МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ Цифрового развития

КАФЕДРА информационных систем и технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

по дисциплине «Технологии программирования»

на тему:

Разработка приложения «Головоломка»

**Выполнил:**

Сусоев Дмитрий Сергеевич

студент 2 курса

группы ИНС-б-о-20-2

направления 09.03.02 Информационные системы и технологии

очной формы обучения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель работы:**

Николаев Е.И., доцент ИСТ

(ФИО, должность, кафедра)

Работа допущена к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя) (дата)

Работа выполнена и

защищена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель комиссии:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Зав. кафедрой ИСТ |  |  |  | В.И. Дроздова |
| (должность) |  | (подпись) |  | (И.О. Фамилия) |

Члены комиссии:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Доцент кафедры ИСТ |  |  |  |  |
| (должность) |  | (подпись) |  | (И.О. Фамилия) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Доцент кафедры ИСТ |  |  |  |  |
| (должность) |  | (подпись) |  | (И.О. Фамилия) |

Ставрополь, 2022

**Министерство науки и высшего образования российской федерации**

**Министерство науки и высшего образования российской федерации**

**ФГАОУ ВО «Северо-Кавказский федеральный университет»**

**Институт цифрового развития**

**Кафедра информационных систем и технологий**

**УТВЕРЖДАЮ**

Зав. кафедрой информационных систем и технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.И. Дроздова

*(подпись)*

« » 2022 г.

Институт цифрового развития   
Кафедра информационных систем и технологий

Направление 09.03.02 Информационные системы и технологии

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| студента | | Сусоева Дмитрия Сергеевича | | | |
|  | | *(фамилия, имя, отчество)* | | | |
| по дисциплине | | | | | Технологии программирования |
| **1. Тема работы** | | | | Разработка приложения «Головоломка» | |
| **2. Цель** | Разработка приложения на основе технологий .NET Framework для демонстрации механизма графического пароля. | | | | |
| **3. Задачи:** | | |  | | |
| 1. Разработать игровое приложение-головоломку: игровое поле разбито на равные квадратные клетки. В каждой клетке располагается вентиль с двумя состояниями. При переключении одного вентиля, своё состояние меняют на противоположные все вентили в данном столбце и данной строке. Игра завершается, если все вентили повернуты в одинаковое состояние.  2. Реализовать игровое поле с изменяемыми размерами.  3. Реализовать различные варианты оформления вентилей.  4. Реализовать описанный алгоритм работы приложения.  5. Реализовать механизм пользовательских настроек для приложения. | | | | | |

**4. Перечень подлежащих разработке вопросов:**

|  |
| --- |
| а) по теоретической части: архитектура приложений, построенных на основе технологии .NET Framework; основные библиотечные классы, используемые в процессе разработки приложения; основные процессы предметной области, подлежащие автоматизации.  б) по практической части: дизайн приложения; алгоритмы графического пароля; разработка многомодульного приложения; использование GDI+; использование классов System.IO; реализация механизма пользовательских настроек. |

**5. Исходные данные:**

|  |
| --- |
| Внешний вид приложения студент выбирает самостоятельно. В приложении реализовать: количество шаблонов вентиля: 3; игровое поле: конфигурируемое с максимальным размером 20x20; количество пользовательских настроек: более 5. |

**6. Критерии оценивания**

|  |  |
| --- | --- |
| Наименование критерия | Балл |
| Реализованы все задачи индивидуального задания | 20 |
| Выполнены все требования к исходным данным приложения | 20 |
| Выполнены требования к содержанию пояснительной записки | 20 |
| Выполнены требования к форматированию пояснительной записки | 10 |
| Презентация отражает все стадии работы над проектом | 10 |
| В ходе защиты студент отвечает на все вопросы, касающиеся программы и содержимого пояснительной записки | 20 |
| **Максимальное количество баллов:** | **100** |

**7. Шкала оценивания**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Баллы | Оценка |  | Баллы | Оценка |
| [90; 100] | Отлично |  | [60; 80) | Удовлетворительно |
| [80; 90) | Хорошо |  | менее 60 | Неудовлетворительно |

**8. Список рекомендуемой литературы**

1. Язык программирования C# 7 и платформы .NET и .NET Core (8-е издание) / Э. Троелсен. – М.: Диалектика, 2018. – 1330 с.

