|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  Факультет среднего профессионального образования | |
| **Курсовая работа**  по междисциплинарному курсу «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей»    **Разработка программного обеспечения для тренировки памяти**  Пояснительная записка  ОГУ 09.02.07. 3024. 853 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 22ИСП-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Козловская  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

|  |  |
| --- | --- |
| Утверждаю  председатель ПЦК дисциплин профессионального цикла | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись | Ж.В. Михайличенко |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту \_\_\_\_\_Козловская Екатерина Александровна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

по специальности \_\_\_09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по междисциплинарному курсу \_\_Технология разработки программного обеспечения\_\_\_\_\_\_\_\_\_\_

1. Тема работы: \_\_\_Разработка программного обеспечения для тренировки памяти\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_июня\_\_\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Программно смоделировать программное обеспечение, способное эффективно тренировать память пользователя и повышать его когнитивные способности.\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и теории игр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Перечень вопросов, подлежащих разработке: \_а) изучить предметную область, выполнить анализ требований к программному обеспечению, составить техническое задание на разработку; б) выполнить проектирование системы с помощью CASE-средств; в) для решения поставленной задачи реализовать оконное приложение на языке C# и протестировать его; г) сформулировать предложения по внедрению, эксплуатации и сопровождению разработанного программного обеспечения. Сделать выводы по результатам проделанной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Перечень графического (иллюстративного) материала: таблицы, графики, рисунки, схемы, отражающие теоретический материал и программную реализацию поставленной задачи\_\_\_\_

Дата выдачи и получения задания

Руководитель «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ж.В. Михайличенко\_\_\_

подпись инициалы, фамилия

Студент «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Е.А. Козловская\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 853 ПЗ

Разраб.

Козловская Е.А.

Провер.

Михайличенко Ж

Реценз.

Н. Контр.

Утверд.

Разработка программного обеспечения для тренировки памяти

Лит.

Листов

\*

22ИСП-1

В курсовой работе по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществление интеграции программных модулей» проведена разработка компьютерной логической игры «Запомни!»

В первой главе курсовой работы были описаны анализ требований и проектирование программного продукта.

Во второй главе курсовой работе были описаны реализация и тестирование программного продукта.

В третьей главе курсовой работы были описаны руководство пользователя, рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения.

Пояснительная записка содержит 28 страницы, в том числе 7 рисунков, 3 таблицы, 9 источников, 1 приложение.

Разработка приложения выполнена в среде разработки Visual Studio 2022 на программном языке С#.

**Содержание**

[Введение 5](#_Toc169012162)

[1 Анализ требований и проектирование программного продукта 6](#_Toc169012163)

[1.1 Анализ предметной области 6](#_Toc169012164)

[1.2 Техническое задание на разработку 7](#_Toc169012165)

[1.3 Построение модели 10](#_Toc169012166)

[2 Реализация и тестирование программного продукта 13](#_Toc169012167)

[2.1 Обоснование программных средств реализации 13](#_Toc169012168)

[2.2 Разработка пользовательского интерфейса 14](#_Toc169012169)

[2.3 Алгоритмизация и программирование 16](#_Toc169012170)

[2.4 Тестирование 19](#_Toc169012171)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения 21](#_Toc169012172)

[3.1 Руководство пользователя 21](#_Toc169012173)

[3.2 План внедрения и сопровождения 24](#_Toc169012174)

[Заключение 26](#_Toc169012175)

[Список использованных источников 27](#_Toc169012176)

[Приложение А 28](#_Toc169012177)

# Введение

В современном мире, многие люди страдают от плохой памяти, все более важными становятся такие человеческие качества, как хорошая память. В сети Интернет, существуют игры направленные на тренировку памяти, они могут стать эффективным инструментом для саморазвития и поддержания хорошей памяти в тонусе.

Игра «Запомни!» как раз относится к этой категории. Она проста в освоении, но при этом требует от игрока сосредоточенности, и навыки быстрого зрительного запоминания.

Цель данной курсовой работы заключается в разработке программно смоделированной компьютерной логической игры «Запомни!», предназначенной для тренировки памяти пользователя.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Изучение темы тренировки памяти и её методики.
* Изучение темы разработки логических игр и создание концепции компьютерной логической игры «Запомни!».
* Реализация механики появления «карточек» – клеток разного цвета в отдельных клетках-окошках сразу после начала игры.
* Разработка логики взаимодействия пользователя с компьютерной логической игрой «Запомни»: возможность «щелкнуть» по кнопкам выбора цвета и карточке мышкой для её окрашивания в выбранный цвет.
* Реализация возможности выбора количества карточек и времени на их запоминание.

