

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий

Кафедра Информатики и систем управления

Лабораторная работа №7 «Построение пакета документации.»

ОТЧЕТ по лабораторной работе № 7

по дисциплине

Технологии программирования

РУКОВОДИТЕЛЬ:

(подпись)

Капранов С.Н.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Куликова Е.А.
(фамилия, и.,о.)

18-ИСТ-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород

2020

Содержание

Введение.....	2
1. Цель работы	3
2. Задачи	3
3. Документация лабораторной работы №6	3
3.1 Инструкция пользователя	3
3.2 Инструкция программиста	5
3.3 Инструкция системного администратора	11
4. Документация курсовой работы.....	12
4.1 Инструкция пользователя.....	12
4.2 Инструкция программиста	14
4.3 Инструкция системного администратора	17
Заключение	18
Используемая литература.....	19

					ЛР7 – НГТУ – 18-ИСТ-4 – 908 – 10								
Изм	Лист	№ Докум.	Подпись	Дата	Лабораторная работа №7				Лит.	Лист	Листов		
Разраб.	Куликова Е.А.										1	19	
Проверил	Капранов С.Н.								Каф. ИСУ 18-ИСТ-4				
Н. контр.													
Утв.													

Введение

Документация – это совокупность данных и документов.

Документирование связано с делопроизводством (документационным обеспечением управления), отраслью деятельности, обеспечивающей документирование и организацию работы с официальными документами.

Руководство пользователя – документ, назначение которого – предоставить людям помощь в использовании некоторой системы. Документ входит в состав технической документации на систему и, как правило, подготавливается техническим писателем.

Большинство руководств пользователя помимо текстовых описаний содержат изображения. В случае программного обеспечения, в руководство обычно включаются снимки экрана, при описании аппаратуры – простые и понятные рисунки. Используется стиль и язык, доступный предполагаемой аудитории, использование жаргона сокращается до минимума либо подробно объясняется.

Руководство системного администратора – вспомогательный документ для прикладных программных продуктов и основной для серверных и системных, не имеющих непосредственных пользователей.

В седьмой лабораторной работе необходимо составить документацию, состоящую из инструкции пользователя, инструкции программиста и инструкции системного администратора для курсовой работы (игровая программа) и для лабораторной работы № 6.

1. Цель работы

Создать документацию, состоящую из инструкции пользователя, инструкции программиста и инструкции системного администратора для курсовой работы (игровая программа) и для лабораторной работы № 6.

2. Задачи

Поставленные задачи:

1. Проанализировать повторно сделанные работы;
2. Разработать документацию для лабораторной работы №6;
3. Разработать документацию для курсовой работы.

3. Документация лабораторной работы №6

3.1 Инструкция пользователя

Программа с вводом, сохранением, загрузкой и отображением данных, представленных в виде графа.

Рисунок 1 – Основной интерфейс

Чтобы начать построение графа предварительно необходимо выбрать тип желаемого графа – это можно сделать в выпадающем списке, расположенном в верхнем правом углу интерфейса, где по умолчанию будет выбран тип «Ориентированный граф». Результатом данных действий будет выбор типа будущего графа.

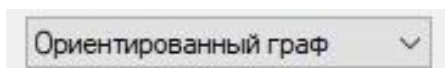


Рисунок 2 – Выпадающий список для выбора типа графа

После того как был выбран тип строящегося графа можно приступить непосредственно к самому построению, для этого по желанию необходимо ввести текст в первую графу заполнения для события (любой текст или описание на усмотрение пользователя), после чего необходимо нажать на кнопку «Добавить» и указать место расположения корневого узла на рабочем поле, расположенного на большей части всего приложения в правом нижнем углу. В результате данных действий будет построена первая вершина графа.

