

Учреждение образования
«Гродненский государственный политехнический колледж»

ОТЧЁТ
ПО УЧЕБНОЙ ПРАКТИКЕ ПО ПРОГРАММИРОВАНИЮ

Учащейся _____ 3 _____ курса, группы _____ ПЗТ-40
специальности 2 - 40 01 01 «Программное обеспечение информационных
технологий» _____

Место прохождения практики _____ УО «Гродненский государственный
политехнический колледж» _____

Тема проекта: «Разработка программного продукта «Экопамять» _____
Ссылка на проект: <https://github.com/KragelDarya/EcoMemory> _____

Выполнила	_____	Д.Д. Крагель (инициалы, фамилия)
Руководитель	_____	А.Г. Бабуль (инициалы, фамилия)
практики от колледжа	_____	_____

Содержание

Введение. Описание структуры предприятия.....	3
1 Анализ предметной области и формулировка требований к программе.....	4
1.1 Исследование предметной области	4
1.2 Инструменты разработки	4
2 Проектирование	6
2.1 Диаграмма вариантов использования	6
2.2 Диаграмма деятельности.....	8
3 Построение программы.....	9
4 Тестирование	10
5 Применение	11
5.1 Назначение и условия применения программы	11
5.2 Инсталляция	11
5.3 Выполнение программы	13
Заключение.....	17
Список использованных источников	18
Приложение А Листинг программы	19
Приложение Б Тест-кейсы	26

					УП КПиЯП 2-40 01 01.35.40.07.25 ПЗ			
Изм.	Лист	№докум.	Подпись	Дата				
Разраб.	Крагель				Разработка программного продукта «Экопамять»		Лит.	Лист
Пров.	Бабуль							2
								28
Н. контр.								
Утв.								

Введение. Описание структуры предприятия

На данной учебной практике была поставлена задача разработать игровое приложение «Экопамять».

Цель курсового проекта заключается в разработке программного продукта, который позволит массовому пользователю улучшить память, внимательность, концентрацию и когнитивные способности.

Создаваемое приложение будет рассчитано для любого рода пользователей. Применять его смогут как дети, так и взрослые.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ предметной области и формулировка требований к программе». В нем можно ознакомиться с постановкой задачи, которая включает в себя исследование предметной области поставленной задачи. В подразделе «Инструменты разработки» рассмотрена среда, в которой создается данный программный продукт.

В разделе «Проектирование» рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе будут представлены диаграмма вариантов использования и диаграмма деятельности.

«Построение программы» – это третий раздел отчета, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе представлена диаграмма компонентов.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы. Будут смоделированы все возможные действия пользователя при работе с приложением, начиная от запуска и заканчивая закрытием приложения.

В разделе «Применение» будет описано назначение программы, инсталляция. Будет указана последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы.

«Заключение» содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В разделе «Список использованных источников» приведен список используемых при разработке источников.

В приложении А будет приведен листинг программы.

В приложении Б будут представлены тест-кейсы.

1 Анализ предметной области и формулировка требований к программе

1.1 Исследование предметной области

Наименование задачи – «Разработка программного продукта «Экопамять».

Цель разработки: создание программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

Назначение: для развития способности к фокусировке внимания на задаче и увеличения продуктивности своей деятельности, для улучшения памяти, повышения качества внимания, улучшения памяти и когнитивных функций, а также снижения уровня стресса.

Периодичность использования данного программного продукта зависит от нужд потребителя, неограниченно

1.2 Инструменты разработки

Для разработки данного проекта была использована среда Visual Studio и язык программирования C#, это обусловлено рядом факторов:

- широкая распространенность платформы .NET среди разработчиков;
- простота освоения языка C# благодаря ясному синтаксису и большому количеству документации;
- наличие встроенных библиотек (.NET Framework), обеспечивающих удобные инструменты для взаимодействия с графикой и мультимедиа;
- высокая производительность компилятора и оптимизированная виртуальная машина CLR обеспечивают быстрое выполнение приложений.

Выбор Visual Studio для реализации проекта обусловлен её возможностями как мощной интегрированной среды разработки, которая обеспечивает удобную и эффективную работу с языком программирования C# и платформой .NET. Среда предоставляет визуальный редактор для быстрого создания пользовательского интерфейса, интуитивно понятную структуру проекта и продвинутые инструменты для написания и редактирования кода. Это особенно важно для приложений с графическим интерфейсом, таких как игры с игровым полем и элементами управления.

Кроме того, Visual Studio обладает мощной системой отладки и тестирования, позволяющей быстро выявлять и устранять ошибки. Поддержка различных форматов проекта, встроенные шаблоны, широкая база знаний и активное сообщество разработчиков делают эту среду идеальным выбором для создания учебных, развивающих и игровых программ. Всё это позволяет создать стабильный, удобный и функциональный продукт с минимальными затратами времени и усилий.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

- WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;
- Microsoft Office Word 2019 – для написания документации к программному продукту;
- Dr.Explain – инструмент разработки пользовательской документации;
- GitHub – веб-сервис для хостинга IT-проектов;
- Inkscape – инструмент для разработки иллюстраций;

					УП КПиАП 2-40 01 01.35.40.07.25 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		5

2 Проектирование

2.1 Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования.

Актером или действующим лицом является любая сущность, взаимодействующая с системой извне. Это может быть как живое существо, так и любая другая система, которая может служить источником воздействия на моделируемую систему так, как определяет сам разработчик. На рисунке 1 представлено графическое обозначение актера.



Рисунок 1 – Графическое обозначение актера

Вариант использования является стандартным языком UML и применяется для спецификаций общих особенностей системы и любой другой сущности. Отдельные варианты использования обозначаются на диаграмме эллипсом, в котором содержится его краткое название. Пример представлен на рисунке 2.

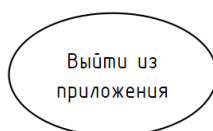


Рисунок 2 – Графическое обозначение вариантов использования

Отношение ассоциации является главным понятием языка UML и используется при построении всех графических моделей. Оно служит для обозначения роли актера и отдельном варианте использования. На диаграмме отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Пример отношения ассоциации представлен на рисунке 3.

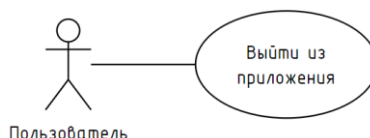


Рисунок 3 – Графическое обозначение отношения ассоциации

					УП КПиАП 2-40 01 01.35.40.07.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данная линия помечается ключевым словом <extend>. Пример изображен на рисунке 4.

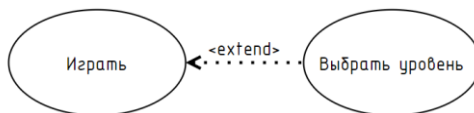


Рисунок 4 – Графическое обозначение отношения расширения

Отношение включения между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Данная линия помечается ключевым словом <include>. Пример изображен на рисунке 5.



Рисунок 5 – Графическое обозначение отношения включения

Определяя для выбранного актера варианты использования и устанавливая отношения между вариантами использования, получим полную диаграмму вариантов использования, она представлена на рисунке 6.

