

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 познавательного приложения «Правила оказания
первой помощи» _____

Ссылка на проект: <https://github.com/Angellina-Kot/First-aid.git> _____

Выполнила _____ А.С. Кот
(инициалы, фамилия)

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

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..... | 24 |

| | | | | | | | | | | | |
|-----------|--------|---------|---------|------|--|--|--|------|------|--------|----|
| | | | | | УП КПиЯП 2-40 01 01.35.40.06.25 ПЗ | | | | | | |
| | | | | | | | | | | | |
| Изм. | Лист | №докум. | Подпись | Дата | Разработка программного продукта «Правила оказания первой помощи»» | | | Лит. | Лист | Листов | |
| Разраб. | Кот | | | | | | | | | 2 | 27 |
| Пров. | Бабуль | | | | | | | | | | |
| | | | | | | | | | | | |
| Н. контр. | | | | | | | | | | | |
| Утв. | | | | | | | | | | | |

Введение. Описание структуры предприятия

На данной учебной практике была поставлена задача разработать познавательное приложение «Правила оказания первой помощи».

Цель проекта заключается в создании приложения, которое позволит пользователю получить и закрепить знания не только о первой помощи в различных ситуациях, но и признаки, по которым можно определить какая именно помощь нужна пострадавшему.

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 представлена диаграмма компонентов.

Четвертый раздел – «Тестирование». В нем описано функциональное тестирование данной программы, смоделированы все основные действия пользователя при работе с программой.

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 «Правила оказания первой помощи».

Цель разработки данного программного продукта заключается в создании познавательного приложения для предоставления пользователям доступной и актуальной информации о правилах оказания первой медицинской помощи в различных ситуациях.

Назначение: данный программный продукт ориентирован на пользователей, которые хотят ознакомиться с правилами оказания первой помощи пострадавшим и проверить свои знания в этой области.

Актуальность темы данного программного продукта заключается в необходимости знаний оказания первой помощи пострадавшим, так как в экстренных ситуациях знание основ первой помощи может спасти жизнь. Многие люди сталкиваются с травмами или болезнями в повседневной жизни, и умение правильно действовать в таких ситуациях значительно увеличивает шансы на успешное восстановление пострадавшего. Современные условия жизни характеризуются увеличением количества аварий, природных катастроф, техногенных происшествий и несчастных случаев. Эти события требуют быстрого реагирования и знания правил оказания первой медицинской помощи, поскольку от своевременных действий окружающих часто зависит жизнь пострадавших.

Периодичность использования данного программного продукта неограниченна.

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 2021 – для написания документации к программному продукту;
- Dr.Explain – инструмент разработки пользовательской документации;
- GitHub – веб-сервис для хостинга IT-проектов;
- Inkscape – инструмент для разработки иллюстраций.

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.

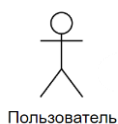


Рисунок 1 – Графическое обозначение актера

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.

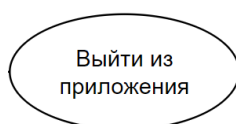


Рисунок 2 – Графическое обозначение вариантов использования

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.

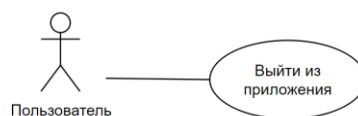


Рисунок 3 – Графическое обозначение отношения ассоциации

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.

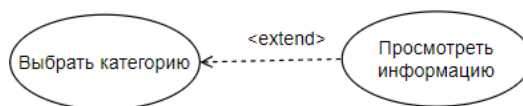


Рисунок 4 – Графическое обозначение отношения расширения

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.