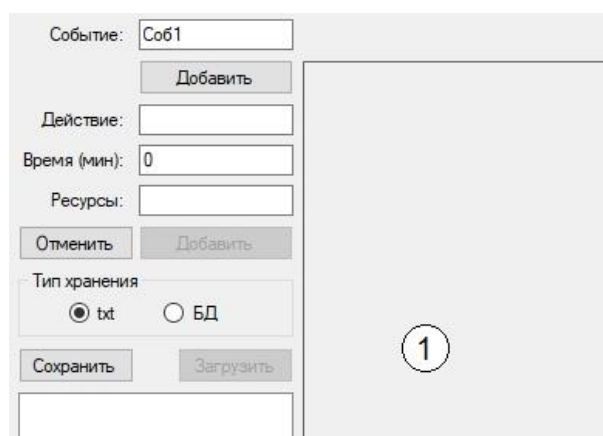


Рисунок 3 – Построение первой вершины

Чтобы начать построение рёбер необходимо иметь на рабочем поле как минимум две вершины, после чего по желанию пользователя можно ввести описание создаваемого объекта в виде трёх пунктов: Действие, Время (мин), Ресурсы. Далее необходимо нажать на кнопку «Добавить» и указать две вершины на рабочем поле в порядке направления стрелки (стрелка при построении ориентированного графа и дерева, в неориентированном графе

прямые). Если необходимо создать фиктивную работу (пунктирное ребро), то поле под указание ресурсов должно оставаться незаполненным.

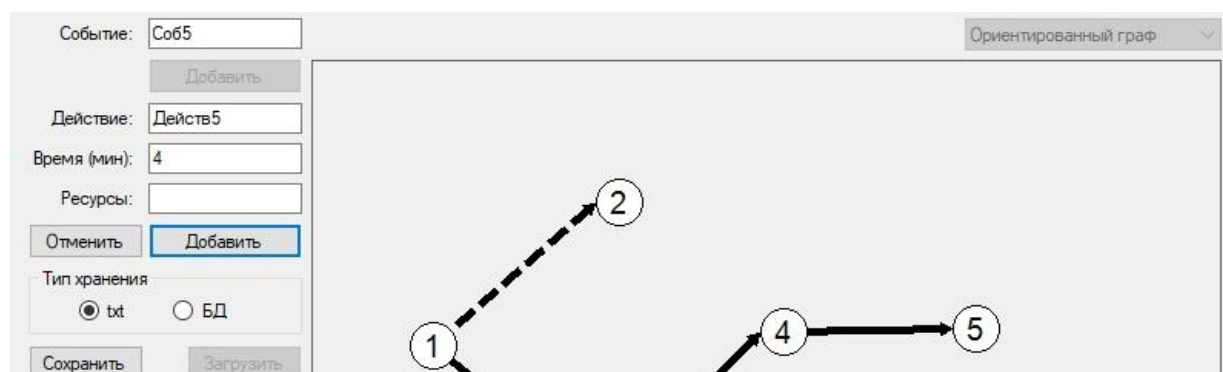


Рисунок 4 – Построение рёбер в ориентированном графе

Если необходимо просмотреть информацию по созданным объектам, то вначале нужно убедиться, что вы не находитесь в режиме добавления вершины/ребра, на всякий случай можно нажать на кнопку «Отменить», которая отменит добавление вершины/ребра, далее необходимо щёлкнуть на желаемый объект, информацию которого вы хотите узнать.

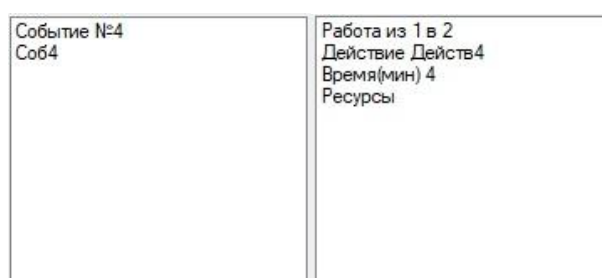


Рисунок 5 – Информационное окно с информацией вершины/ребра

Если вы хотите не потерять построенный вами граф, то вы можете его сохранить, для этого вам необходимо выбрать формат сохранения, путём нажатия на соответствующий кружок рядом с текстом «txt» или «БД». Далее необходимо нажать на кнопку «Сохранить».

Если вы хотите загрузить ранее созданный граф, то необходимо выбрать формат, в котором была сохранена работа, и нажать на кнопку «Загрузить».