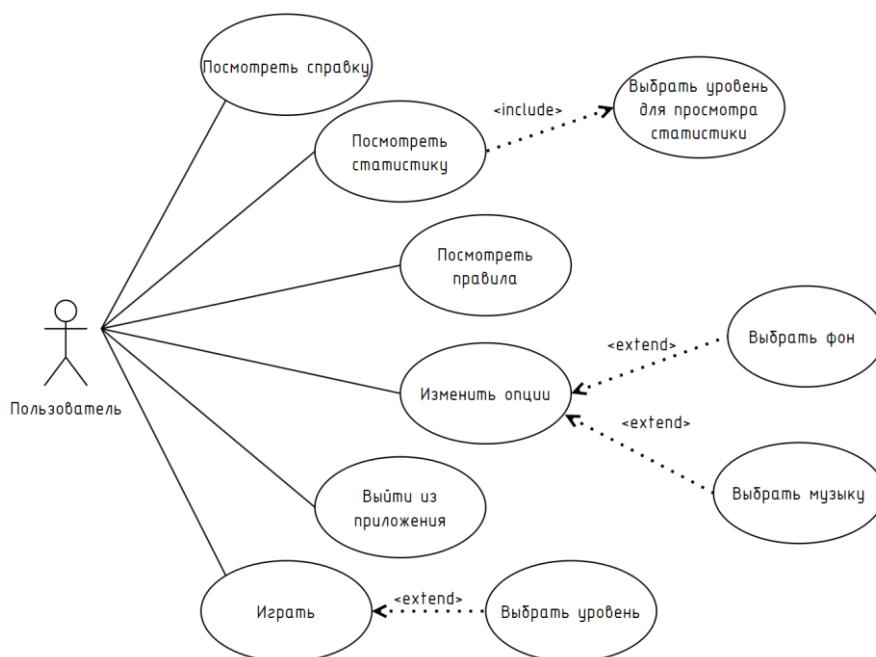


Рисунок 6 – Диаграмма вариантов использования

2.2 Диаграмма деятельности

Диаграмма деятельности – поведенческая диаграмма, показывающая поток работы или действий в рамках системы или процесса. Она иллюстрирует последовательность шагов и возможные варианты выполнения работы, включая параллельные процессы и ветвления. Диаграмма деятельности включает в себя такие элементы, как начальные и конечные узлы, узлы действий, узлы решений, вилки и слияния, а также потоки управления, которые связывают эти узлы.

На диаграмме отображен процесс прохождения тестирования.

Диаграмма деятельности представлена на рисунке 7.

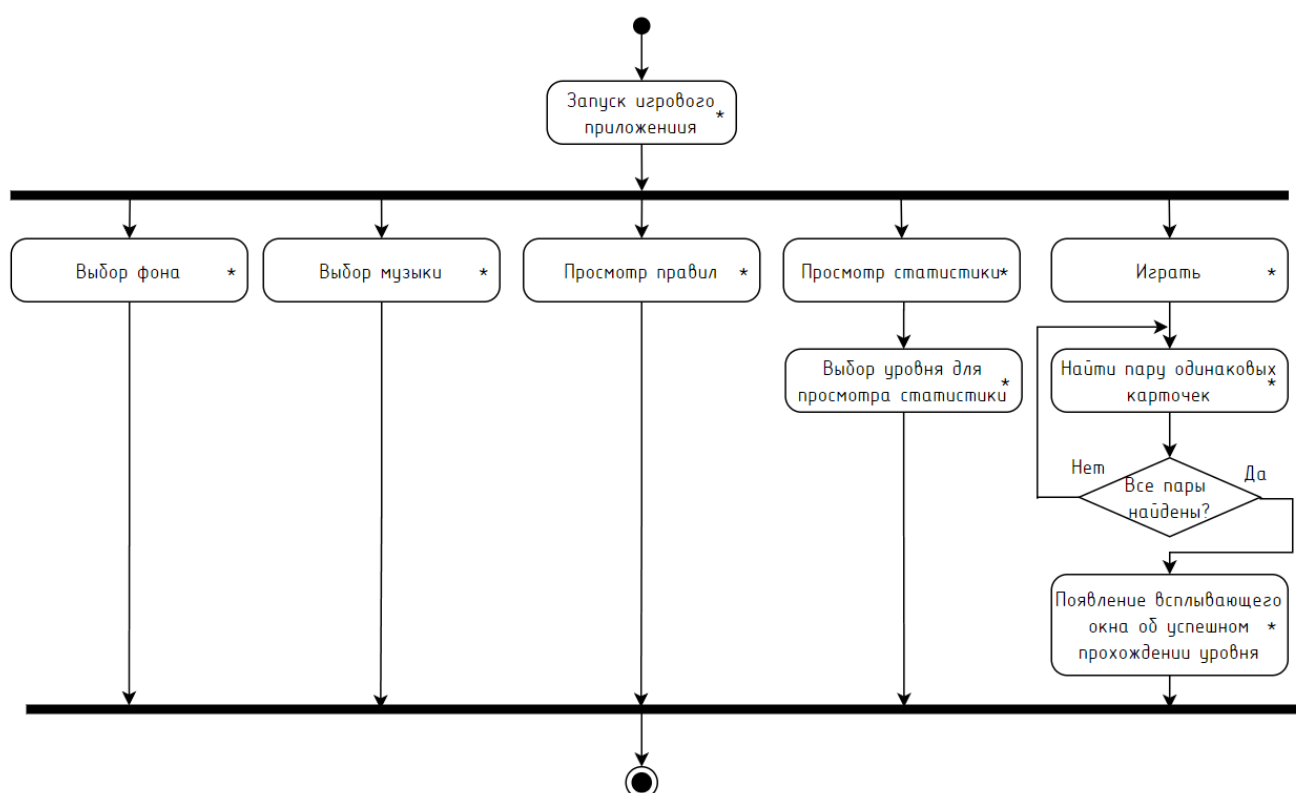


Рисунок 7 – Диаграмма деятельности

3 Построение программы

Диаграмма компонентов – статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи между компонентами.

Диаграмма компонентов представлена на рисунке 8.