2. Учебное пособие по дисциплине «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 148 с.

3. Учебное пособие (лабораторный практикум) по дисциплине «Технологии программирования» для студентов специальности 09.03.02 "Информационные системы и технологии" / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 150 с.

4. Методические указания к выполнению курсовой работы по дисциплине «Технологии программирования» для студентов специальности 09.03.02 "Информационные системы и технологии" / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 25 с.

**9. Контрольные сроки:**

- предоставление работы на проверку «7» мая 2022 г.

- доработка программы, пояснительной записки «14» мая 2022 г.

- защита курсовой работы с презентацией «21» мая 2022 г.

Руководитель курсовой работы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | Е.И. Николаев |
| *(ученая степень, звание)* |  | *(личная подпись)* |  | *(инициалы, фамилия)* |

Задание принял(а) к исполнению студент(ка)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| *(дата)* |  | *(личная подпись)* |  | *(инициалы, фамилия)* |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc104498429)

[1 Разработка приложения «Головоломка» 7](#_Toc104498430)

[1.1 Общие сведения о проекте 7](#_Toc104498431)

[1.2 Функциональная структура проекта 12](#_Toc104498432)

[1.3 Описание логической структуры приложения 12](#_Toc104498433)

[2 Информационное и программное обеспечение 28](#_Toc104498434)

[2.1 Требования к техническому обеспечению 28](#_Toc104498435)

[2.2 Руководство пользователя 28](#_Toc104498436)

[2.3 Формат входных данных. Выходные данные 28](#_Toc104498437)

[ЗАКЛЮЧЕНИЕ 29](#_Toc104498438)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc104498439)

[ПРИЛОЖЕНИЕ 1. Диаграмма классов приложения 31](#_Toc104498440)

[ПРИЛОЖЕНИЕ 2. Исходный код файла Form1.cs 32](#_Toc104498441)

[ПРИЛОЖЕНИЕ 3. Исходный код файла Form1.Designer.cs 38](#_Toc104498442)

[ПРИЛОЖЕНИЕ 4. Исходный код файла Program.cs. 44](#_Toc104498443)

# ВВЕДЕНИЕ

Компьютерная головоломка - в играх этого поджанра повторяется геймплей обычных игр-головоломок: пасьянсов, маджонга, пятнашек и т. д. (возможно, с чисто компьютерными дополнениями вроде ограничения по времени).

В некоторых головоломках игроку даются случайные блоки или кусочки, которые требуется собрать в определённой последовательности и форме. К таким играм относятся Тетрис, Klax, Lumines. При этом Тетрис породил множество продолжений, вариаций и клонов с участием «падающих блоков». Некоторые из таких головоломок имеют режим игры, обратный принципу Тетриса: так, например, в Tetrisphere и Tetris Attack, игрок должен очистить площадь от деталей за ограниченное количество ходов.

Ещё одним представителем жанра является созданная в 1980 году Хироюки Имабаяси игра Sokoban, в которой игрок передвигает ящики по лабиринту с целью поставить их на заданные конечные позиции.

Многие квесты и экшен-адвенчуры содержат в себе элементы головоломки. К примеру, Resident Evil, Silent Hill, LittleBigPlanet и серия игр The Legend of Zelda.

Обычно головоломки не вызывают сложностей для распространения и адаптации, их можно встретить на аркадных автоматах, игровых приставках, карманных компьютерах, мобильных телефонах.

Целью же моей курсовой работы будет разработка игры – «Головоломка» с переключающимися вентилями. Написать программу, которая позволяет игроку выбирать вид вентиля, количество рядов и колонок. Программа начинается с кнопки «Новая Игра», при нажатии которой появится игровое поле соответствующее заданным настройкам. Положение вентилей управляется одиночным кликом по игровому полю.