3.2 Инструкция программиста

Используемые классы, структуры и функции:

SqlConnection – Представляет подключение к базе данных SQL Server.

Этот класс не может быть унаследован.

- SqlConnection.Open – Открывает подключение к базе данных на основе параметров, заданных в строке соединения connectionString.

Dictionary<TKey,TValue> – Представляет коллекцию ключей и значений.

- Dictionary<TKey,TValue>.Add(TKey, TValue) – Добавляет указанные ключ и значение в словарь.
- Dictionary<TKey,TValue>.Values – Возвращает коллекцию, содержащую значения из словаря Dictionary<TKey,TValue>.
- Dictionary<TKey,TValue>.Remove – Удаляет значение с указанным ключом из словаря Dictionary<TKey,TValue>.

KeyValuePair<TKey,TValue> – Определяет пару ключ / значение, которую можно установить или получить.

- KeyValuePair<TKey,TValue>.Key – Получает ключ в паре ключ / значение.
- KeyValuePair<TKey,TValue>.Value – Получает значение в паре ключ / значение.

List<T> – Представляет строго типизированный список объектов, доступных по индексу. Поддерживает методы для поиска по списку, выполнения сортировки и других операций со списками.

- List<T>.Add(T) – Добавляет объект в конец очереди List<T>.
- List<T>.Remove(T) – Удаляет первое вхождение указанного объекта из коллекции List<T>.
- List<T>.Count – Получает число элементов, содержащихся в интерфейсе List<T>.

ComboBox – Представляет элемент управления со списком Windows.

- ComboBox.SelectedIndex – Получает или задает индекс, указывающий текущий выбранный элемент.

Button – Представляет элемент управления "кнопка Windows".

Brushes – Кисти для каждого из стандартных цветов. Этот класс не может быть унаследован.

Graphics – Инкапсулирует поверхность рисования GDI+. Этот класс не наследуется.

- Graphics.DrawArc – Рисует дугу, которая является частью эллипса, заданного парой координат, шириной и высотой.
- Graphics.DrawLine – Проводит линию, соединяющую две точки, задаваемые парами координат.
- Graphics.FillEllipse – Заполняет внутреннюю часть эллипса, определяемого ограничивающим прямоугольником, заданным с помощью пары координат, ширины и высоты.
- Graphics.DrawEllipse – Рисует эллипс, определяемый ограничивающим прямоугольником, заданным с помощью пары координат, ширины и высоты.
- Graphics.DrawString – Создает заданную текстовую строку в указанном месте с помощью заданных объектов Brush и Font.

Pen – Определяет объект, используемый для рисования прямых линий и кривых. Этот класс не может быть унаследован.

- Pen.EndCap – Получает или задает стиль окончания линий, нарисованных при помощи объекта Pen.
- Pen.DashStyle – Получает или задает стиль, используемый для пунктирных линий, нарисованных при помощи объекта Pen.

Pens – Перья для всех стандартных цветов. Этот класс не наследуется.

Font – Определяет конкретный формат текста, включая атрибуты шрифта, размера и стиля. Этот класс не может быть унаследован.

Point – Представляет упорядоченную пару целочисленных x- и y-координат, которая определяет точку в двумерной плоскости.

Char – Представляет символ как кодовую единицу UTF-16.

- Char.IsDigit – Указывает, относится ли символ Unicode к десятичной цифре.

KeyPressEventArgs – Предоставляет данные для события KeyPress.

- KeyPressEventArgs.KeyChar – Возвращает или задает символ, соответствующий нажатой клавише.
- KeyPressEventArgs.Handled – Возвращает или задает значение, показывающее, было ли обработано событие KeyPress.

MouseEventArgs – Предоставляет данные для событий MouseUp, MouseDown и MouseMove.

- MouseEventArgs.X – Возвращает координату X указателя мыши в момент создания события мыши.
- MouseEventArgs.Y – Возвращает координату Y указателя мыши в момент создания события мыши.

Label – Представляет стандартную метку Windows.

- Label.Text – Возвращает или задает текст, связанный с этим элементом управления.