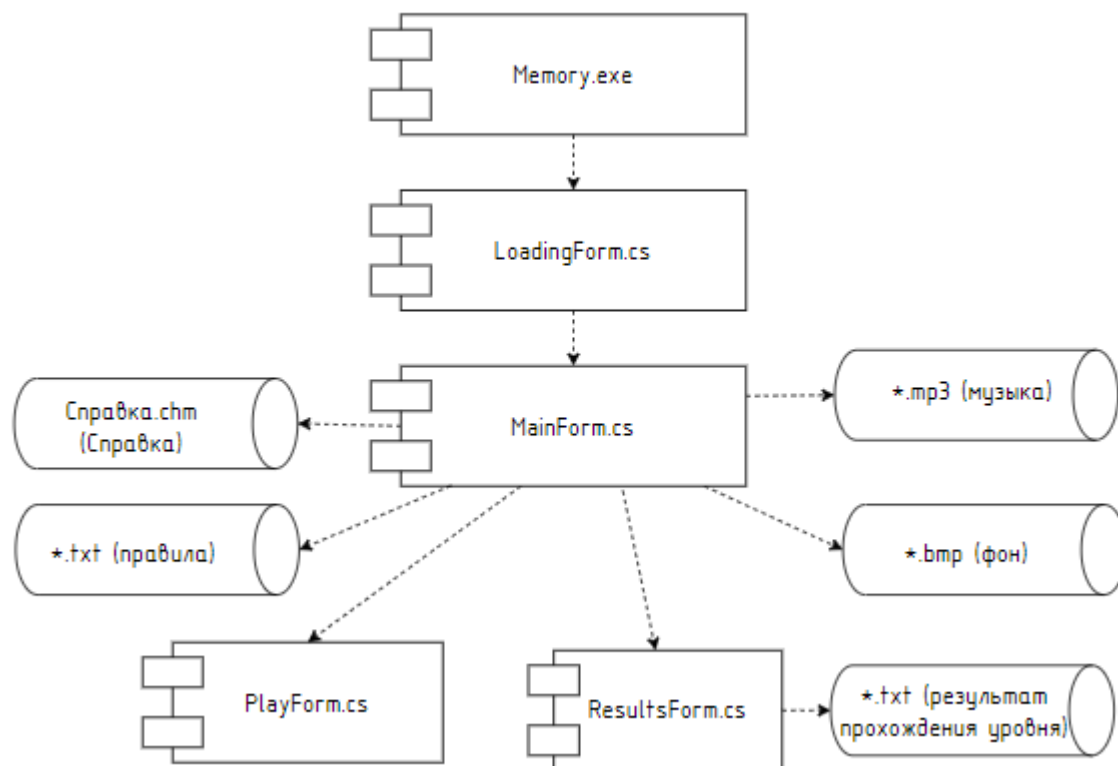


Рисунок 8 – Диаграмма компонентов

4 Тестирование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Разработанные тест-кейсы и статус их выполнения представлены в приложение Б.

Расписание проведения и время, затраченное на тестирование, описано в таблице 1.

Таблица 1 – Расписание работ над проектом

Имя	Дата	Деятельность	Продолжительность, ч
Крагель Дарья	04.05.2025	Разработка тестов	1
Крагель Дарья	05.05.2025	Тестирование познавательного приложения	2
Крагель Дарья	05.05.2025	Исправление найденных ошибок	1,5
Крагель Дарья	06.05.2025	Проведение регрессионного тестирования	1
Крагель Дарья	06.05.2025	Составление отчета о результатах тестирования	2

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

Статистика по всем дефектам представлена в таблице 2.

Таблица 2 – Статистика по всем дефектам

Статус	Количество	Важность			
		Низкая	Средняя	Высокая	Критическая
Найдено	0	0	0	0	0
Исправлено	0	0	0	0	0
Проверено	0	0	0	0	0
Открыто заново	0	0	0	0	0
Отклонено	0	0	0	0	0

5 Применение

5.1 Назначение и условия применения программы

Цель данного проекта заключается в разработке программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

Качество и скорость работы приложения всегда зависит от самих характеристик персонального компьютера. Поэтому приложение должно было быть протестировано на разных машинах. Тестирование проводилось на разных персональных компьютерах и результаты были удовлетворительные.

Сама программа была разработана на программном устройстве со следующими характеристиками:

- процессор: 12th Gen Intel(R) Core(TM) i5-12450H, 2000 МГц, ядер: 8, логических процессоров: 12
- объем оперативной памяти: 16.00 GB;
- ОС: Майкрософт Windows 10 Корпоративная LTSC

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Setup.exe. Появится мастер установки игрового приложения «Экопамять», представленный на рисунке 9.

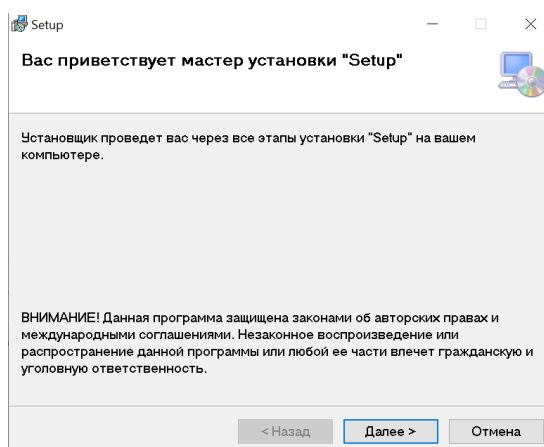


Рисунок 9 – Мастер установки игрового приложения «Экопамять»

После нажатия кнопки «Далее» появляется возможность выбора места для установки программного продукта, представленное на рисунке 10.

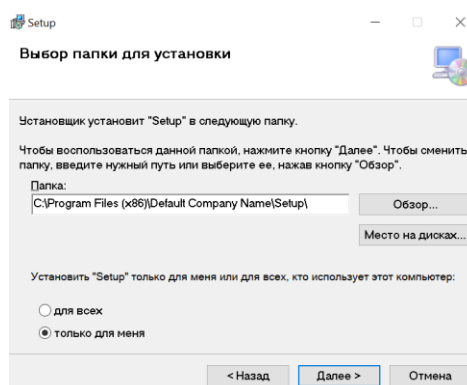


Рисунок 10 – Выбор папки установки

После нажатия на кнопку «Далее» появляется окно подтверждения установки, представленное на рисунке 11.

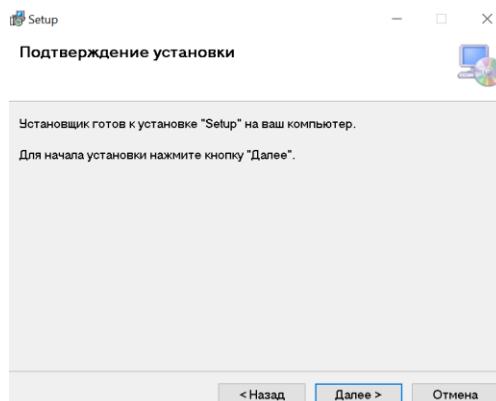


Рисунок 11 – Окно подтверждения установки

После успешной установки на рабочем столе появится окно «Установка завершена», представленное на рисунке 12 и ярлык для запуска приложения, представленный на рисунке 13.

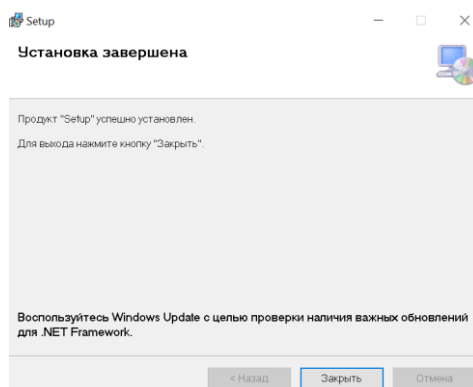


Рисунок 12 – Окно «Установка завершена»



Рисунок 13 – Ярлык приложения

5.3 Выполнение программы

После запуска приложения открывается форма «Загрузка», представленная на рисунке 14.