Цель игры – перевести все вентили из «красного» состояния в «зеленое». При изменении положения одного вентиля, будут меняться положения всех вентилей в этой строке и колонне. Игра автоматически завершится, если все вентили переведены в «зеленое» состояние.

# 1 Разработка приложения «Головоломка»

## 1.1 Общие сведения о проекте

В курсовой работе представлена разработка приложения «Головоломка» с переключающимися вентилями.

В качестве среды разработки была выбрана линейка продуктов Microsoft Visual Studio и графический интерфейс Windows Forms. Windows Forms представляет собой событийно-ориентированное приложение, поддерживаемое Microsoft .NET Framework. В отличие от пакетных программ, большая часть времени тратится на ожидание от пользователя каких-либо действий, как, например, ввод текста в текстовое поле или клика мышкой по кнопке.

При разработке приложения использовались библиотеки и пространства имен, представленные в таблице 1.

Таблица 1 – Библиотеки и пространства имен.

|  |  |
| --- | --- |
| Свойство | Функция |
| **System** | Содержит фундаментальные и базовые классы, определяющие часто используемые типы значений и ссылочных данных, события и обработчики событий, интерфейсы, атрибуты и исключения обработки. |
| **System.Collections.Generic** | Содержит интерфейсы и классы, определяющие универсальные коллекции, которые позволяют пользователям создавать строго типизированные коллекции, обеспечивающие повышенную производительность и безопасность типов по сравнению с неуниверсальными строго типизированными коллекциями. |
| **System.ComponentModel** | Предоставляет классы, используемые для реализации поведения компонентов и элементов управления во время разработки и во время выполнения. Это пространство имен содержит базовые классы и интерфейсы для реализации атрибутов и преобразователей типов, привязки к источникам данных и лицензирования компонентов. |
| **System.Data** | Предоставляет доступ к классам, представляющим архитектуру ADO.NET. ADO.NET позволяет создавать компоненты, эффективно управляющие данными из нескольких источников данных. |
| **System.Drawing** | Предоставляет доступ к основным графическим функциям GDI+. System.Drawing.Drawing2DSystem.Drawing.Imaging |
| **System.Linq** | Предоставляет классы и интерфейсы, поддерживающие запросы с использованием LINQ. |
| **System.Text** | Содержит классы, которые представляют кодировки ASCII и Юникода; абстрактные базовые классы для преобразования блоков знаков в блоки байтов и обратно; вспомогательный класс, который обрабатывает и форматирует объекты String, не создавая промежуточные экземпляры String. |
| **System.Threading.Tasks** | Предоставляет типы, которые упрощают работу по написанию параллельного и асинхронного кода. Основные типы: Task, представляющий асинхронную операцию, которую можно ожидать и отменить, и Task<TResult>, представляющий собой задачу, которая может вернуть значение. Класс TaskFactory предоставляет статические методы для создания задач, а класс TaskScheduler предоставляет инфраструктуру планирования потоков по умолчанию. |
| **System.Windows.Forms** | Содержит классы для создания приложений Windows, которые позволяют наиболее эффективно использовать расширенные возможности пользовательского интерфейса, доступные в операционной системе Microsoft Windows. |
| **CourseWork** | Пространство самого приложения. |

Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, С++, так и на VB.Net, J# и др.

С одной стороны, Windows Forms рассматривается как замена более старой и сложной библиотеке MFC, изначально написанной на языке C++. С другой стороны, WF не предлагает парадигмы, сравнимой с MVC. Для исправления этой ситуации и реализации данной функциональности в WF существуют сторонние библиотеки. Одной из наиболее используемых подобных библиотек является User Interface Process Application Block, выпущенная специальной группой Microsoft, занимающейся примерами и рекомендациями, для бесплатного скачивания. Эта библиотека также содержит исходный код и обучающие примеры для ускорения обучения.