TextBox – Представляет текстовое поле, элемент управления Windows.

- TextBox.Text – Возвращает или задает текст, связанный с этим элементом управления.

ListBox – Представляет элемент управления Windows для отображения списка элементов.

- ListBox.Items – Получает позиции элемента управления ListBox.
- ListBox.ObjectCollection.Clear – Удаляет все предметы из коллекции.
- ListBox.ObjectCollection.Add(Object) – Добавляет элемент в список элементов для ListBox.

RadioButton – Позволяет пользователю выбрать единственный вариант из группы доступных, когда используется вместе с другими элементами управления RadioButton.

- `RadioButton.Checked` – Возвращает или задает значение, указывающее, выбран ли данный элемент управления.

`Enumerable` – Предоставляет набор методов типа `static` (Shared в Visual Basic) для выполнения запросов к объектам, реализующим интерфейс `IEnumerable<T>`.

- `Enumerable.FirstOrDefault` – Возвращает первый элемент последовательности или значение по умолчанию, если ни одного элемента не найдено.
- `Enumerable.Last` – Возвращает последний элемент последовательности.

`Encoding` – Представляет кодировку символов.

- `Encoding.Default` – Возвращает кодировку по умолчанию для данной реализации .NET.

`StreamWriter` – Реализует `TextWriter` для записи символов в поток в определенной кодировке.

`TextWriter` – Представляет писателя, который может написать последовательную серию символов. Этот класс является абстрактным.

- `TextWriter.WriteLine` – Записывает данные в текстовый поток, после которого следует разделитель строки.

`SqlCommand` – Представляет инструкцию Transact-SQL или хранимую процедуру, выполняемую над базой данных SQL Server. Этот класс не может быть унаследован.

- `SqlCommand.ExecuteNonQuery` – Выполняет для подключения инструкцию Transact-SQL и возвращает количество задействованных в инструкции строк.

`StreamReader` – Реализует `TextReader`, который читает символы из байтового потока в определенной кодировке.

- `StreamReader.ReadLine` – Читает строку символов из текущего потока и возвращает данные в виде строки.

Convert – Преобразует значение одного базового типа данных в другой базовый тип данных.

- Convert.ToInt32 – Преобразует заданное значение в 32-битовое целое число без знака.
- Convert.ToInt32 – Преобразует заданное значение в 32-битовое целое число со знаком.
- Convert.ToBoolean – Преобразует заданное значение в эквивалентное логическое значение.

SqlDataAdapter – Представляет набор команд данных и подключение к базе данных, которые используются для заполнения DataSet и обновления базы данных SQL Server. Этот класс не наследуется.

DbDataAdapter – Помогает реализовать интерфейс IDbDataAdapter. Наследники DbDataAdapter реализуют набор функций для обеспечения строгой типизации, но наследуют большинство функций, необходимых для полной реализации DataAdapter.

- DbDataAdapter.Fill – Заполняет DataSet или DataTable.

DataSet – Представляет кэш в памяти для данных.

- DataSet.Tables – Возвращает коллекцию таблиц класса DataSet.

DataTable – Представляет одну таблицу данных в памяти.

- DataTable.Rows – Получает коллекцию строк, которые принадлежат этой таблице.

DataRowCollection – Представляет коллекцию строк для DataTable.

- DataRowCollection.Count – Получает общее количество объектов DataRow в этой коллекции.

Math – Предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

- Math.Pow(Double, Double) – Возвращает указанное число, возведенное в указанную степень.

- Math.Sqrt(Double) – Возвращает квадратный корень из указанного числа.

Control – Определяет базовый класс для элементов управления, являющихся компонентами с визуальным представлением.

- Control.Enabled – Возвращает или задает значение, указывающее, может ли элемент управления отвечать на действия пользователя.
- Control.CreateGraphics – Создает объект Graphics для элемента управления.

PictureBox – Представляет элемент управления Windows Picture Box для отображения изображения.