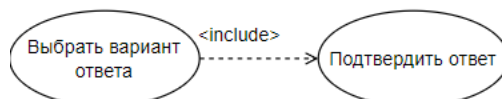


Рисунок 5 – Графическое обозначение отношения включения

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

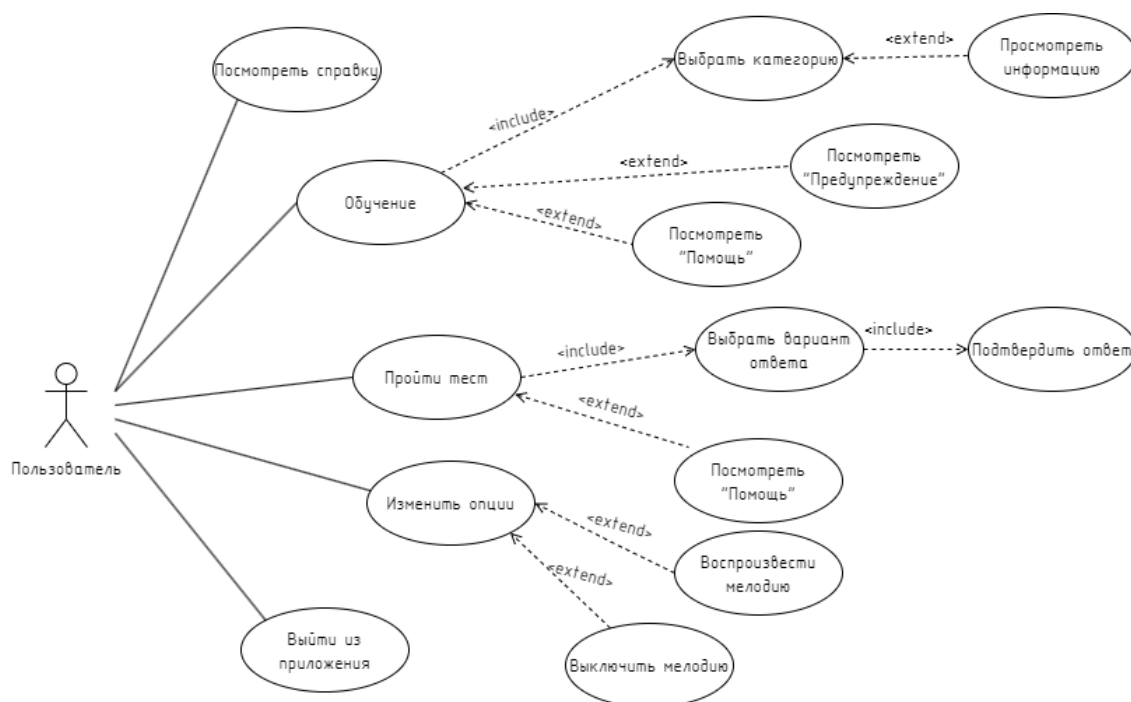


Рисунок 6 – Диаграмма вариантов использования

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.

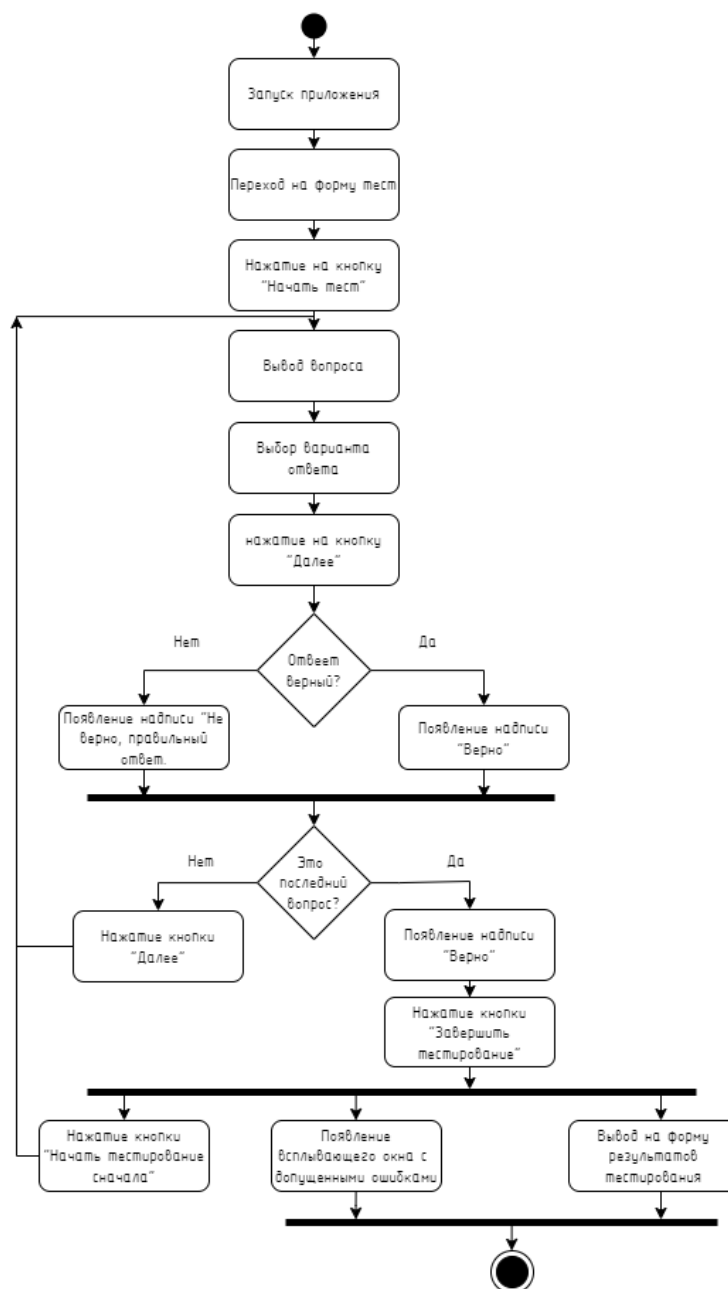


Рисунок 7 – Диаграмма деятельности

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.