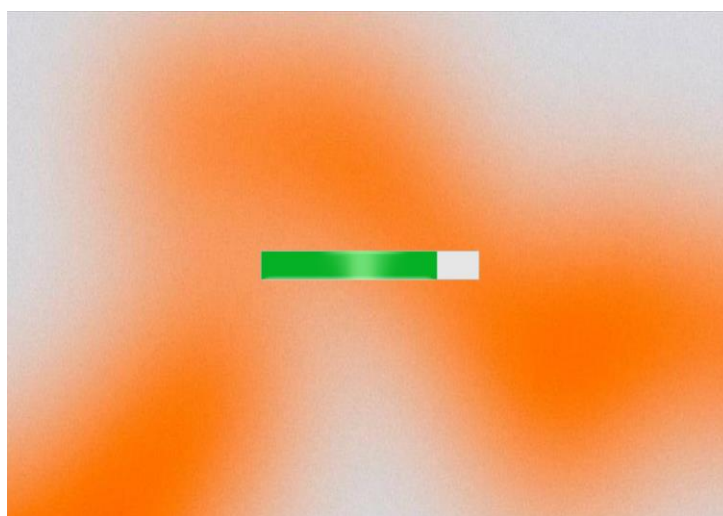


Рисунок 14 – Загрузка приложения

После загрузки открывается главное меню приложения, представленный на рисунке 15.

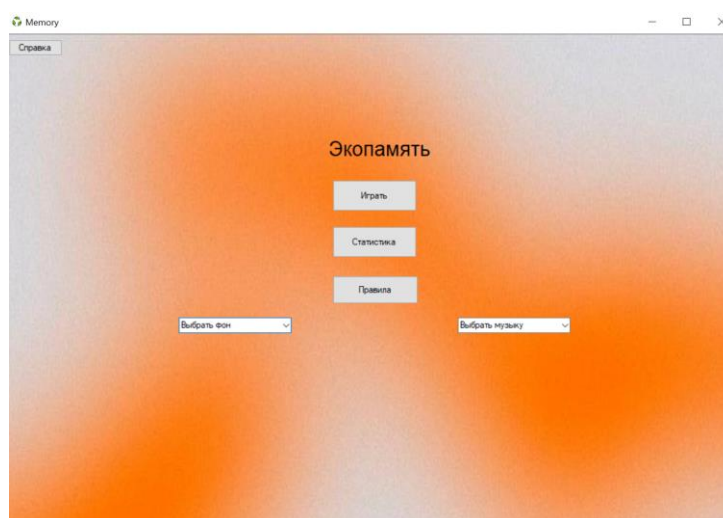


Рисунок 15 – Главное меню

У пользователя есть возможность нажать на кнопки «Играть», «Статистика», «Правила», «Справка», а также выбрать фон и музыку из сворачиваемых групп.

При нажатии на кнопку «Справка» будет открыта справка.

При нажатии на кнопку «Правила» появляется всплывающее сообщение, в котором отображаются правила игры, который расположен на рисунке 16.

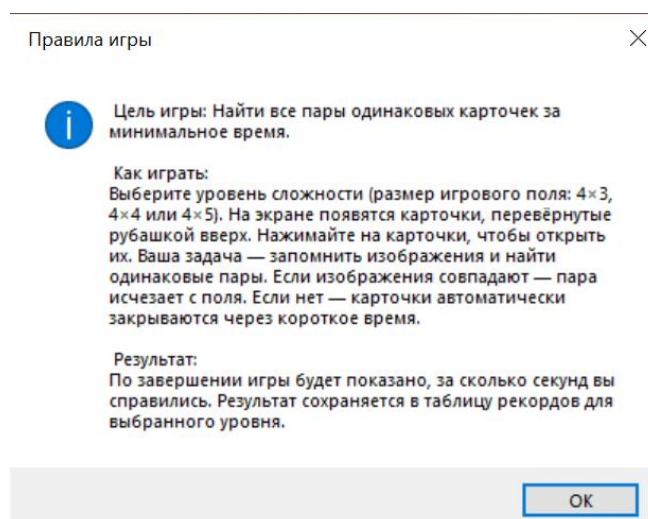


Рисунок 16 – Всплывающее сообщение «Правила»

При нажатии на кнопку «Играть» пользователь переходит на игровую форму, где представлено поле с карточками, расположенными в случайном порядке. Форма «Играть» представлена на рисунке 17. На игровой форме предусмотрены дополнительные кнопки: «На главную» — возвращает пользователя на стартовую форму, а также кнопки для выбора уровней.

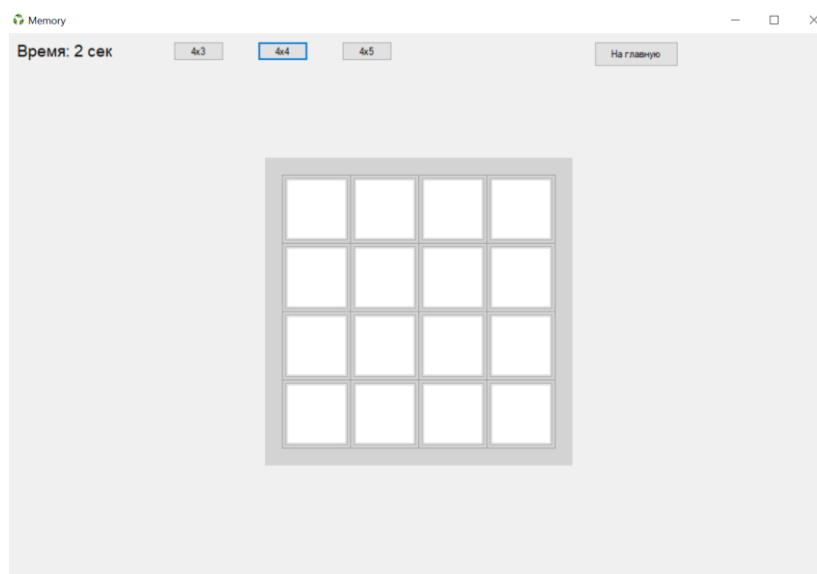


Рисунок 17 – Форма «Играть»

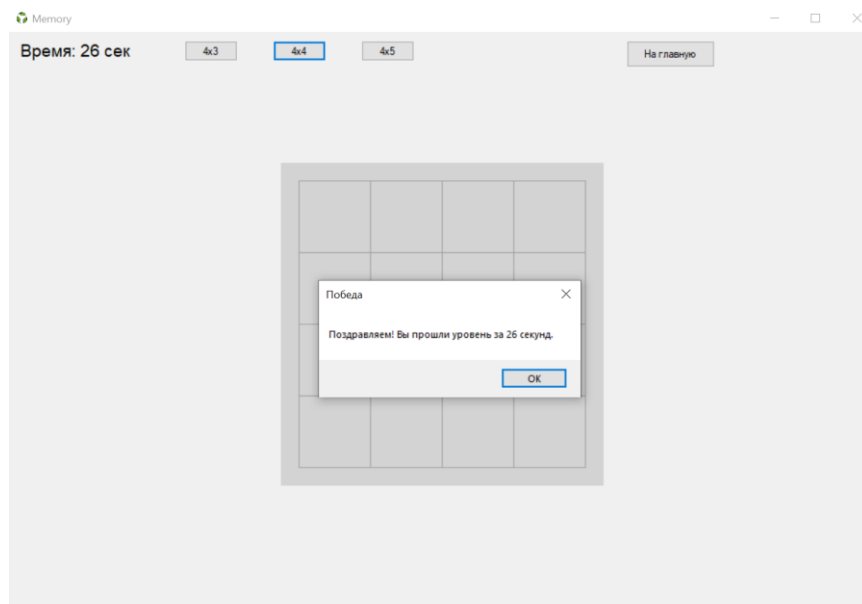


Рисунок 18 – Прохождение игры

После успешного прохождения уровня появляется всплывающее сообщение с результатом прохождения уровня, представленное на рисунке 18.