3.3 Инструкция системного администратора

Область применения системы – данная программа предназначена для построения графов (ориентированных, неориентированных или дерева), она может быть применена в разных областях, при помощи данной программы можно отобразить, к примеру, электрические сети, дорожные развязки внутри города и между населенными пунктами, схемы отображения метро, графы причинно–следственных связей, молекулярные структуры веществ, сетевые графики производства работ.

Комплект поставки – exe файл и база данных.

Требования к техническим средствам определяются установленной операционной системой. Программа не является ресурсоемкой и не предъявляет никаких особых требований к техническим средствам.

Минимальные системные требования:

- Операционная система: Windows 2000/XP/Vista/7/8/10 (32 или 64 bit).
- Процессор Pentium75 MHz.
- Оперативная память 16 МБ и выше.
- Жесткий диск от 90 МБ.

- Дисплей.
- Мышь, клавиатура.

Программное обеспечение должно функционировать на операционных системах семейства MS Windows 2000/XP/Vista/7/10. Функционирование программы под операционной системой MS Windows 98/Me также возможно, но тестироваться в рамках данного проекта не будет.

Программа не предъявляет особых требований к интерфейсу пользователя.

Порядок установки программный компонентов – сначала необходимо установить базу данных, затем exe файл.

4. Документация курсовой работы

4.1 Инструкция пользователя

Правила игры – двое по очереди (человек и компьютер) "откусывают" от прямоугольной доски. Игрок выбирает любое поле доски и снимает все фишки, которые находятся не выше и не левее избранного поля. Снявший последнюю фишку – проигрывает.

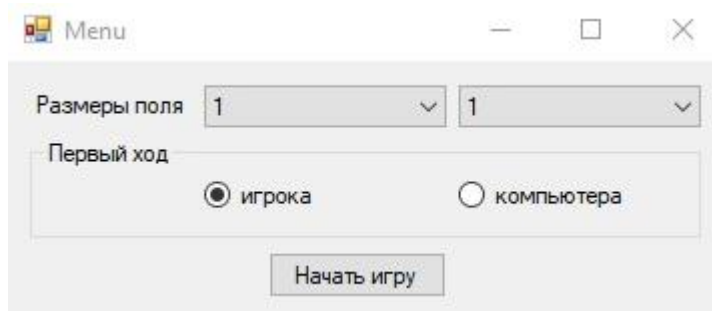


Рисунок 6 – Начальное меню

При запуске игры следует поочерёдно открыть рядом с фразой «Размеры поля» выпадающие списки, путём нажатия мыши на прямоугольную область с цифрой и стрелкой вниз. При появлении списка необходимо выбрать пункт от одного до пяти с соответствующими значениями. В результате данных действий будет сгенерировано игровое поле.

Рисунок 7 – Выбор размеров поля

В разделе «Первый ход» необходимо выбрать из двух вариантов – «игрока» или «компьютера» один, путём нажатия мыши на соответствующий кружок, расположенный рядом с пунктами. В результате данных действий будет осуществляться право на первый ход в игре.

Рисунок 8 – Выбор первенства хода

Для начала игры необходимо нажать на кнопку «Начать игру». В результате данного действия закроется окно с изначальным меню и появится игровое поле.

Рисунок 9 – Кнопка «Начать игру»

В зависимости от выбранных размеров на поле будут расположены белые (недоступные для взаимодействия) и зелёные квадраты, а также один красный. При доступном ходе игрока, следует нажимать на зелёные и красный квадраты (на красный квадрат рекомендуется нажимать в крайних безвыходных положениях, так как данное действие ведёт к проигрышу). Поле при нажатии на зелёные квадраты будет отсекается от игрового пространства по правилам игры (всё правее и ниже выбранного зелёного квадрата включительно будут становиться белыми и переставать быть доступными для взаимодействия) до тех пор, пока не останется красный квадрат. В результате данных действий будет реализована игровая партия человека и компьютера.

