

1. 1. **Функциональное назначение**

Программа обеспечивает следующие функциональные возможности:

* проведение тестирования на составление логических выражений;
* вывод результатов тестирования.

**Функциональные ограничения**

В базу шаблонов нельзя добавить новое задание или удалить существующее.

* 1. **Инсталляция и выполнение**

Для установки программного продукта необходимо запустить файл LogicSetup.exe и следовать подсказкам системы. После установки на рабочем столе появится ярлык «Проверка знания булевых выражений» (если было выбрано создание ярлыка) либо в установочной папке запустить файл «Проверка знания булевых выражений».

После инсталляции в папке программы появятся файлы, перечисленные в таблице 2.1.

Таблица 2.1 – Файлы программного продукта.

|  |  |
| --- | --- |
| **Файл** | **Назначение** |
| **Генерация логических задач.exe** | Исполняемый файл. |
| **statistic.html** | Файл статистики. |

Продолжение таблицы 2.1.

|  |  |
| --- | --- |
| **base.xml** | База шаблонов. |
| **help.html** | Файл справки. |
| **Папка img** | Содержит 3 изображения для справки. |

* 1. **Общий алгоритм работы программного продукта**

Блок-схема, демонстрирующая общий алгоритм работы программы, представлена в приложении на рисунке П.1.

Модули, реализованные в программе представлены в таблице 2.2.

Таблица 2.2 – Реализованные модули и классы.

|  |  |
| --- | --- |
| **Модуль** | **Класс** |
| MainWindow.xaml | MainWindow |
| MainWindow.xaml.cs |
| Test.xaml | Test |
| Test.xaml.cs |
| Result.xaml | Result |
| Result.xaml.cs |
| Help.xaml | Help |
| Help.xaml.cs |
| Statistic.xaml | Statistic |
| Statistic.xaml.cs |
| RPN.cs | RPN |
| TestChec.cs | TestChec |
| from\_XML\_to\_Task.cs | from\_XML\_to\_Task |
| Task.cs | Task |

Продолжение таблицы 2.2.

|  |  |
| --- | --- |
| Primitive.cs | Primitive |
| polyline.cs | polyline |
| poligon.cs | poligon |
| line.cs | line |
| ellips.cs | ellips |
| lable.cs | lable |
| point.cs | point |

Функциональное назначение модулей представлено в таблицах 2.3-2.9.

Модуль MainWindow*.xaml.cs* содержит класс MainWindow, который реализует логику работы модуля MainWindow*.xaml,* описывающего процесс навигации в главном меню.

Модуль *Test.xaml.cs* содержит класс *Test*, который реализует логику работы модуля *Test.xaml,* описывающего процесс прохождения тестирования.

В таблице 2.3 приведены методы класса *Test*.

Таблица 2.3 – Описание класса *Test*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **taskList** | **List<Task>** | | Список заданий. |
| **userName** | **string** | | Фамилия и имя тестируемого. |
| **fileAdress** | **string** | | Адрес базы шаблонов. |
| **Base** | **from\_XML\_to\_Task** | | Дешифратор базы. |
| **Прототип** | | **Назначение** | |
| **public void drawXOY(Canvas draw)** | | Рисует оси координат. | |

Модуль Result*.cs* содержит класс Result, которыйреализует вывод результата тестирования на экран с последующим сохранением его в файл статистики.

В таблице 2.4 приведены методы класса Result.

Таблица 2.4 – Описание методов класса Result

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **ChekTest** | **TestChek** | | Проверка теста. |
| **taskResult** | **List<bool>** | | Список результатов теста. |
| **answer** | **List<string[]>** | | Список ответов. |
| **userName** | **string** | | Фамилия и имя тетсируемого. |
| **Прототип** | | **Назначение** | |
| **private void addStatistic()** | | Добавляет статистику тестирования в файл статистики. | |

Модуль RPN*.cs* содержит класс RPN, которыйреализует перевод логического выражения из инфиксной записи в постфиксную.

В таблице 2.5 приведены методы класса RPN.

Таблица 2.5 – Описание методов класса RPN

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| **private bool isANumber(string value)** | Проверяет символ на принадлежность к целым числам. |
| **private string retEndDel(string returnString)** | Возвращает последний символ строки и удаляет его. |
| **public string inPostfix(string IN)** | Переводит строку в постфиксную запись. |

Модуль *Task.cs* содержит класс *Task*, которыйпредставляет из себя задание для тестирования.

В таблице 2.6 приведены методы класса *Task*.

Таблица 2.6 – Описание методов класса *Task*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **ID** | **int** | | Идентификационный номер. |
| **ListPrimitives** | **List<Primitive>** | | Список примитивов. |
| **Area** | **string** | | Область выделения |
| **points** | **bool[,]** | | Массив точек, которые необходимо закрашивать. |
| **UserAnswer** | **string** | | Ответ пользователя. |
| **EtalonAnswer** | **string** | | Эталонное решение. |
| **Прототип** | | **Назначение** | |
| **public void Draw(Canvas myCanvas)** | | Выводит на экран задания. | |

Модуль from\_XML\_to\_Task*.cs* содержит класс from\_XML\_to\_Task, которыйреализует разбор базы шаблонов на задания.