При нажатии на кнопку «На главную», представленную на рисунке 17 игрок возвращается на главное меню.

При нажатии на кнопку «Статистика» будет осуществлен переход на форму со статистикой, которая представлена на рисунке 19. При нажатии на одну из кнопок уровней будет отображаться статистика по данному уровню. Также на этой форме есть кнопка «На главную», которая осуществляет переход на главную форму

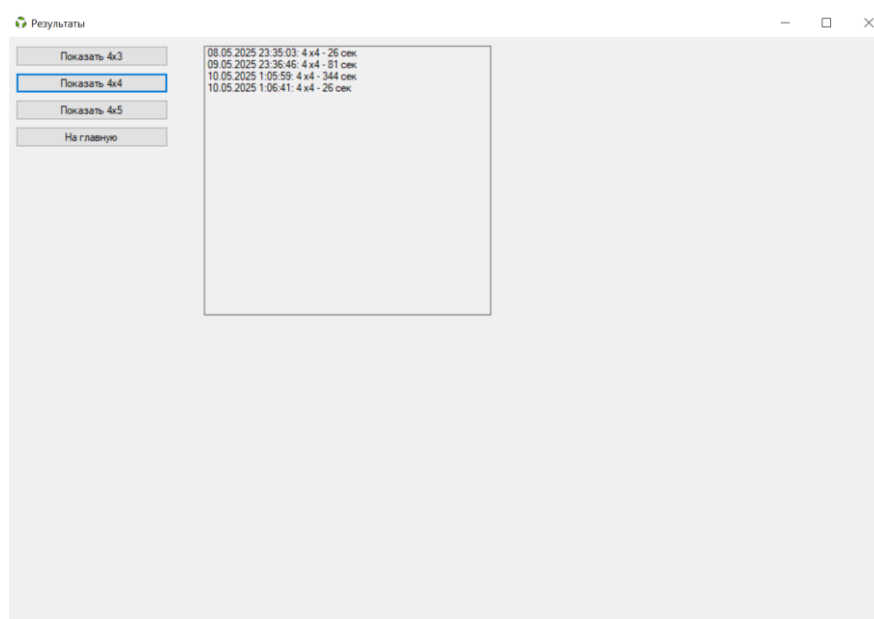


Рисунок 19 – Форма «Статистика»

Также, на главной форме на главном меню реализована кнопка «Справка», нажав на которую открывается справочная система. Она представлена на рисунке 20.

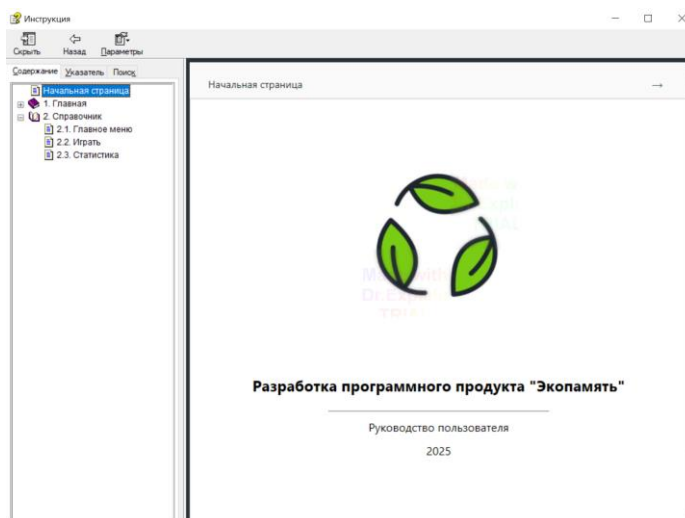


Рисунок 20 – Справка

Заключение

Цель данного проекта заключается в разработке программного продукта, который рассчитан на людей, заинтересованных в развитии различных качеств, таких как: повышение качества внимания, улучшение памяти и когнитивных функций, а также снижение уровня стресса. Игра может быть использована в образовательных и развивающих целях, как инструмент для повышения когнитивных способностей у детей и взрослых. Она также может быть применена в реабилитационных программах для людей, восстанавливающихся после травм головного мозга или страдающих от когнитивных нарушений.

В ходе выполнения данной курсовой работы было разработано и протестировано приложение «Экопамять» с графическим интерфейсом. Приложение содержит 3 уровня сложности, представляющих собой игровое поле с ячейками, которые случайным образом заполняются парными картинками, игровой процесс имеет различное визуальное и звуковое оформление, также после прохождения уровней пользователь может посмотреть статистику.

Для разработки данного проекта была использована среда Visual Studio и язык программирования C#, так как Платформа Windows Forms предлагает интуитивно понятный визуальный конструктор форм, позволяющий быстро создавать интерфейсы и взаимодействовать с элементами управления, а обширная библиотека стандартных компонентов облегчает создание различных элементов UI (меню, кнопки, поля ввода), значительно ускоряя процесс проектирования графического интерфейса. Наличие множества сторонних компонентных библиотек расширяет возможности разработчиков и позволяет добавлять новые функциональные элементы без необходимости писать весь код вручную.

Таким образом, выбор Windows Forms и C# становится оптимальным решением для быстрого запуска качественного продукта с низкой стоимостью разработки и высоким уровнем производительности, ориентированного преимущественно на пользователей операционных систем семейства Windows.

В заключении можно сказать, что данный программный продукт (игровое приложение) является законченной, полнофункциональной программой.

Поставленная задача выполнена в соответствии со всеми ранее задуманными требованиями, созданы и протестированы все необходимые компоненты проекта.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

Список использованных источников

1. Руководство. Создание приложения Windows Forms в Visual Studio с помощью C# [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio/> – Дата доступа: 15.04.2025.

2. Полное руководство по языку программирования C# 13 и платформе .NET 9 [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/tutorial/> – Дата доступа: 15.04.2025.

3. Руководство по C# [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> – Дата доступа: 16.04.2025.

					УП КПиЯП 2-40 01 01.35.40.07.25 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

Приложение А
Листинг программы

```

using System;
using System.IO;
using System.Windows.Forms;
namespace Memory
{
    public partial class LoadingForm : Form
    {
        private Timer timer;
        private int elapsedTime = 0;
        private string backgroundPath;
        public LoadingForm()
        {
            InitializeComponent();
            InitializeSplash();
        }
        private void InitializeSplash()
        {
            this.FormBorderStyle = FormBorderStyle.None;
            this.StartPosition =
            FormStartPosition.CenterScreen;
            this.Width = 1000;
            this.Height = 700;
            ProgressBar progressBar = new ProgressBar();
            progressBar.Style =
            ProgressBarStyle.Continuous;
            progressBar.Width = (int)(300);
            progressBar.Height = (int)(40);
            progressBar.Top = (this.ClientSize.Height -
            progressBar.Height) / 2;
            progressBar.Left = (this.ClientSize.Width -
            progressBar.Width) / 2;
            progressBar.Name = "progressBar";
            this.Controls.Add(progressBar);
            timer = new Timer();
            timer.Interval = 100;
            timer.Tick += (s, e) =>
            {
                elapsedTime += 100;
                progressBar.Value =
                Math.Min(progressBar.Maximum,
                (int)(progressBar.Maximum * elapsedTime / 3000));
                if (elapsedTime >= 3600)
                {
                    timer.Stop();
                    MainForm mainForm = new MainForm();
                    mainForm.Show();
                }
            }
        }
    }
}