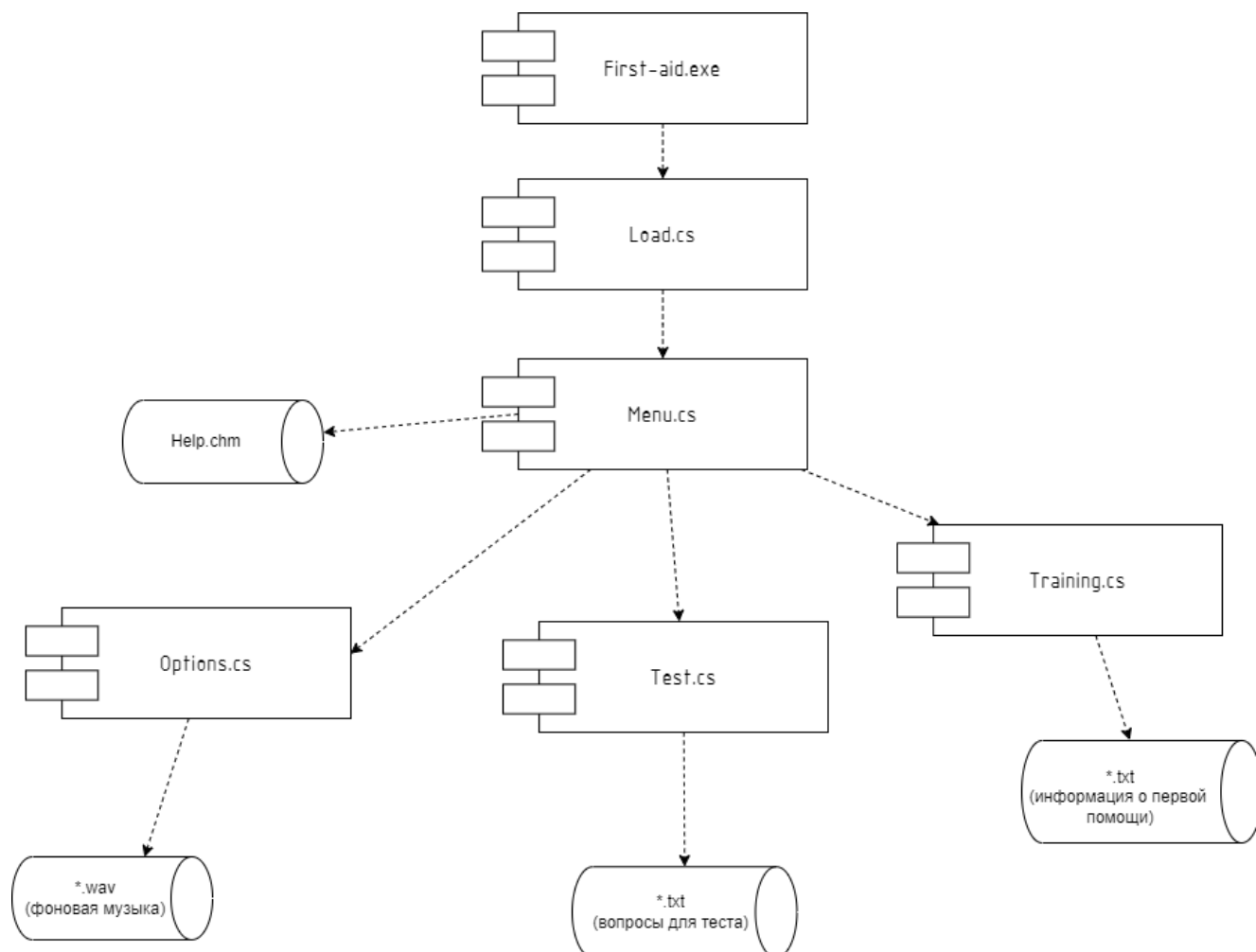


Рисунок 8 – Диаграмма компонентов

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.

При разработке приложения необходимо будет провести тестирование.

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 |
|--------------|------------|---|----------------------|
| Кот Ангелина | 04.05.2025 | Разработка тестов | 1 |
| Кот Ангелина | 05.05.2025 | Тестирование познавательного приложения | 0,5 |
| Кот Ангелина | 05.05.2025 | Исправление найденных ошибок | 1 |
| Кот Ангелина | 06.05.2025 | Проведение регрессионного тестирования | 1 |
| Кот Ангелина | 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 программного продукта заключается в предоставлении возможностей пользователю ознакомиться с действиями в ситуациях, когда он сам или кто-то попал в беду.

Создаваемое приложение будет рассчитано на любого пользователя.

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: AMD Ryzen 7 5800H with Radeon Graphics, 3201 МГц, ядер: 8, логических процессоров: 16;
- объем оперативной памяти: 16.00 GB;
- ОС: Windows 11 Pro.

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл setup.exe. Появится мастер установки игрового приложения «Правила оказания первой помощи», представленный на рисунке 9.

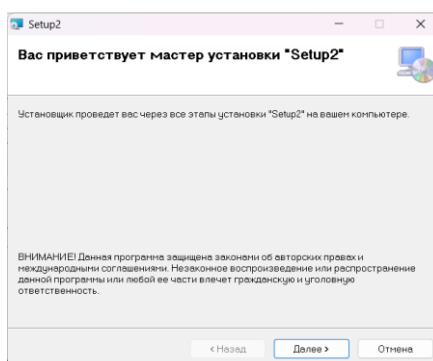


Рисунок 9 – Мастер установки игрового приложения «Правила оказания первой помощи»

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

| | | | | | | |
|------|------|----------|---------|------|------------------------------------|------|
| | | | | | УП КПиАП 2-40 01 01.35.40.06.25 ПЗ | Лист |
| | | | | | | 11 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

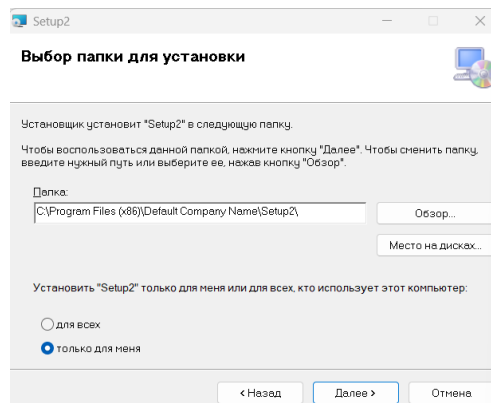


Рисунок 10 – Выбор папки установки

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

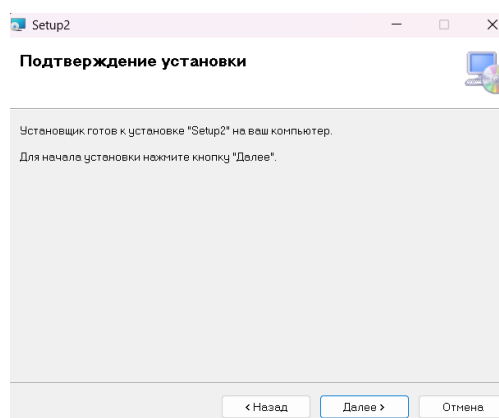


Рисунок 11 – Окно подтверждения установки

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

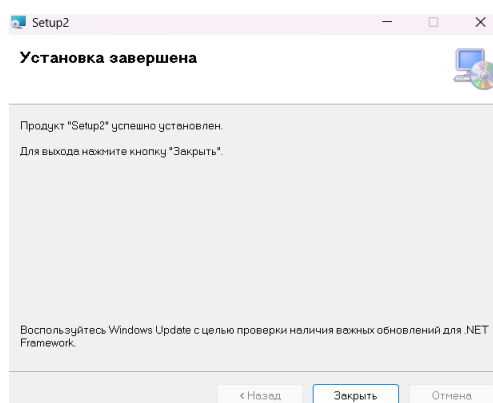


Рисунок 12 – Окно «Установка завершена»