Внутри .NET Framework Windows Forms реализуется в рамках пространства имён System.Windows.Forms.

В процессе разработки игры мною активно были использованы методы и компоненты встроенные в Windows Forms.

Button - Представляет элемент управления "кнопка Windows".

В таблице 2 представлены свойства и их функции используемые в данной работе для компонента Button

Таблица 2 – Свойства и функции Button.

|  |  |
| --- | --- |
| Свойство | Функция |
| **Name** | Возвращает или задает имя элемента управления. |
| **Size** | Возвращает или задает высоту и ширину элемента управления. |
| **Location** | Возвращает или задает координаты левого верхнего угла элемента управления относительно левого верхнего угла его контейнера. |
| **Text** | Возвращает или задает текст, связанный с этим элементом управления. |
| **Click (Событие)** | Происходит при щелчке элемента управления. |

Функции и свойства не упомянутые в таблице 2 либо не используются, либо заданы по умолчанию.

Label - Представляет стандартную метку Windows.

В таблице 3 представлены свойства и их функции доступные для компонента Label.

Таблица 3 – Свойства и функции Label.

|  |  |
| --- | --- |
| Свойство | Функция |
| **Name** | Возвращает или задает имя элемента управления. |
| **Size** | Возвращает или задает высоту и ширину элемента управления. |
| **Location** | Возвращает или задает координаты левого верхнего угла элемента управления относительно левого верхнего угла его контейнера. |
| **Text** | Возвращает или задает текст, связанный с этим элементом управления. |

Функции и свойства не упомянутые в таблице 3 либо не используются, либо заданы по умолчанию.

ComboBox - Представляет элемент управления "поле со списком" Windows.

В таблице 4 представлены параметры и их свойства доступные для ComboBox

Таблица 4 – Параметры и свойства ComboBox.

|  |  |
| --- | --- |
| Свойство | Функция |
| **Name** | Возвращает или задает имя элемента управления. |
| **Size** | Возвращает или задает высоту и ширину элемента управления. |
| **Location** | Возвращает или задает координаты левого верхнего угла элемента управления относительно левого верхнего угла его контейнера. |
| **Text** | Возвращает или задает текст, связанный с этим элементом управления. |
| **Items** | Возвращает объект, представляющий коллекцию элементов, содержащихся в данном поле со списком ComboBox. |

Функции и свойства не упомянутые в таблице 4 либо не используются, либо заданы по умолчанию.

## 1.2 Функциональная структура проекта

Для создания данного проекта понадобится создать поле, в котором будут располагаться объекты типа Button, Label, ComboBox и собственно игровое поле с вентилями.

Для того, чтобы изменять ширину, высоту и визуальный стиль игрового поля, пользователю нужно будет выбрать их значения в объектах ComboBox и нажать на объект Button «Новая Игра».

Управление в приложении осуществляется польностью через левые щелчки мыши, поэтому используется событие Click для Button и событие MouseClick для самой формы.

## 1.3 Описание логической структуры приложения

При запуске игры происходит запуск формы GameForm, в которой расположены кнопка «Новая Игра», поля со списками для количеста строк и рядов игрового поля, а также стиля вентилей, и метки для них. В режиме игры это выглядит так (Рисунок 1).

|  |
| --- |
|  |
| Рисунок 1 – Приложение «Головоломка» |

На данной форме расположены:

- NewGame – объект типа Button с текстом «Новая Игра»;

- label1, label2 и label3 – объекты типа Label с текстами «Ряды», «Столбцы» и «Стиль» соответсвенно;

- GameObject Weapon\_0 – бластер;

- rowsBox, columnsBox и textureBox – объекты типа comboBox;

Все свойства вышеупомянутых объектов представлены на Рисунках 2 - 8.