```

```

        this.Hide();
    }
    timer.Start();
    private void LoadingForm_Load(object sender,
    EventArgs e)
    {
        backgroundPath =
        Path.Combine(AppDomain.CurrentDomain.BaseDirector
        y, "files/f2.bmp");
        this.BackgroundImage =
        System.Drawing.Image.FromFile(backgroundPath);
    }
}
using System;
using System.Drawing;
using System.IO;
using System.Media;
using System.Windows.Forms;
using WMPLib;
namespace Memory
{
    public partial class MainForm : Form
    {
        private string backgroundPath;
        private WindowsMediaPlayer mediaPlayer = new
        WindowsMediaPlayer();
        public MainForm()
        {
            InitializeComponent();
        }
        private void
        comboBox1_SelectedIndexChanged(object sender,
        EventArgs e)
        {
            SetBackgroundImage(comboBox1.SelectedIndex + 1);
        }
        private void SetBackgroundImage(int index)
        {
            backgroundPath =
            Path.Combine(AppDomain.CurrentDomain.BaseDirector
            y, "files", $"f{index}.bmp");
            if (File.Exists(backgroundPath))
            {
                this.BackgroundImage =
                System.Drawing.Image.FromFile(backgroundPath);
                this.BackgroundImageLayout =
                ImageLayout.Stretch;
            }
        }
        private void
        comboBox2_SelectedIndexChanged(object sender,
        EventArgs e)
        {
        }
    }
}

```

```

        { mediaPlayer.settings.setMode("loop", true); //
Включаем повтор
        switch (comboBox2.SelectedIndex)
        { case 0: mediaPlayer.URL = "files/calm1.mp3"; break;
case 1: mediaPlayer.URL = "files/dinamic1.mp3";break;
case 2: mediaPlayer.URL = "files/dinamic2.mp3"; break;
case 3: mediaPlayer.URL = "files/zvuki-gitary.mp3";
break;
case 4: mediaPlayer.URL = "files/calm2.mp3"; break;}
        mediaPlayer.controls.play();}
        private void MainForm_Load(object sender,
EventArgs e)
        { backgroundImage =
Path.Combine(AppDomain.CurrentDomain.BaseDirector
y, "files/f2.bmp");
        this.BackgroundImage =
System.Drawing.Image.FromFile(backgroundPath);
        this.StartPosition =
FormStartPosition.CenterScreen;
        this.Width = 1000;
        this.Height = 700;
        label1.Text = "Экопамять";
        label1.Font = new Font("Arial", 20);
        label1.BackColor = Color.Transparent;}
        private void button1_Click(object sender,
EventArgs e)
        { PlayForm playForm = new PlayForm();
        playForm.Show();
        this.Hide();}
        private void PlayMusic(string filename)
        { SoundPlayer player = new SoundPlayer(filename);
        player.PlaySync();}
        private void button2_Click(object sender,
EventArgs e)
        { ResultsForm resultsForm = new ResultsForm();
        resultsForm.Show();
        this.Hide();}
        private void button3_Click(object sender,
EventArgs e)

```

```

        { MessageBox.Show(" Цель игры: Найти все
пары одинаковых карточек за минимальное
время.\n\n" + " Как играть:\n" + "Выберите уровень
сложности (размер игрового поля: 4×3, 4×4 или 4×5).
На экране появятся карточки, перевёрнутые
рубашкой вверх. Нажимайте на карточки, чтобы
открыть их. Ваша задача — запомнить изображения
и найти одинаковые пары. Если изображения
совпадают — пара исчезает с поля. Если нет —
карточки автоматически закрываются через короткое
время.\n\n" + " Результат:\n" + "По завершении игры
будет показано, за сколько секунд вы справились.
Результат сохраняется в таблицу рекордов для
выбранного уровня.", " Правила игры",
MessageBoxButtons.OK,
MessageBoxIcon.Information);}
        private void button4_Click(object sender,
EventArgs e)
        { Help.ShowHelp(this, helpProvider1.HelpNamespace);
        }}}
using System;
using System.Collections.Generic;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Windows.Forms;
namespace Memory
{ public partial class PlayForm : Form
    { private int gridSize = 3; // по умолчанию 4x4
    private List<Image> images; // список всех
изображений
    private List<Card> cards = new List<Card>();
    private Card firstSelected = null;
    private Card secondSelected = null;
    private Timer gameTimer;
    private int secondsElapsed = 0;
    private string statsFile = "stats.txt";
    public PlayForm()

```

```

{ InitializeComponent();
    //this.FormBorderStyle = FormBorderStyle.None;
this.StartPosition = FormStartPosition.CenterScreen;
    this.Width = 1000;
    this.Height = 700;
    // Создаем таймер
    gameTimer = new Timer();
    gameTimer.Interval = 1000; // 1 секунда
    gameTimer.Tick += GameTimer_Tick;
    // Создаем панель для карточек
    FlowLayoutPanel panel = new
FlowLayoutPanel();
    panel.Dock = DockStyle.Fill;
    panel.WrapContents = true;
    panel.AutoSize = false;
    this.Controls.Add(panel);
    // Создаем метку для таймера
    Label lblTimer = new Label();
    lblTimer.Name = "lblTimer";
    lblTimer.Text = "Время: 0 сек";
    lblTimer.Font = new Font("Arial", 14);
    lblTimer.AutoSize = true;
    lblTimer.Location = new Point(10, 10);
    this.Controls.Add(lblTimer);
    lblTimer.BringToFront();
    // Создаем кнопки для выбора уровня
    Panel topPanel = new Panel();
    topPanel.Height = 50;
    topPanel.Dock = DockStyle.Top;
    this.Controls.Add(topPanel);
    Button btn4x4 = new Button() { Text = "4x4",
Tag = 4, Width = 60, Left = 300, Top = 10 };
    btn4x4.Click += LevelButton_Click;
    topPanel.Controls.Add(btn4x4);
    Button btn4x5 = new Button() { Text = "4x5",
Tag = 5, Width = 60, Left = 400, Top = 10 };
    btn4x5.Click += LevelButton_Click;
    topPanel.Controls.Add(btn4x5);

```

```

    Button btn4x3 = new Button() { Text = "4x3",
Tag = 3, Width = 60, Left = 200, Top = 10 };
    btn4x3.Click += LevelButton_Click;
    topPanel.Controls.Add(btn4x3);
    Button btnBackToMain = new Button()
    { Text = "На главную",
    Width = 100,
    Height = 30,
    Left = 700,
    Top = 10 };
    btnBackToMain.Click += BtnBackToMain_Click;
    topPanel.Controls.Add(btnBackToMain);
    // Загрузка изображений
    LoadImages();
    // Запуск игры с выбранным уровнем
    StartGame(gridSize); }
//Загрузка изображений для карточек
private void LoadImages()
{ images = new List<Image>();
    for (int i = 1; i <= 10; i++)
    { Image img = Image.FromFile($"files/img{i}.png");
        images.Add(img); } }
    private void LevelButton_Click(object sender,
EventArgs e)
    { Button btn = sender as Button;
        int size = (int)btn.Tag;
        StartGame(size); }
    private void StartGame(int size)
    { gridSize = size;
        secondsElapsed = 0;
        var lblTimer = this.Controls.Find("lblTimer",
true).FirstOrDefault() as Label;
        if (lblTimer != null)
            lblTimer.Text = "Время: 0 сек";
        gameTimer.Stop();
    // Удаляем старые карточки
    var oldTable =
this.Controls.OfType<TableLayoutPanel>().FirstOrDefa
ult();