В таблице 2.7 приведены методы класса from\_XML\_to\_Task.

Таблица 2.7 – Описание методов класса from\_XML\_to\_Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **rand** | **Random** | | Случайное число. |
| **fileAdress** | **string** | | Адрес файла. |
| **Прототип** | | **Назначение** | |
| **public List<Task> Decode()** | | Декодирование файла с базой шаблонов. | |
| **private bool isInCircle(int x, int y, int x0, int y0, int r)** | | Проверка на принадлежность точки кругу. | |

Модуль TestChec*.cs* содержит класс TestChec, которыйреализует проверку введенных пользователем ответов.

В таблице 2.8 приведены методы класса TestChec.

Таблица 2.8 – Описание методов класса TestChec

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **newRPN** | **RPN** | | Переменная, переводящая логическое выражение из инфиксной записи в постфиксную. |
| **Прототип** | | **Назначение** | |
| **public bool getChek(string Etal, string User)** | | Сверяет строку эталонного решения с ответом пользователя. | |

Модуль Primitive*.cs* содержит базовый класс Primitive, описывающий модель примитива.

В таблице 2.9 приведены методы класса Primitive.

Таблица 2.9 – Описание методов класса Primitive

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | | **Назначение** |
| **ID** | **int** | | Идентификационный номер. |
| **LineWidth** | **int** | | Ширина линии. |
| **Width** | **int** | | Ширина примитива. |
| **Hieght** | **int** | | Высота примитива. |
| **R** | **int** | | Радиус круга. |
| **Coordinate** | **point** | | Центр круга. |
| **One** | **point** | | Первая точка линии. |
| **Two** | **point** | | Вторая точка линии. |
| **Прототип** | | **Назначение** | |
| **public virtual void Draw(Canvas myCanvas)** | | Выводит на экран примитив. | |
| **public virtual void Draw(Canvas myCanvas, bool lbl)** | | Выводит на экран примитив с подсказкой. | |

Продолжение таблицы 2.9.

|  |  |
| --- | --- |
| **public virtual void aDraw(ref bool[,] points)** | Выделяет область выше линии. |
| **public virtual void bDraw(ref bool[,] points)** | Выделяет область ниже линии. |
| **public virtual void iDraw(ref bool[,] points)** | Выделяет область внутри замкнутого примитива. |
| **public virtual void uDraw(ref bool[,] points)** | Выделяет область снаружи замкнутого примитива. |
| **public virtual bool Included(Point point)** | Проверяет принадлежность точки примитиву. |
| **public virtual string Above()** | Возвращает уравнение области, расположенной выше линии. |
| **public virtual string Below()** | Возвращает уравнение области, расположенной ниже линии. |
| **public virtual string Included()** | Возвращает уравнение области, расположенной внутри замкнутого примитива. |
| **public virtual string unIncluded()** | Возвращает уравнение области, расположенной вне замкнутого примитива. |

Классы line, polyline, polygon, label, point, ellipse наследуются от класса Primitive и используют методы, объявленные в нем.

Модуль Help*.xaml.cs* содержит класс Help, который реализует логику работы модуля Help*.xaml,* описывающий вывод справки по проведению тестирования.

* 1. **Разработанные меню и интерфейсы**

После запуска исполняемого файла программы, откроется главное окно.

Главное окно программы показаны на рисунке 2.1.

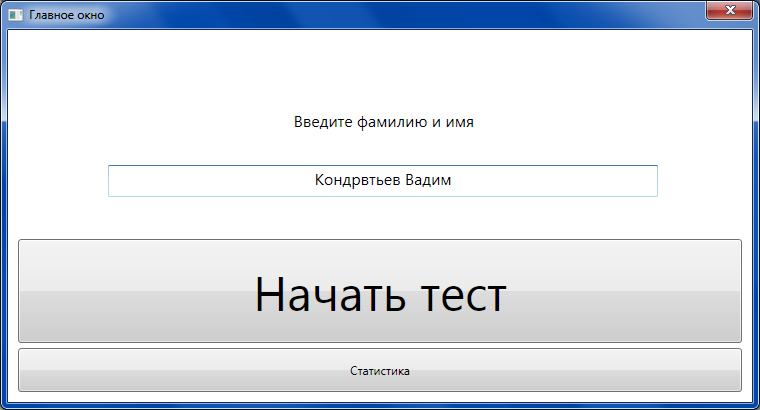


Рисунок 2.1 – Основное окно программы.

Основное меню программы содержит следующие кнопки:

1. кнопка перехода в режим тестирования;
2. кнопка перехода к окну статистики.

Чтобы начать тестирование, необходимо нажать кнопку «Начать тест». Чтобы перейти к окну статистики, необходимо нажать кнопку «Статистика».