Данная курсовая работа имеет практическую значимость, поскольку разработанный программный продукт будет полезен для тех, кто желает проверить работоспособность своей памяти. Благодаря отсутствию необходимости в специфических знаниях компьютера, игру «Запомни!» сможет оценить любой желающий.

Таким образом, разработка компьютерной логической игры «Запомни!» имеет не только развлекательное, но и образовательное значение, способствуя развитию навыков программирования и предоставляя возможность проверить свои способности в увлекательной и динамичной игровой форме.

# Анализ требований и проектирование программного продукта

## Анализ предметной области

Тренировка памяти – это неотъемлемая часть нашей повседневной жизни. Наш мозг постоянно сталкивается с информационным потоком, и способность к запоминанию и воспроизведению информации играет ключевую роль в нашем успехе. Умение запоминать и вспоминать важные детали, события и факты помогает нам принимать обоснованные решения, а также эффективно взаимодействовать с окружающим миром. Тренировка памяти не только помогает нам лучше усваивать информацию, но и повышает нашу способность принимать обоснованные решения в различных ситуациях.

Есть ситуации, в которых нужно запомнить очень много информации. Например, нужно выучить к следующему дню параграф по истории на 15 страниц, и для человека с плохо развитой памятью запомнить такое количество информации будет очень сложно.

Для успешного запоминания информации необходимо обладать определенным набором навыков и стратегий:

1) Навык внимания и концентрации;

2) Навык ассоциации;

3) Навык повторения и повторного воспроизведения;

4) Навык организации информации.

Забывчивость – это склонность к забыванию, невнимательность к деталям, частые забывания информации или событий. Это может быть связано как с физиологическими особенностями мозга, так и с психологическими факторами, например, стрессом, усталостью или недостатком внимания.

Причины забывчивости:

1) Стресс и эмоциональное напряжение. Временные переживания стресса могут влиять на способность мозга к концентрации и запоминанию информации.

2) Недостаток сна. Недостаток сна может привести к ухудшению памяти и концентрации.

3) Отвлечения, многозадачность и недостаток концентрации могут привести к забывчивости.

4) Физиологические изменения. С возрастом могут происходить изменения в мозге, которые влияют на память и способность запоминания информации.

5) Недостаток упражнений для мозга: Недостаток умственной активности и тренировок для мозга может привести к ухудшению памяти.

6) Перегруженность информацией. Избыток информации и постоянные потоки данных могут затруднять процесс запоминания и вызывать забывчивость.

Существует несколько методик для тренировки памяти:

а) Метод ассоциаций основан на принципе ассоциативного мышления. Для запоминания информации нужно связать ее с чем-то уже известным или создать яркие образы в уме. Например, чтобы запомнить список покупок, можно представить каждый предмет в виде картинки и связать его с каким-то знакомым местом или событием.

б) Метод локусов заключается в том, чтобы связывать информацию с конкретными местами или объектами в вашем окружении. Например, чтобы запомнить список дел на день, вы можете представить каждое дело в определенной комнате вашего дома и при прохождении по комнатам легко вспомнить все задачи.

в) Метод фраз заключается в создании ассоциативных фраз или историй, в которых каждое слово или элемент помогает запомнить определенную информацию. Например, для запоминания порядка цветов радуги можно использовать фразу «Каждый охотник желает знать, где сидит фазан».

г) Метод повторения это один из самых простых способов тренировки памяти – это повторение информации. Повторение помогает закрепить знания в памяти и улучшает их сохранение на долгосрочной основе.

д) Метод игры в головоломки и кроссворды. Решение головоломок, кроссвордов и других умственных задач способствует развитию памяти, логического мышления и внимания.

е) Метод медитации и концентрации. Практика медитации и концентрации помогает улучшить способность к сосредоточению и управлению своими мыслями, что также положительно сказывается на памяти.

## Техническое задание на разработку

Техническое задание на разработку программного обеспечения – документ, который содержит подробное описание требований программному продукту. Техническое задание на разработку программного обеспечения для тренировки памяти составлено согласно ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» и представлена ниже.

1. Общие сведения.

а) Наименования автоматизированной системы (АС): «Запомни!»