```

```

if (oldTable != null)
    this.Controls.Remove(oldTable);
cards.Clear();
firstSelected = null;
secondSelected = null;
int totalCards = 4 * size;
int rows = 4;
int cols = size;
// Выбор и перемешивание изображений
var rnd = new Random();
var selectedImages = images.OrderBy(x =>
rnd.Next()).Take(totalCards / 2).ToList();
var gameImages = new List<Image>();
foreach (var img in selectedImages)
{
    gameImages.Add(img);
    gameImages.Add(img);
}
gameImages = gameImages.OrderBy(x =>
rnd.Next()).ToList();
// Создание новой таблицы
TableLayoutPanel table = new
TableLayoutPanel();
table.RowCount = rows;
table.ColumnCount = cols;
table.Dock = DockStyle.None;
table.AutoSize = true;
table.AutoSizeMode =
AutoSizeMode.GrowAndShrink;
table.Margin = new Padding(10);
table.Padding = new Padding(20);
table.BackColor = Color.LightGray;
table.CellBorderStyle =
TableLayoutPanelCellBorderStyle.Single;
// Устанавливаем фиксированный размер строк и
столбцов
for (int r = 0; r < rows; r++)
    table.RowStyles.Add(new
RowStyle(SizeType.Absolute, 80));
for (int c = 0; c < cols; c++)

```

```

table.ColumnStyles.Add(new
ColumnStyle(SizeType.Absolute, 80));
this.Controls.Add(table);
table.BringToFront();
// Добавление карточек
int index = 0;
for (int r = 0; r < rows; r++)
{
    for (int c = 0; c < size; c++)
    {
        Card card = new Card(gameImages[index]);
        card.Button.Click += Card_Click;
        cards.Add(card);
        table.Controls.Add(card.Button, c, r);
        index++;
    }
}
gameTimer.Start();
// Центрируем таблицу по форме
table.Anchor = AnchorStyles.None; //
отключаем привязку к сторонам
// Обновляем расположение вручную
table.Location = new Point(
(this.ClientSize.Width - table.PreferredSize.Width) / 2,
(this.ClientSize.Height -
table.PreferredSize.Height) / 2);
private void Card_Click(object sender, EventArgs e)
{
    Button btn = sender as Button;
    Card clickedCard = cards.First(c => c.Button == btn);
    if (clickedCard.IsMatched || clickedCard.IsRevealed)
        return;
    if (firstSelected != null && secondSelected != null)
        return;
    // Раскрываем карточку
    clickedCard.Reveal();
    if (firstSelected == null)
    {
        firstSelected = clickedCard;
    }
    else if (secondSelected == null)
    {
        secondSelected = clickedCard;
    }
    // Проверка совпадения
    if (firstSelected.Image == secondSelected.Image)
    {
        firstSelected.IsMatched = true;
        secondSelected.IsMatched = true;
    }
}

```

```

        Timer matchDelayTimer = new Timer();
matchDelayTimer.Interval = 300; //0,3 секунды
matchDelayTimer.Tick += (s2, e2) =>
{
    matchDelayTimer.Stop();
    firstSelected.Button.Visible = false;
    secondSelected.Button.Visible = false;
    firstSelected = null;
    secondSelected = null;

    Проверка завершения игры после задержки
    if (cards.All(c => c.IsMatched))
    {
        gameTimer.Stop();
        MessageBox.Show($"Поздравляем! Вы прошли
уровень за {secondsElapsed} секунд.", "Победа");
        SaveResult(secondsElapsed);
        matchDelayTimer.Start();
    }
    else
    {
        // Не совпадают, задержка перед закрытием
        Timer delayTimer = new Timer();
        delayTimer.Interval = 700; // 0,7 секунды
        delayTimer.Tick += (s, args) =>
        {
            delayTimer.Stop();
            firstSelected.Hide();
            secondSelected.Hide();
            firstSelected = null;
            secondSelected = null;
            delayTimer.Start();
        }

        private void GameTimer_Tick(object sender,
        EventArgs e)
        {
            secondsElapsed++;
            var lblTimer = this.Controls.Find("lblTimer",
            true).FirstOrDefault() as Label;
            if (lblTimer != null)
            {
                lblTimer.Text = $"Время: {secondsElapsed} сек";
            }
            private void SaveResult(int time)
            {
                string statsFileName = $"stats_{gridSize}.txt";
                string folderPath =
                AppDomain.CurrentDomain.BaseDirectory; // Путь к
                папке Debug

```

```

                string fullPath = Path.Combine(folderPath,
                statsFileName);
                string entry = $"{DateTime.Now}: 4 x {gridSize} -
                {time} сек";
                File.AppendAllText(fullPath, entry +
                Environment.NewLine);
                private void BtnBackToMain_Click(object sender,
                EventArgs e)
                {
                    this.Hide();
                    MainForm mainForm = new MainForm();
                    mainForm.Show();
                }
                // Класс карточки
                public class Card
                {
                    public Button Button { get; private set; }
                    public Image Image { get; private set; }
                    public bool IsRevealed { get; private set; } = false;
                    public bool IsMatched { get; set; } = false;
                    private Image backImage;
                    public Card(Image image)
                    {
                        Image = image;
                        backImage = GenerateBackImage(); // создаем задний
                        образ
                        Button = new Button();
                        Button.Width = 80;
                        Button.Height = 80;
                        Button.Margin = new Padding(2);
                        Button.BackgroundImageLayout = ImageLayout.Stretch;
                        Hide();
                    }
                    public void Reveal()
                    {
                        Button.BackgroundImage = Image;
                        IsRevealed = true;
                    }
                    public void Hide()
                    {
                        Button.BackgroundImage = backImage;
                        IsRevealed = false;
                    }
                    //генерация изображения для обратной стороны
                    private Image GenerateBackImage()
                    {
                        Bitmap bmp = new Bitmap(80, 80);
                        using (Graphics g = Graphics.FromImage(bmp))
                        {
                            g.Clear(Color.White);
                        }
                    }
                }

```



```

        return bmp;}}}}
using System;
using System.IO;
using System.Windows.Forms;
namespace Memory
{public partial class ResultsForm : Form
    { private Label lblResults;
      public ResultsForm()
      { InitializeComponent();
        InitializeUI();}
      private void InitializeUI()
      { this.Text = "Результаты";
        this.Size = new System.Drawing.Size(1000, 700);
        Button btn4x3 = new Button { Text = "Показать
4x3", Top = 10, Left = 10, Width = 170 };
        Button btn4x4 = new Button { Text = "Показать
4x4", Top = 40, Left = 10, Width = 170 };
        Button btn4x5 = new Button { Text = "Показать
4x5", Top = 70, Left = 10, Width = 170 };
        Button btnBack = new Button { Text =
"На главную", Top = 100, Left = 10, Width = 170 };
        btn4x3.Click += (s, e) => ShowResults(3);
        btn4x4.Click += (s, e) => ShowResults(4);
        btn4x5.Click += (s, e) => ShowResults(5);
        btnBack.Click += (s, e) =>
        { this.Hide();
          var mainForm = new MainForm();
          mainForm.Show(); };
        lblResults = new Label
        { Top = 10,
          Left = 220,
          Width = 320,
          Height = 300,
          AutoSize = false,
          BorderStyle = BorderStyle.FixedSingle };
        this.Controls.Add(btn4x3);
        this.Controls.Add(btn4x4);
        this.Controls.Add(btn4x5);
        this.Controls.Add(btnBack);