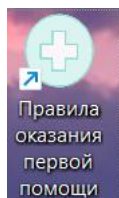


Рисунок 13 – Ярлык приложения

5.3 Выполнение программы

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

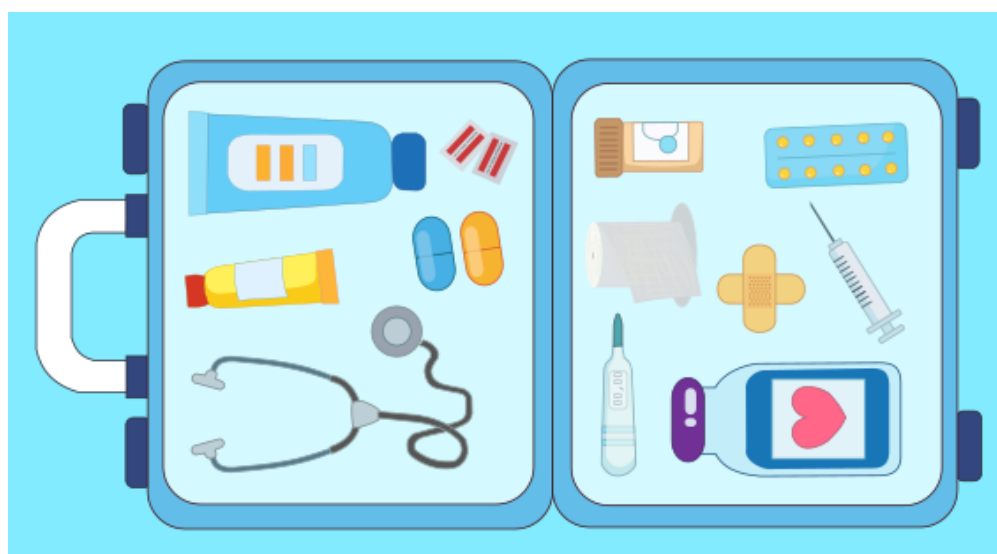


Рисунок 14 – Загрузка приложения

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

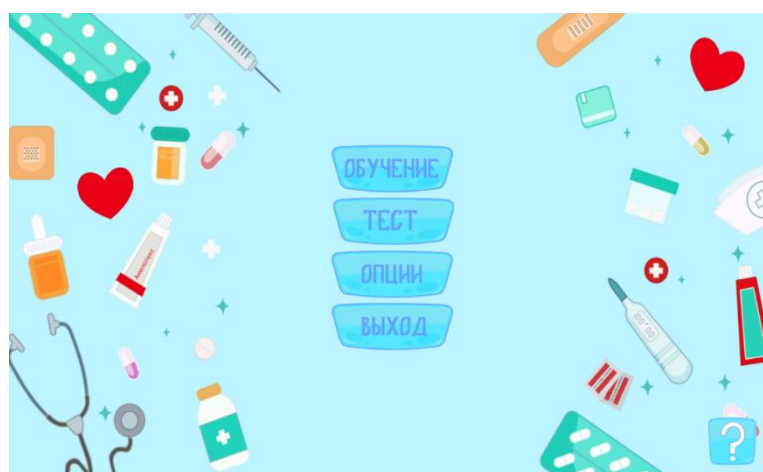


Рисунок 15 – Главное меню

У пользователя есть возможность нажать на кнопки «Обучение», «Тест», «Опции», «Выход» и «Помощь».

При нажатии на кнопку «Помощь», представленную на рисунке 16, будет открыта справка.



Рисунок 16 – Кнопка «Помощь»

При нажатии на кнопку «Выход» приложение будет закрыто.

При нажатии на кнопку «Опции» всплывает форма настройки звука, расположенная на рисунке 17.



Рисунок 17 – Форма «Опции»

Выбрать мелодию можно нажав по соответствующей кнопке. Также можно прекратить все звуки, нажав на кнопку «Стоп», расположенную справа от кнопок выбора мелодий.

При нажатии на кнопку «Назад», расположенную на рисунке 18 игрок возвращается на главное меню.



Рисунок 18 – Кнопка «Назад»

При нажатии на кнопку «Обучение» пользователь переходит на форму, где он может выбрать травму или поражение из списка, и в правой части экрана появится информация о том, что именно представляет собой эта категория состояния пострадавшего, её признаки и первая помощь в этом случае. Форма «Обучение» представлена на рисунке 19.

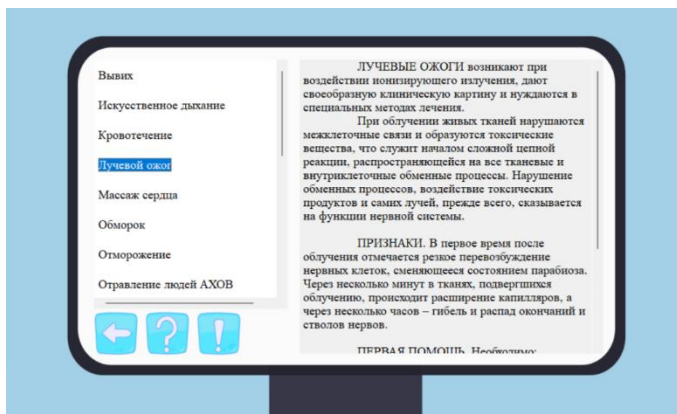


Рисунок 19 – Форма «Обучение»

При нажатии на кнопку «Назад», расположенную на рисунке 18 игрок возвращается на главное меню.