|  |
| --- |
|  |
| Рисунок 2 – Свойства объекта Button «NewGame» |
| Рисунок 3 – Свойства объекта Label «label1»    Рисунок 4 – Свойства объекта Label «label2» |
| Рисунок 5 – Свойства объекта Label «label3» |
| Рисунок 6 – Свойства объекта ComboBox «rowsBox»    Рисунок 7 – Свойства объекта ComboBox «columnsBox»    Рисунок 8 – Свойства объекта ComboBox «textureBox»  Так же Visual Studio позволяет просмотреть структуру приложения в удобной форме (Рисунок 9).    Рисунок 9 – Структура приложения  Для работы приложения потребуется объявить несколько переменных: GaugeWidth и GaugeHeight – длина и ширина вентиля в пикселях, Columns и Rows – количество колонок и строк, GreenButtons – количество вентилей в «зеленом» состоянии, GameFieldStartX и GameFieldStartY – отступы по горизонтали и вертикали от левого верхнего угла окна в пикселях, GaugeStates – состояния вентилей, GameActive – проверка того, что игра была начата, redButtonTexture и greenButtonTexture – текстуры для положений вентиля и GFX – графика для приложения (Рисунок 10).    Рисунок 10 – Объявленные переменные  Теперь нужно создать событие Click для NewGame и присвоить значение GFX в конструкторе GameForm. В событии Click будет проиходить следующее: переменным Columns и Rows присваиваются значения с columnsBox и rowsBox, изменяется размер окна, чтобы уместить игровое поле в форме, массиву GaugeStates задается размер и присваиваются значения false, присваивается значение переменным redButtonTexture и greenButtonTexture в зависимости от выбора в textureBox. Затем игровое поле отрисовывается при помощи функции GFX.DrawImage и переменной GameActive присваивается значение true (Рисунок 11).    Рисунок 11 – Конструктор GameForm и событие Click  Далее требуется добавит событие MouseClick для GameForm. В нем, если игра начата и курсор игрока находится в пределах игрового поля, произойдет проверка на совпадение координат мыши в окне и координат вентилей по горизонтали и вертикали. Из этих совпадений в массив GaugeStates перезаписываются значения (true если после нажатия вентиль стал «зеленым» и false если «красным»), после чего вентили отрисовываются заново в соответствии с их состояниями. В конце события идет пересчет «зеленых» вентилей, и если их количество совпадает с общим числом вентилей, игра завершается, выводя победный экран (Рисунок 12).    Рисунок 12 – Событие MouseClick для GameForm  Теперь, при запуске приложения и нажатии кнопки «Новая Игра», пользователь будет видеть следующее (Рисунок 13).    Рисунок 13 – Новая игра с полем 13\*17 и обычным стилем вентилей  А после нажатия на какой-либо вентиль произойдет следующее (Рисунок 14).    Рисунок 14 – Игровое поле после переключения вентиля  Если же игрок сумеет перевести все вентили в «зеленое» состояние, игра завершится и сообщит ему о победе (Рисунок 15).    Рисунок 15 – Завершенная игра  Стили разработанные для вентилей представлены в таблице 5.  Таблица 5 – Стили вентилей.   |  |  |  | | --- | --- | --- | | Стиль | «Зеленый» вентиль | «Красный» вентиль | | Обычный | C:\Users\azzri\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GaugeGreen.png | D:\ConsoleApps\source\repos\CourseWork\CourseWork\bin\Debug\GaugeRed.png | | Красивый | D:\ConsoleApps\source\repos\CourseWork\CourseWork\bin\Debug\ButtonGreen.png | D:\ConsoleApps\source\repos\CourseWork\CourseWork\bin\Debug\ButtonRed.png | | Упрощенный | D:\ConsoleApps\source\repos\CourseWork\CourseWork\bin\Debug\MarkGreen.png | D:\ConsoleApps\source\repos\CourseWork\CourseWork\bin\Debug\MarkRed.png |   Программный код всех компонентов приложения можно найти в ПРИЛОЖЕНИЯХ 2 - 4. |
|  |

# 2 Информационное и программное обеспечение

## 2.1 Требования к техническому обеспечению

Для обеспечения нормальной работоспособности программы необходим компьютер с конфигурацией не ниже:

* Процессор с тактовой частотой 1.6 ГГц;
* Оперативная память объёмом 2 Гб;
* Свободное место на диске – 500 Кб;
* Монитор;
* Клавиатура;
* Мышь.