```

```

        this.Controls.Add(lblResults);}
      private void ShowResults(int gridSize)
      { string path =
Path.Combine(AppDomain.CurrentDomain.BaseDirector
y, $"stats_{gridSize}.txt");
        if (File.Exists(path))
        { lblResults.Text = File.ReadAllText(path);}
        else
        { lblResults.Text = $"Файл результатов для
4x{gridSize} не найден.";}}}

```

Приложение Б
Тест-кейсы

Изм.		Таблица Б.1 – Тестирование					
Исст.		Идентификатор	Приоритет	Заглавие и шаги выполнения	Ожидаемый результат	Фактический результат	Статус
		1	2	3	4	5	6
№ док-м.		T_01	крайне высокий	Запуск приложения 1. Запустить файл «Экопамять.exe».	1. Отображается экран загрузки. Загружается главное меню игры с кнопками: «Играть», «Статистика», «Помощь», «Справка» и сворачиваемые группы «Выбрать фон» и «Выбрать музыку».	1. Отобразился экран загрузки. Загрузилось главное меню игры с кнопками: «Играть», «Статистика», «Помощь», «Справка» и сворачиваемые группы «Выбрать фон» и «Выбрать музыку».	Пройдено
Подпись		T_02	Средний	Справка 1. Нажать на кнопку «Справка» в главном меню. 2. Нажать на раздел «Главная». 3. Нажать на подраздел «О программе». 4. Нажать на подраздел «Об авторе». 5. Нажать на раздел «Справочник». 6. Нажать на подраздел «Главное меню». 7. Нажать на подраздел «Играть». 8. Нажать на подраздел «Статистика». 9. Нажать на кнопку «Закрыть».	1. Загружается окно справки с начальной страницей и разделами: «Главная» с подразделами «О программе», «Об авторе»; раздел «Справочник» с подразделами «Главное меню», «Играть», «Статистика». 2. Осуществляется переход на страницу «Главная» с базовой информацией. 3. Осуществляется переход на страницу «О программе» с основной информацией о приложении. 4. Осуществляется переход на страницу «Об авторе» с основной информацией об авторе приложения. 5. Осуществляется переход на страницу «Справочник» с информацией по главному экрану. 6. Осуществляется переход на страницу «Главное меню» с основной информацией с информацией по главному экрану. 7. Осуществляется переход на страницу «Играть» с основной информацией об игре. 8. Осуществляется переход на страницу «Статистика» с информацией о статистике прохождений уровней. 9. Закрывается справка.	1. Загрузилось окно справки с начальной страницей и разделами: «Главная» с подразделами «О программе», «Об авторе»; раздел «Справочник» с подразделами «Главное меню», «Играть», «Статистика». 2. Осуществился переход на страницу «Главная» с базовой информацией. 3. Осуществился переход на страницу «О программе» с основной информацией о приложении. 4. Осуществился переход на страницу «Об авторе» с основной информацией об авторе приложения. 5. Осуществился переход на страницу «Справочник» с информацией по главному экрану. 6. Осуществился переход на страницу «Главное меню» с основной информацией с информацией по главному экрану. 7. Осуществился переход на страницу «Играть» с основной информацией об игре. 8. Осуществился переход на страницу «Статистика» с информацией о статистике прохождений уровней. 9. Закрылась справка.	Пройдено
Дата		T_03	высокий	Кнопка «Выход» 1. Нажать на кнопку «Выход» на панели управления окном.	1. Приложение закрывается.	1. Приложение закрылось.	Пройдено
УП КП/АП 2-40 01 01.35.40.07.25 ПЗ							
Исст.	27						

Продолжение таблицы Б.1

1	2	3	4	5	6
T_04	высокий	Главное меню 1. Нажать на сворачиваемую группу «Выбрать фон». 2. Нажать на кнопку «Фон 3». 3. Нажать на сворачиваемую группу «Выбрать музыку» 4. Нажать на кнопку «Музыка 1» 5. Нажать на кнопку «Правила» 6. На всплывающем сообщении нажать на кнопку «Ок».	1. Разворачивается сворачиваемая группа для выбора фона. 2. Загружается изображение «Фон 3». 3. Разворачивается сворачиваемая группа для выбора музыки. 4. Запускается аудио «Музыка 1». 5. Открывается всплывающее сообщение с правилами игры. 6. Всплывающее сообщение закрывается.	1. Развернулась сворачиваемая группа для выбора фона. 2. Загрузилось изображение «Фон 3». 3. Развернулась сворачиваемая группа для выбора музыки. 4. Запустилось аудио «Музыка 1». 5. Открылось всплывающее сообщение с правилами игры. 6. Всплывающее сообщение закрылось.	Пройдено
T_05	высокий	Форма «Играть», прохождение игры 1. Нажать на кнопку «Играть» в главном меню. 2. На форме «Играть» нажать на кнопку «4*4» для выбора уровня. 3. На игровом поле нажать на две одинаковые карточки. 4. На игровом поле нажать на две разные карточки. 5. На игровом поле нажать на две последние карточки. 6. На форме «Играть» нажать на кнопку «На главную».	1. Загружается форма «Играть» с кнопками «4*3», «4*4», «4*5», «На главную». 2. Загружается игровое поле размером 4*4. 3. Карточки пропадают с игрового поля. 4. Карточки переворачиваются на игровом поле. 5. Карточки пропадают с игрового поля, открывается всплывающее сообщение о времени, затраченном на прохождение уровня. 6. Форма «Играть» закрывается. Загружается Главное меню.	1. Загрузилась форма «Играть» с кнопками «4*3», «4*4», «4*5», «На главную». 2. Загрузилось игровое поле размером 4*4. 3. Карточки пропали с игрового поля. 4. Карточки перевернулись на игровом поле. 5. Карточки пропали с игрового поля. Открылось всплывающее сообщение о времени, затраченном на прохождение уровня. 6. Форма «Играть» закрылась. Загрузилось Главное меню.	Пройдено
T_06	высокий	Форма «Статистика» 1.Нажать на кнопку «Статистика» на главном меню. 2.Нажать на кнопку «Показать 4*3». 3. На форме «Статистика» нажать на кнопку «На главную».	1. Загружается форма «Играть» с кнопками «Показать 4*3», «Показать 4*4», «Показать 4*5», «На главную». 2. Появляется информация о результатах прохождения уровня 4*3: дата прохождения, время прохождения, затраченное время на прохождение уровня. 3. Форма «Статистика» закрывается. Загружается Главное меню.	1. Загрузилась форма «Играть» с кнопками «Показать 4*3», «Показать 4*4», «Показать 4*5», «На главную». 2. Появилась информация о результатах прохождения уровня 4*3: дата прохождения, время прохождения, затраченное время на прохождение уровня. 3. Форма «Статистика» закрылась. Загрузилось Главное меню.	Пройдено