При нажатии на кнопку «Внимание», которая представлена на рисунке 20, пользователь может ознакомиться с важной информацией, которая отобразится в правой части формы.



Рисунок 20 – Кнопка «Внимание»

При нажатии на кнопку «Тест» в главном меню, пользователь переходит на форму «Тест», представленную на рисунке 20 которая направлена на закрепление полученных знаний.

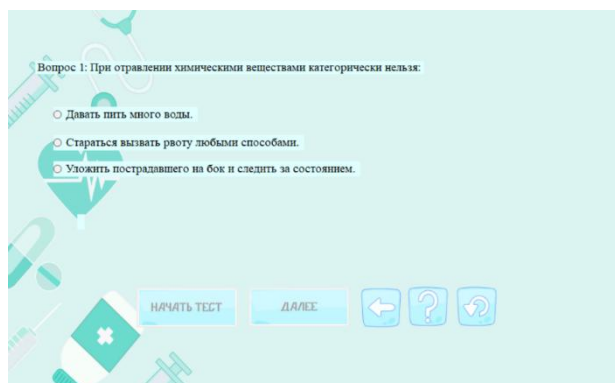


Рисунок 20 – Форма «Тест»

После нажатия на кнопку «Начать тест», расположенную на рисунке 21, пользователю будет доступен первый вопрос, кнопка «Далее» доступна только после выбора варианта ответа.

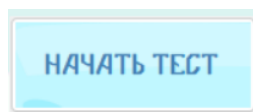


Рисунок 21 – Кнопка «Начать тест»

После прохождения теста появляется всплывающее окно со списком вопросов, в которых пользователь допустил ошибку, также на саму форму выводятся результаты тестирования, и появляется кнопка «Начать тестирование сначала», которая представлена на рисунке 22.

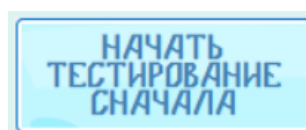


Рисунок 22 – Кнопка «Начать тестирование сначала»

Для того, чтобы начать тест заново в любой момент можно использовать кнопку «Реверс», представленную на рисунке 23.



Рисунок 23 – Кнопка «Реверс»

Чтобы узнать правила, воспользуйтесь кнопкой «Помощь», представленной на рисунке 16.

Рекомендуется проходить тест после ознакомления с правилами оказания первой помощи пострадавшим.

Заключение

Целью данной учебной практики была в разработке программного продукта, который позволит массовому пользователю ознакомиться с правилами оказания первой помощи пострадавшим в различных ситуациях.

Данное приложение рассчитано для любого рода пользователей. Применять его смогут не только учащиеся, но и просто люди, интересующиеся данной темой.

Благодаря этому проекту будет легче узнать и познакомиться с тем, как помочь людям, которые попали в беду.

Приложение предоставит краткую и достоверную информацию, которая будет полезна для ознакомления как детям, так и взрослым.

В ходе выполнения учебной практики был разработан и протестирован программный продукт «Правила оказания первой помощи» с графическим интерфейсом.

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 (познавательное 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 01.35.40.18.25 ПЗ | Лист |
| Изм. | Лист | № докум. | Подпись | Дата | | 18 |

Приложение А
Листинг программы

```

using System;
using System.Windows.Forms;
namespace First_aid
{ public partial class Load : Form
    { private int currentIndex = 0;
      public Load()
      { ApplicationForms.Load = this;
        InitializeComponent();
        // Устанавливаем интервал таймера (0,2 секунды)
        timer.Interval = 200;    }
      private void timer_Tick(object sender, EventArgs e) //
Проверяем, закончились ли изображения
      {if (currentIndex >= 14)
      { // Остановка таймера и открытие новой формы
        timer.Stop();
        OpenMainForm(); }
      else { // Загрузка следующего изображения
        switch (currentIndex
        { case 1: pictureBox1.Visible = true; break;
          case 2: pictureBox2.Visible = true; break;
          case 3: pictureBox3.Visible = true; break;
          case 4: pictureBox4.Visible = true; break;
          case 5: pictureBox5.Visible = true; break;
          case 6: pictureBox6.Visible = true; break;
          case 7: pictureBox7.Visible = true; break;
          case 8: pictureBox8.Visible = true; break;
          case 9: pictureBox9.Visible = true; break;
          case 10: pictureBox10.Visible = true; break;
          case 11: pictureBox11.Visible = true; break;
          case 12: pictureBox12.Visible = true; break;
          case 13: pictureBox13.Visible = true; break; }
        currentIndex++; } }
      private void OpenMainForm()
      { this.Hide();
        Menu form = new Menu(this);
        form.Show(); }
      protected override void OnLoad(EventArgs e)
      { base.OnLoad(e); timer.Start(); // Начинаем отсчет
таймера сразу после открытия формы } }