Данная программа работает под управлением семейства Windows.

## 2.2 Руководство пользователя

Программа представляет собой проект, созданный в Visual Studio, который можно запускать в любой операционной системе семейства Windows.

При запуске программы открывается окно, в котором есть ячейки с выбором размеров поля и стилем вентилей, а также кнопка запуска игры. После начала игры пользователю нужно нажимать ЛКМ по вентилю, который он хочет переключить, при этом все вентили в этой строке и столбце тоже изменят свои состояния. Игра завершится, если пользователь сможет перевести все вентили в «зеленое» состояние.

## 2.3 Формат входных данных. Выходные данные

Выходные данными в данной программе отсутствуют.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы были выполнены все поставленные цели и задачи. А именно: реализовано игровое поле с пользовательскими настройками длины и ширины, и несколько стилей вентилей.

Разработка игры оказалось очень увлекательным делом, из которого я вынес много полезной информации в сфере разработки игр, с которой в будущем я хочу связать свою профессиональную деятельность. Надеюсь, что опыт полученный в процессе разработки игры мне пригодится. Среда разработки Visual Studio оказалась проще, чем кажется, вся нужная документация есть в открытом доступе в интернете.

# СПИСОК ЛИТЕРАТУРЫ

1. Троелсен, A. C# 6.0 and the .NET 4.6 Framework / A. Troelsen, P. Japikse. - М. : «apress», 2015. - 1625 c.

2. Робинсон, С. C# для профессионалов. Том 1. / С. Робинсон, О. Корнес, Д. Глин, Б. Харвей. – М. :Лори, 2010. – 1002 с.

3. Робинсон, С. C# для профессионалов. Том 2. / С. Робинсон, О. Корнес, Д. Глин, Б. Харвей. – М. :Лори, 2010. – 998 с.

4. Шилдт, Г. C#, учебный курс. / Г. Шилдт. – СПб. : Питер, 2010. – 512 с.

5. Нортроп, Т. Основы разработки приложений на платформе Microsoft .NET Framework. Учебный курс Microsoft. Перевод с англ./ Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – М. : «Русская редакция», 2010. – 864 с.

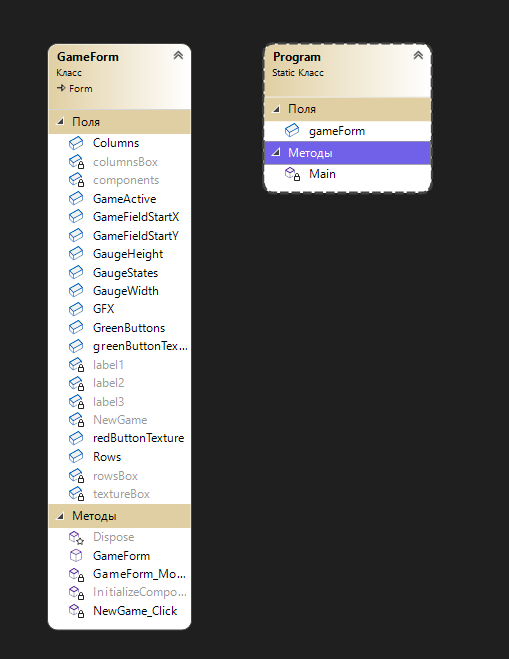
6. Язык программирования C# 5.0 и платформа .NET 4.5 (6-е издание) / Э. Троелсен. – М.: Вильямс, 2015. – 1311 с.

7. Учебное пособие по дисциплине «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И. – Ставрополь: СКФУ, 2016. – 148 с.

8. Учебное пособие (лабораторный практикум) по дисциплине «Технологии программирования» для студентов специальности 09.03.02 "Информационные системы и технологии" / сост. Николаев Е.И. – Ставрополь: СКФУ, 2014. – 150 с.