Окно статистики представлено на рисунке 2.5.

После нажатия кнопки «Начать тестирование» пользователь попадает в окно режима тестирования, которое продемонстрированно на рисунке 2.2.

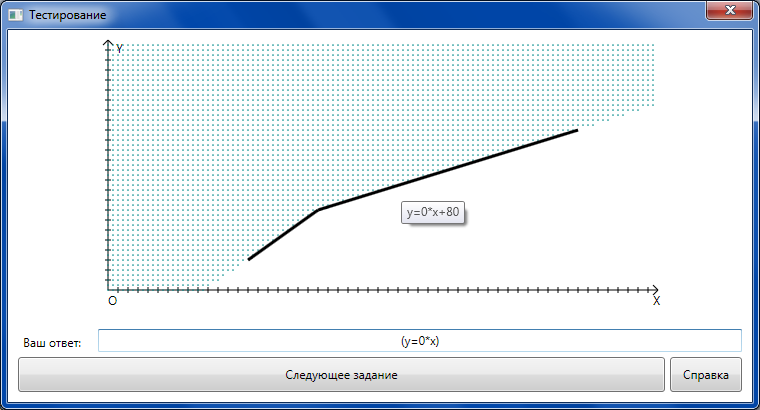


Рисунок 2.2 – Интерфейс окна режима тестирования

Окно режима тестирования содержит графическое представление области, подсказки по вводу функций, поле ввода ответа и кнопку следующего задания.

При нажатии на кнопку «Справка» откроется окно справки, представленное на рисунке 2.4.

После завершения теста появляется окно результата, показанное на рисунке 2.3.

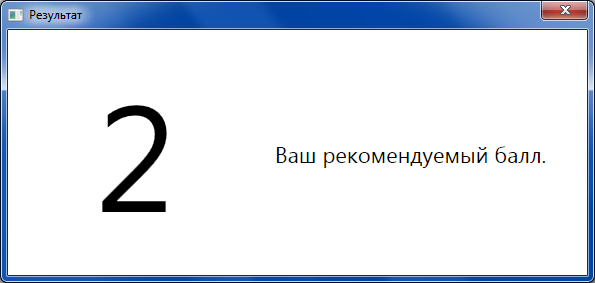


Рисунок 2.3 – Окно результата тестирования

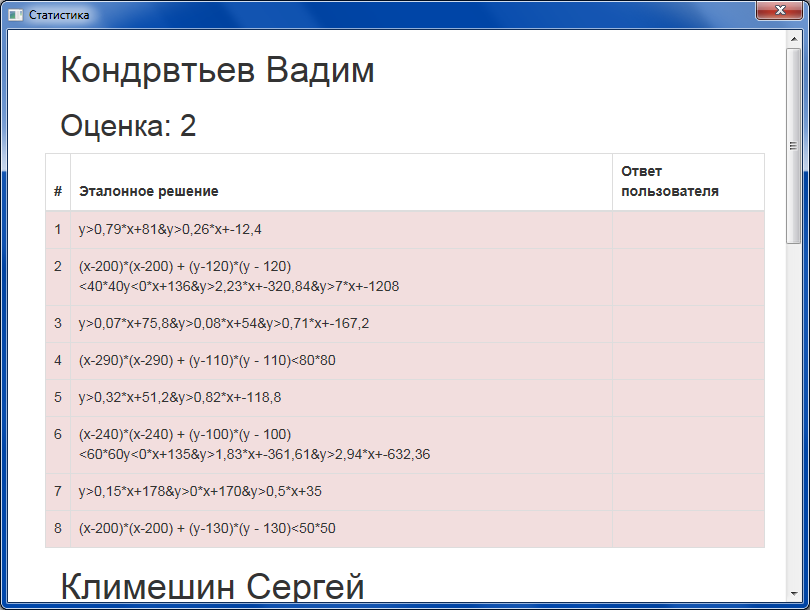


Рисунок 2.4 – Окно результатов тестирования с информацией о пройденных заданиях

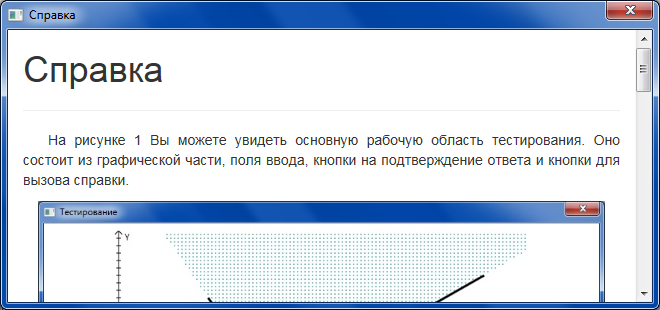


Рисунок 2.5 – Окно справки.

**ПРИЛОЖЕНИЕ**

**Блок-схемы и рисунки**



Рисунок П.1 – Блок-схема, демонстрирующая общий алгоритм работы программы