```

```

using System;
using System.Windows.Forms;
namespace First_aid
{ public partial class Menu : Form
    { Test formT = new Test();
      public Menu(Load load)
      { ApplicationForms.Menu = this;
        InitializeComponent();
        private void ExitGame_but_Click(object sender,
EventArgs e)
        { ApplicationForms.Load.Close(); }
        private void Training_but_Click(object sender,
EventArgs e)
        { this.Hide();
          Training form = new Training();
          form.Show(); }
        private void Option_Butt_Click(object sender,
EventArgs e)
        { Options form1 = new Options();
          form1.ShowDialog();
          private void pictureBox1_Click(object sender,
EventArgs e)
          { this.Hide();
            formT.Show(); }
          private void Help_Butt_Click(object sender,
EventArgs e)
          { Help.ShowHelp(this,
helpProvider1.HelpNamespace); } } }
using System;
using System.IO;
using System.Windows.Forms;
namespace First_aid
{ public partial class Training : Form
    { public Training()
      { ApplicationForms.Training = this;
        InitializeComponent();
        private void listView1_Click(object sender, EventArgs e)
        { if (listView1.SelectedItems.Count > 0)
          { string a = listView1.SelectedItems[0].Tag.ToString();

```

```

        StreamReader reader = new StreamReader(a); //
Открываем файл для чтения
        textBox1.Text = reader.ReadToEnd(); } }
private void Exit_Butt_Click(object sender, EventArgs e) {
    textBox1.Clear();
        this.Close();
        ApplicationForms.Menu.Show();}
private void Remark_Butt_Click(object sender, EventArgs
e)
    { StreamReader reader = new
StreamReader("Внимание.txt"); // Открываем файл для
чтения  textBox1.Text = reader.ReadToEnd();}
private void Help_Butt_Click(object sender, EventArgs e)
{ MessageBox.Show(" Чтобы начать обучение
необходимо кликнуть \n" + "на нужную категорию в
левой колонке, в правой \n" + "части появится описание
категории, признаки и \n" + "первая
помощь пострадавшему.\n" + " Перед
началом обучение рекомендуется \n" +
"ознакомиться с важной информацией, \n" +
"нажав на кнопку '!"); } }
using System;
using System.Windows.Forms;
namespace First_aid
{ public partial class Test : Form
    { private Timer timer = new Timer();
        string answer;//хранение правильного ответа
private Timer timer = new Timer();
string answer;//хранение правильного ответа
int question_count; //счетчик вопросов
int correcl_answers; //количество правильных ответов
int wrong_answers; //кол-во не правильных ответов
string[] array; //массив инф-и
int correcl_answers_num;//номер правильного ответа
int selected_response; //номер выбранного ответа
System.IO.StreamReader Read;//переменная для
считывания инф-и из файла
        public Test()
        { ApplicationForms.Test = this;

```

```

        InitializeComponent();
        timer.Interval = 2000; // Интервал срабатывания
таймера (1 секунда)
        timer.Tick += timer1_Tick; // Обработчик события
Tick }
private void Exit_Butt_Click(object sender, EventArgs
e)
    { this.Hide();
        ApplicationForms.Menu.Show(); }
void Start()
    { var encoding = System.Text.Encoding.UTF8;
try{ Read = new System.IO.StreamReader(@"t.txt",
encoding);
        this.Text = Read.ReadLine();//Считываем строку
и выводим в label
question_count = 0; correcl_answers = 0;
wrong_answers = 0; array = new string[20];
        } catch (Exception)
        { MessageBox.Show("error1"); }
        Question(); }
void Question()//смена вопроса
{ label1.Text = Read.ReadLine();
radioButton1.Text = Read.ReadLine();
radioButton2.Text = Read.ReadLine();
radioButton3.Text = Read.ReadLine();
correcl_answers_num = int.Parse(Read.ReadLine());
if (correcl_answers_num == 1)
    answer = radioButton1.Text;
if (correcl_answers_num == 2)
    answer = radioButton2.Text;
if (correcl_answers_num == 3)
    answer = radioButton3.Text;
radioButton1.Checked = false;
radioButton2.Checked = false;
radioButton3.Checked = false;
Next_butt.Enabled = false;
if (Read.EndOfStream == true) Next_butt.Text =
"завершить"; }
void Switching(object sender, EventArgs e)

```

```

{ Next_but.Enabled = true;
  Next_but.Focus();
  RadioButton Переключатель =
(RadioButton)sender;
  var tmp = Переключатель.Name;
  selected_response = int.Parse(tmp.Substring(11));}
private void timer1_Tick(object sender, EventArgs e)
{ timer.Stop();    // Останавливаем таймер
  label2.Visible = false;// Скрываем метку    }
private void Test_Load(object sender, EventArgs e) { }
private void pictureBox1_Click(object sender, EventArgs e)
{  MessageBox.Show("  Нажмите кнопку 'Начать
тест', \пзатем выберите вариант ответа и \пнажмите
кнопку 'Далее' "); }
private void pictureBox2_Click(object sender,
EventArgs e)
{ label2.Text = " ";
  Next_but.Text = "далее";
  radioButton1.Visible = true;
  radioButton2.Visible = true;
  radioButton3.Visible = true;
  Start();//начинаем заново тест    }
private void Start_but_Click(object sender, EventArgs
e)
{ Start_but.Enabled = false;
  Next_but.Enabled = true;
  radioButton1.CheckedChanged += new
EventHandler(Switching);
  radioButton2.CheckedChanged += new
EventHandler(Switching);
  radioButton3.CheckedChanged += new
EventHandler(Switching);
  Start(); }
private void Next_but_Click(object sender, EventArgs e)
{ label2.Visible = true; // Показываем метку сразу же
  timer.Start();    // Запускаем таймер
  question_count++;
  if (selected_response == correcr_answers_num)
  { correcr_answers += 1;
    label2.Text = "Верно"; }
  if (selected_response != correcr_answers_num)
  { label2.Text = $"Не верно, правильный ответ:\n
{answer}";
    wrong_answers += 1;
    array[wrong_answers] = label1.Text;
    //записываем в массив неправильные ответы    }
  if (Next_but.Text == "начать тестирование
сначала")
  { label2.Text = " ";
    Next_but.Text = "далее";
    radioButton1.Visible = true;
    radioButton2.Visible = true;
    radioButton3.Visible = true;
    Start();//начинаем заново тест
    return;//завершаем функцию }
  if (Next_but.Text == "завершить")
  { Read.Close();
    radioButton1.Visible = false;
    radioButton2.Visible = false;
    radioButton3.Visible = false;
    label1.Text = string.Format("Тестирование
завершено.\n" + "Правильных ответов {0} из {1}.\n" +
"Набранные баллы: {2:F2} ", correcr_answers,
question_count, (correcr_answers * 10 / question_count));
    var Str = "список ошибок:\n\n";
    for (int i = 1; i <= wrong_answers; i++)
    { Str = Str + array[i] + "\n"; }
    if (wrong_answers != 0)
    {  MessageBox.Show(Str, "Тестирование
завершено"); }
    Next_but.Text = "начать тестирование
сначала"; }
  if (Next_but.Text == "далее")
  { Question(); } } }
using System;
using System.Windows.Forms;
using WMPLib;
namespace First_aid