9. Методические указания к выполнению курсовой работы по дисциплине «Технологии программирования» для студентов специальности 09.03.02 "Информационные системы и технологии" / сост. Николаев Е.И. – Ставрополь: СКФУ, 2016. – 25 с.

# ПРИЛОЖЕНИЕ 1. Диаграмма классов приложения



# ПРИЛОЖЕНИЕ 2. Исходный код файла Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using CourseWork;

namespace CourseWork

{

public partial class GameForm : Form

{

public int GaugeWidth = 45;

public int GaugeHeight = 45;

public int Columns;

public int Rows;

public int GreenButtons = 0;

public int GameFieldStartX = 25;

public int GameFieldStartY = 80;

public bool[,] GaugeStates;

public bool GameActive = false;

public Image redButtonTexture = Bitmap.FromFile("GaugeRed.png");

public Image greenButtonTexture = Bitmap.FromFile("GaugeGreen.png");

public Graphics GFX;

public GameForm()

{

InitializeComponent();

GFX = this.CreateGraphics();

}

private void NewGame\_Click(object sender, EventArgs e)

{

if (columnsBox.Text != "" && rowsBox.Text != "")

{

Columns = Int32.Parse(columnsBox.Text);

Rows = Int32.Parse(rowsBox.Text);

GaugeStates = new bool[Columns,Rows];

if (Columns \* GaugeWidth + 75 < 700)

Program.gameForm.Size = new System.Drawing.Size(700, Rows \* GaugeHeight + 130);

else

Program.gameForm.Size = new System.Drawing.Size(Columns \* GaugeWidth + 75, Rows \* GaugeHeight + 130);

GFX.Clear(Color.White);

if (textureBox.Text == "Обычный")

{

redButtonTexture = Bitmap.FromFile("GaugeRed.png");

greenButtonTexture = Bitmap.FromFile("GaugeGreen.png");

}

else if (textureBox.Text == "Красивый")

{

redButtonTexture = Bitmap.FromFile("ButtonRed.png");

greenButtonTexture = Bitmap.FromFile("ButtonGreen.png");

}

else if (textureBox.Text == "Упрощенный")

{

redButtonTexture = Bitmap.FromFile("MarkRed.png");

greenButtonTexture = Bitmap.FromFile("MarkGreen.png");

}

for (int i = 0; i < Columns; i++)

{

for (int j = 0; j < Rows; j++)

{

GFX.DrawImage(redButtonTexture, new Point(GameFieldStartX + GaugeWidth \* i, GameFieldStartY + GaugeHeight \* j));

GaugeStates[i,j] = false;

}

}

GameActive = true;

}

}

private void GameForm\_MouseClick(object sender, MouseEventArgs e)

{

if (GameActive)

{

if (e.X > GameFieldStartX && e.X <= GameFieldStartX + Columns \* GaugeWidth && e.Y > GameFieldStartY && e.Y <= GameFieldStartY + Rows \* GaugeHeight)

{

for (int i = 0; i < Columns; i++)

{

for (int j = 0; j < Rows; j++)

{

if (e.X > GameFieldStartX + GaugeWidth \* i && e.X <= GameFieldStartX + GaugeWidth \* (i + 1))

{

if (GaugeStates[i, j])

GaugeStates[i, j] = false;

else

GaugeStates[i, j] = true;

}

else if (e.Y > GameFieldStartY + GaugeHeight \* j && e.Y <= GameFieldStartY + GaugeHeight \* (j + 1))

{

if (GaugeStates[i, j])

GaugeStates[i, j] = false;

else

GaugeStates[i, j] = true;

}

if (GaugeStates[i, j])

GFX.DrawImage(greenButtonTexture, new Point(GameFieldStartX + GaugeWidth \* i, GameFieldStartY + GaugeHeight \* j));

else

GFX.DrawImage(redButtonTexture, new Point(GameFieldStartX + GaugeWidth \* i, GameFieldStartY + GaugeHeight \* j));