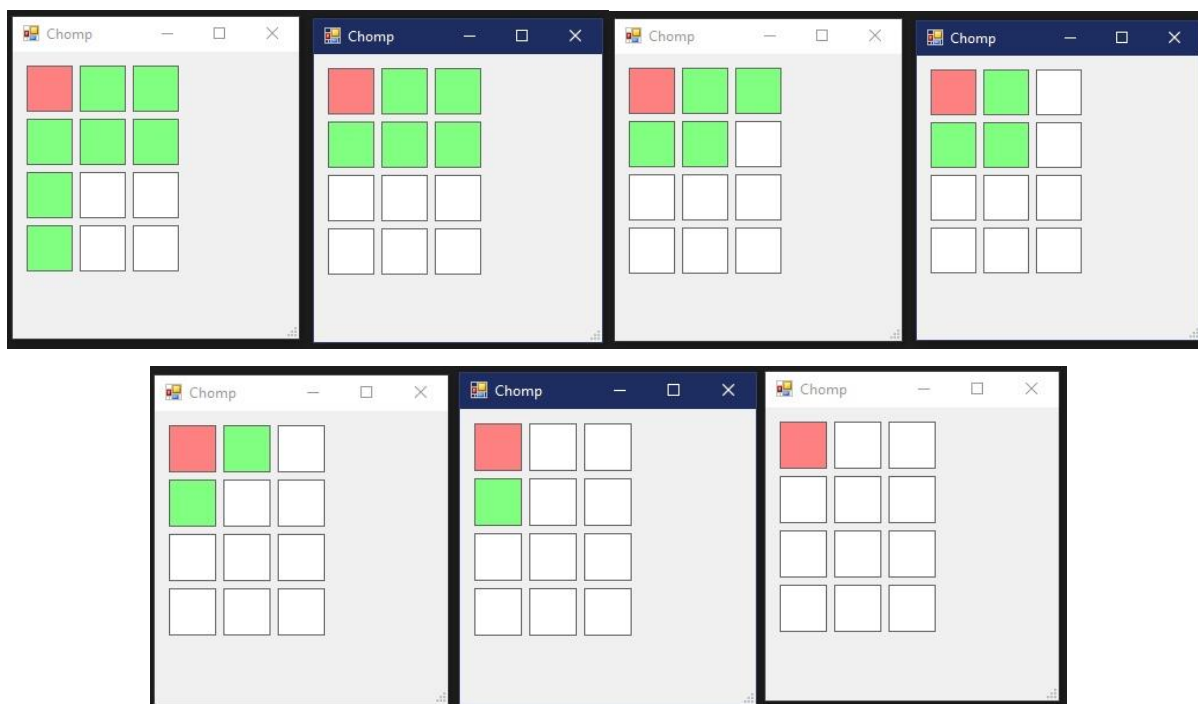


Рисунок 10 – Пример сыгранной партии с первым ходом компьютера

После нажатия игрока на красный квадрат появляется окно с результатом сыгранной игры для человека. После ознакомления с результатами необходимо нажать на кнопку «Ок». В результате данных действий игрок-человек узнаёт свой результат в игре и полностью закрывает приложение.

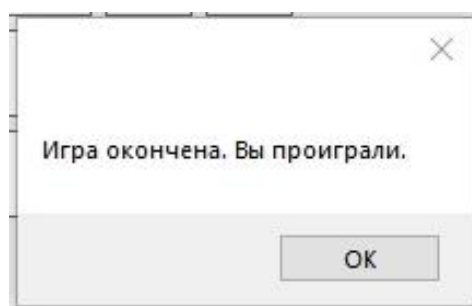


Рисунок 11 – Пример результата игры

4.2 Инструкция программиста

Используемые классы, структуры и функции:

`List<T>` – Представляет строго типизированный список объектов, доступных по индексу. Поддерживает методы для поиска по списку, выполнения сортировки и других операций со списками.

- `List<T>.Add(T)` – Добавляет объект в конец очереди `List<T>`.

Control – Определяет базовый класс для элементов управления, являющихся компонентами с визуальным представлением.