```

```

{ public partial class Options : Form
    { WindowsMediaPlayer player1 = new
WindowsMediaPlayer();
        WindowsMediaPlayer player2 = new
WindowsMediaPlayer();
        WindowsMediaPlayer player3 = new
WindowsMediaPlayer();
        WindowsMediaPlayer player4 = new
WindowsMediaPlayer();
        WindowsMediaPlayer player5 = new
WindowsMediaPlayer();
    public Options()
    { ApplicationForms.Options = this;
        InitializeComponent();
        //путь к .wav-файлу
        player1.URL = @"audio\audio1.wav";
        player2.URL = @"audio\audio2.wav";
        player3.URL = @"audio\audio3.wav";
        player4.URL = @"audio\audio4.wav";
        player5.URL = @"audio\audio4.wav";
        player1.settings.setMode("loop", true); // Включаем
бесконечное повторение
        player2.settings.setMode("loop", true);
        player3.settings.setMode("loop", true);
        player4.settings.setMode("loop", true);
        player1.controls.stop();
        player2.controls.stop();
        player3.controls.stop();
        player4.controls.stop(); }
    private void Exit_Butt_Click(object sender, EventArgs e)
    { this.Close();
        ApplicationForms.Menu.Show();}
    private void Audio1_Click(object sender, EventArgs e)
    { player1.controls.play();
        player2.controls.stop();
        player3.controls.stop();
        player4.controls.stop(); }
    private void Audio2_Click(object sender, EventArgs e)
    { player1.controls.stop();
        player2.controls.play();
        player3.controls.stop();
        player4.controls.stop(); }
    private void Audio3_Click(object sender, EventArgs e)
    { player1.controls.stop();
        player2.controls.stop();
        player3.controls.play();
        player4.controls.stop(); }
    private void Audio4_Click(object sender, EventArgs e)
    { player1.controls.stop();
        player2.controls.stop();
        player3.controls.stop();
        player4.controls.play(); }
    private void Stop_Audio_Click(object sender, EventArgs
e)
    { player1.controls.stop();
        player2.controls.stop();
        player3.controls.stop();
        player4.controls.stop(); } }
using System.Windows.Forms;
namespace First_aid
{ internal static class ApplicationForms
    { public static Menu Menu;
        public static Load Load;
        public static Training Training;
        public static Options Options;
        public static Test Test; } }
using System;
using System.Windows.Forms;
namespace First_aid
{ internal static class Program
    { /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        { Application.EnableVisualStyles();
            plication.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Load()); } }

```

Приложение Б
Тест-кейсы

Таблица Б.1 – Тестирование

| Идентификатор | Приоритет | Заглавие и шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
|---------------|----------------|---|---|---|----------|
| 1 | 2 | 3 | 4 | 5 | 6 |
| T_01 | крайне высокий | Запуск приложения 1. Запустить файл «Правила оказания первой помощи.exe» | 1. Отображается экран загрузки. Загружается главное меню игры с кнопками: «Обучение», «Опции», «Тест», «Выход» и «Помощь». | 1. Отобразился экран загрузки. Загрузилось главное меню игры с кнопками: «Обучение», «Опции», «Тест», «Выход» и «Помощь». | Пройдено |
| T_02 | Средний | Справка 1. Нажать на кнопку «Помощь» 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 «Опции». 7. Нажать на подраздел «Обучение». 8. Нажать на подраздел «Главное меню». 9. Нажать на подраздел «Тест». 10. Нажать на кнопку «Закрыть». | 1. Загружается окно справки с страницей и разделами: «Главная» с подразделами «О программе», раздел «Справочник» с подразделами «Главное меню», «Обучение», «Тест», «Опции». 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 «Опции» с основной информацией о настройке музыки. 7. Осуществляется переход на страницу «Обучение» с основной информацией о механике обучения. 8. Осуществляется переход на страницу «Главное меню» с информацией по главному экрану. 9. Осуществляется переход на страницу «Тест» с основной информацией о механике теста. 10. Закрывается справка. | 1. Загрузилось окно справки с начальной страницей и разделами: «Главная» с подразделами «О программе», раздел «Справочник» с подразделами «Главное меню», «Обучение», «Тест», «Опции». 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 «Опции» с основной информацией о настройке музыки. 7. Осуществился переход на страницу «Обучение» с основной информацией о механике обучения. 8. Осуществился переход на страницу «Главное меню» с информацией по главному экрану. 9. Осуществился переход на страницу «Тест» с основной информацией о механике теста. 10. Закрылась справка. | Пройдено |