}

}

for (int i = 0; i < Columns; i++)

{

for (int j = 0; j < Rows; j++)

{

if (GaugeStates[i, j])

GreenButtons++;

else

{

GreenButtons = 0;

break;

}

}

}

if (GreenButtons == Columns \* Rows)

{

GFX.Clear(Color.White);

Program.gameForm.Size = new System.Drawing.Size(900, 700);

GFX.DrawImage(Bitmap.FromFile("Win.png"), new Point(GameFieldStartX, GameFieldStartY));

GameActive = false;

}

}

}

}

}

}

# ПРИЛОЖЕНИЕ 3. Исходный код файла Form1.Designer.cs

namespace CourseWork

{

partial class GameForm

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.NewGame = new System.Windows.Forms.Button();

this.rowsBox = new System.Windows.Forms.ComboBox();

this.columnsBox = new System.Windows.Forms.ComboBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.textureBox = new System.Windows.Forms.ComboBox();

this.label3 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// NewGame

//

this.NewGame.Location = new System.Drawing.Point(13, 13);

this.NewGame.Name = "NewGame";

this.NewGame.Size = new System.Drawing.Size(155, 57);

this.NewGame.TabIndex = 0;

this.NewGame.Text = "Новая Игра";

this.NewGame.UseVisualStyleBackColor = true;

this.NewGame.Click += new System.EventHandler(this.NewGame\_Click);

//

// rowsBox

//

this.rowsBox.FormattingEnabled = true;

this.rowsBox.Items.AddRange(new object[] {

"5",

"7",

"10",

"13",

"15",

"17",

"20"});

this.rowsBox.Location = new System.Drawing.Point(275, 13);

this.rowsBox.Name = "rowsBox";

this.rowsBox.Size = new System.Drawing.Size(127, 21);

this.rowsBox.TabIndex = 1;

//

// columnsBox

//

this.columnsBox.FormattingEnabled = true;

this.columnsBox.Items.AddRange(new object[] {

"5",

"7",

"10",

"13",

"15",

"17",

"20"});

this.columnsBox.Location = new System.Drawing.Point(275, 48);

this.columnsBox.Name = "columnsBox";

this.columnsBox.Size = new System.Drawing.Size(127, 21);

this.columnsBox.TabIndex = 2;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(216, 16);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(34, 13);

this.label1.TabIndex = 3;

this.label1.Text = "Ряды";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(216, 51);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(51, 13);

this.label2.TabIndex = 4;

this.label2.Text = "Столбцы";

//

// textureBox

//

this.textureBox.FormattingEnabled = true;

this.textureBox.Items.AddRange(new object[] {

"Обычный",

"Красивый",

"Упрощенный"});

this.textureBox.Location = new System.Drawing.Point(504, 13);

this.textureBox.Name = "textureBox";

this.textureBox.Size = new System.Drawing.Size(121, 21);

this.textureBox.TabIndex = 5;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(447, 16);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(37, 13);

this.label3.TabIndex = 6;

this.label3.Text = "Стиль";

//

// GameForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.SystemColors.ControlLightLight;

this.ClientSize = new System.Drawing.Size(1904, 1041);

this.Controls.Add(this.label3);

this.Controls.Add(this.textureBox);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.columnsBox);

this.Controls.Add(this.rowsBox);

this.Controls.Add(this.NewGame);

this.Name = "GameForm";

this.Text = "Головоломка";

this.MouseClick += new System.Windows.Forms.MouseEventHandler(this.GameForm\_MouseClick);

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button NewGame;

private System.Windows.Forms.ComboBox rowsBox;

private System.Windows.Forms.ComboBox columnsBox;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.ComboBox textureBox;

private System.Windows.Forms.Label label3;

}

}

# ПРИЛОЖЕНИЕ 4. Исходный код файла Program.cs.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace CourseWork

{

public static class Program

{

public static GameForm gameForm = new GameForm();

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.Run(gameForm);

}

}

}