- `Control.BackColor` – Возвращает или задает цвет фона для элемента управления.
- `Control.Enabled` – Возвращает или задает значение, указывающее, может ли элемент управления отвечать на действия пользователя.
- `Control.TabIndex` – Возвращает или задает последовательность перехода по клавише TAB между элементами управления внутри контейнера.
- `Control.Location` – Возвращает или задает координаты левого верхнего угла элемента управления относительно левого верхнего угла его контейнера.
- `Control.Size` – Возвращает или задает высоту и ширину элемента управления.
- `Control.TabStop` – Возвращает или задает значение, указывающее, может ли пользователь перевести фокус на данный элемент управления при помощи клавиши TAB.
- `Control.Click` – Происходит при щелчке элемента управления.
- `Control.Controls` – Возвращает коллекцию элементов управления, содержащихся в элементе управления.
- `Control.Hide` – Скрывает элемент управления от пользователя.

Button – Представляет элемент управления "кнопка Windows".

MessageBox – Отображает окно сообщения.

- `MessageBox.Show` – Отображает окно сообщения.

Form – Представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения.

- `Form.ClientSize` – Возвращает или задает размер клиентской области формы.

- Form.Close – Закрывает форму.
- Form.Load – Происходит до первоначального отображения формы.
- Form.ShowDialog – Отображает форму как модальное диалоговое окно.

EventHandler – Представляет метод, обрабатывающий событие, не имеющее данных.

Thread – Создает и контролирует поток, устанавливает его приоритет и получает его статус.

- Thread.Sleep – Приостанавливает текущий поток на указанное время.

Color – Представляет цвета в терминах каналов альфа, красного, зеленого и синего (ARGB).

- Color.FromArgb – Создает структуру Color из указанных 8-разрядных значений компонентов ARGB (альфа, красный, зеленый и синий).

Size – Сохраняет упорядоченную пару целых чисел, указывающих Height и Width.

Int32 – Представляет 32-разрядное целое число со знаком.

- Int32.MinValue – Представляет наименьшее возможное значение Int32. Это поле является постоянным.
- Int32.MaxValue – Представляет максимально возможное значение Int32. Это поле является постоянным.

ComboBox – Представляет элемент управления со списком Windows.

- ComboBox.SelectedIndex – Получает или задает индекс, указывающий текущий выбранный элемент.

RadioButton – Позволяет пользователю выбрать единственный вариант из группы доступных, когда используется вместе с другими элементами управления RadioButton.

- RadioButton.Checked – Возвращает или задает значение, указывающее, выбран ли данный элемент управления.

Требования к техническим средствам определяются установленной операционной системой. Программа не является ресурсоемкой и не предъявляет никаких особых требований к техническим средствам.

Минимальные системные требования:

- Операционная система: Windows 2000/XP/Vista/7/8/10 (32 или 64 bit).
- Процессор Pentium75 MHz.
- Оперативная память 16 МБ и выше.
- Жесткий диск от 90 МБ.
- Дисплей.
- Мышь, клавиатура.

Программное обеспечение должно функционировать на операционных системах семейства MS Windows 2000/XP/Vista/7/10. Функционирование программы под операционной системой MS Windows 98/Me также возможно, но тестироваться в рамках данного проекта не будет.

Программа не предъявляет особых требований к интерфейсу пользователя.

4.3 Инструкция системного администратора

Программа предназначена для игры, основанной на определённых математических алгоритмах, которая помогает человеку в лёгкой форме избавиться от потребности в обучении. Любая игра – есть обучение. Когда человек начинает играть в новую игру, он неловок, невнимателен и делает много ошибок, отчего человек, который по своей натуре не любит проигрывать, стремиться улучшать свои навыки, в результате чего появляется азарт, который в безобидной форме человеку принесёт приятные эмоции.

Обязанности администратора – поставлять exe файл приложения.

Заключение

В ходе седьмой лабораторной работы была создана документация, состоящая из инструкции пользователя, инструкции программиста и инструкции системного администратора для курсовой работы (игровая программа) и для лабораторной работы № 6.

					ЛР7 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Используемая литература

1. Руководство системного администратора – <https://philosoft-services.com/sysadmins-manual.zhtml>
2. Руководство пользователя – https://ru.wikipedia.org/wiki/Руководство_пользователя
3. Документация – <https://ru.wikipedia.org/wiki/Документация>

					ЛР7 – НГТУ – 18-ИСТ-4 – 908 – 10	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		