Продолжение таблицы Б.1

| 1 | 2 | 3 | 4 | 5 | 6 |
|------|---------|--|--|---|----------|
| T_03 | высокий | Форма «Обучение» 1. Нажать на кнопку «Обучение» в главном меню. 2. На форме «Обучение» нажать на иконку «?». 3. Нажать на кнопку «Ок» на всплывающем сообщении. 4. На форме «Обучение» нажать на кнопку «!». 5. Нажать на элемент списка «Вывих» в левой части экрана на форме «Обучение». 6. На форме «Обучение» нажать на кнопку «Назад». | 1. Загружается форма «Обучение» с кнопками «Назад», «Помощь», «Внимание» и список травм и поражений. 2. Появляется всплывающее сообщение с правилами. 3. Всплывающее сообщение закрывается. 4. В правой части экрана появляется важная информация для ознакомления. 5. В правой части экрана появляется информация о вывихе, признаки и первая помощь. 6. Форма «Обучение» закрывается. Загружается главное меню игры. | 1. Загрузилась форма «Обучение» с кнопками «Назад», «Помощь», «Внимание» и список травм и поражений. 2. Появилось всплывающее сообщение с правилами. 3. Всплывающее сообщение закрывлось. 4. В правой части экрана появилась важная информация для ознакомления. 5. В правой части экрана появилась информация о вывихе, признаки и первая помощь. 6. Форма «Обучение» закрывлась. Загрузилось главное меню игры. | Пройдено |
| T_04 | высокий | Форма «Тест», прохождение теста 1. Нажать на кнопку «Тест» в главном меню 2. На форме «Тест» нажать на кнопку «Начать тест» 3. Выбрать верный вариант ответа 4. Нажать на кнопку «Далее» 5. Выбрать не верный вариант ответа 6. Нажать на кнопку «Далее» 7. На форме «Тест» нажать на кнопку «Помощь». 8. На форме «Тест» нажать на кнопку «Рестарт». 9. На форме «Тест» нажать на кнопку «Назад». | 1. Загружается форма «Тест» с кнопками «Начать тест», «Далее», «Назад», «Помощь», «Рестарт». 2. На форме появляется первый вопрос и варианты ответов, кнопка «Далее» не доступна. 3. Кнопка «Далее» становится доступна. 4. На форме появляется следующий вопрос и варианты ответов, появляется текст: «Верно», который исчезает через 2 секунды. Кнопка «Далее» становится не доступна. 5. Кнопка «Далее» становится доступна. 6. На форме появляется текст: «Не верно, правильный вариант ответа», который исчезает через 2 секунды, появляется следующий вопрос и варианты ответов. Кнопка «Далее» становится не доступна. 7. Появляется всплывающее сообщение с правилами. | 1. Загрузилась форма «Тест» с кнопками «Начать тест», «Далее», «Назад», «Помощь», «Рестарт». 2. На форме появился первый вопрос и варианты ответов, кнопка «Далее» не доступна. 3. Кнопка «Далее» стала доступна. 4. На форме появился следующий вопрос и варианты ответов, который появился текст: «Верно», который исчез через 2 секунды. Кнопка «Далее» стала не доступна. 5. Кнопка «Далее» стала доступна. 6. На форме появился текст: «Не верно, правильный вариант ответа: «Правильный вариант ответа», который исчез через 2 секунды, появился следующий вопрос и варианты ответов. Кнопка «Далее» стала не доступна. 7. Появилось всплывающее сообщение с правилами. | Пройдено |

Продолжение таблицы Б.1

| 1 | 2 | 3 | 4 | 5 | 6 |
|------|---------|--|---|--|----------|
| | | | 8. На форме появляется первый вопрос и варианты ответов, кнопка «Далее» не доступна. 9. Форма «Опции» закрывается. Загружается главное меню игры. | 8. На форме появился первый вопрос и варианты ответов, кнопка «Далее» не доступна. 9. Форма «Опции» закрывается. Загружается главное меню игры. | |
| T_05 | высокий | Форма «Тест», завершение теста • Ответ на последний вопрос 1. Выбрать вариант ответа 2. Нажать на кнопку «Завершить тестирование» | 1. Кнопка «Завершить тестирование» становится доступна. 2. Появляется информация о количестве правильных ответов и набранных баллов, а также всплывающее сообщение «Тестирование завершено» и список вопросов, в которых были ошибки. | 1. Кнопка «Завершить тестирование» стала доступна. 2. Появилась информация о количестве правильных ответов и набранных баллов, а также всплывающее сообщение «Тестирование завершено» и список вопросов, в которых были ошибки. | Пройдено |
| T_06 | средний | Форма «Опции» 1. Нажать на кнопку «Опции» в главном меню. 2. Нажать на кнопку №1. 3. Нажать на кнопку «Стоп». 4. На форме «Опции» нажать на кнопку «Выход». | 1. Загружается форма «Опции» с четырьмя кнопками воспроизведения музыки, кнопкой «Стоп» и «Назад». 2. Запускается аудио 1.wav. 3. Прекращается воспроизведение аудио 1.wav. 4. Форма «Опции» закрывается. Загружается главное меню игры. | 1. Загружается форма «Опции» с четырьмя кнопками воспроизведения музыки, кнопкой «Стоп» и «Назад». 2. Запустилось аудио №1. 3. Прекратилось воспроизведение аудио №1. 4. Форма «Опции» закрывлась. Загружается главное меню игры. | Пройдено |
| T_07 | средний | Кнопка «Выход» 1. Нажать на кнопку «Выход» в главном меню. | 1. Приложение закрывается. | 1. Приложение закрывлось. | Пройдено |