б) Наименование заказчика: факультет среднего профессионального образования Орского гуманитарно-технологического института (филиала) ОГУ в лице преподавателя высшей категории Михайличенко Ж.В.;

в) Наименование разработчика: студентка второго курса группы 22ИСП-1 Козловская Е.А.;

г) Документ, на основании которого создаётся АС: протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 года;

д) Дата начала работ: 19.02.2024

е) Дата окончания работ: 10.06.2024

1. Цели и назначения создания АС:

а) Цели создания АС: разработать программное обеспечение, которое будет проверять когнитивных функций, включая память, внимание, скорость мышления и другие аспекты когнитивной деятельности.

б) Назначение АС: предназначена для использования людьми, желающими улучшить свою память и когнитивные способности в целом, кто страдает от проблем с памятью, например, из-за старения или нейрологических расстройств.

1. Характеристика объекта автоматизации.

Объектом автоматизации разрабатываемой программной системы является процесс запоминания.

Для запоминания порядка карточек могут использоваться различные стратегии, такие как визуализация, ассоциации, метод места, повторение. Каждая из этих стратегий помогает улучшить навыки запоминания и восстановления последовательности карточек.

1. Требование к АС:

а) Требования к функциям выполнения АС:

- Выбор количества карточек;

- Выбор ограничения по времени на запоминание;

- Окрашивание карточек;

- Подсчёт очков за правильно пройденный уровень;

- Вывод результата игры на экран;

- Сохранение результата в файл.

б) Требование к видам обеспечения АС:

- Требования к математическому обеспечению: в программном обеспечении используется проверка правильно пройденного уровня и подсчёт очков;

- Требование к информационному обеспечению: данные об игровом процессе хранятся локально на устройстве пользователя, без использования СУБД. Обновление данных происходит в реальном времени в зависимости от действий пользователя;

- Требование к лингвистическому обеспечению: интерфейс системы должен быть русскоязычным, общение с пользователем реализуется с помощью диалоговых окон;

- Требование к программному обеспечению: система должна быть совместима с операционной системой Windows, язык программирования C#, интегрированная среда разработки – Microsoft Visual Studio 2022, Ramus Educational для построения функциональной модели, Microsoft Word для документирования;

- Требования к техническому обеспечению: система не должна предъявлять высоких требований к аппаратным средствам и могла бы исполняться на любом современном персональном компьютере (ПК);

- Требования к организационному обеспечению: взаимодействие с системой будет осуществляться пользователем, который может вводить исходные данные и получать результаты путём вывода на экран монитора или записи в файл;

в) Общие технические требования к АС:

- Требования к численности и квалификации пользователей: для работы программы достаточно одного пользователя со знанием методов принятия решений в условиях неопределенности;

- Требования к эргономике и технической эстетике: программное обеспечение должно иметь интуитивно понятный пользовательский интерфейс, позволяющий легко освоить управление и правила игры;

- Требования к защите информации: программное обеспечение не содержит конфиденциальной информации;

1. Состав и содержание работ по созданию АС.

В таблице 1 показаны этапы разработки АС «Запомни!»

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 | 2 | 3 |
| 1 Анализ требований | 19.02.2024 – 05.03.2024 | Анализ предметной области, изучение программных аналогов, разработка технического задания на создание АС |
| 2 Проектирование | 05.03.2024 – 01.04.2024 | Разработка диаграмм IDEF различного уровня, диаграмм вариантов использования. Выбор и обоснование технологий и инструментов разработки. Проектирование пользовательского интерфейса. |
| 3 Программирование | 01.04.2024 – 01.06.2024 | Разработка и программная реализация алгоритмов и методов ввода, хранения, обработки и вывода данных. Разработка пользовательского интерфейса, организация диалога с пользователем. Отладка работы системы. Документирование программного кода. |
| 4 Тестирование | 01.06.2024 – 10.06.2024 | Проведение функционального, модульного, интеграционного и системного тестирования. Исправление ошибок и несоответствий. |
| 5 Внедрение | 10.06.2024 – 12.06.2024 | Установка и настройка системы. Подготовка сопровождающей документации (пояснительной записки к курсовой работе, руководства пользователя и других.) |
| 6 Эксплуатация и сопровождение | 12.06.2024 – 12.06.2025 | Разработка рекомендаций по эксплуатации и сопровождению системы. |

1. Порядок разработки АС.

Этапы будут выполняться разработчиком в прямой последовательности в соответствии с приведенной в пункте пять таблицей этапов разработки АС с обязательным контролем и консультацией с заказчиком.

1. Порядок контроля и приёмки АС.

Контроль над разработкой осуществляется заказчиком на каждом этапе разработке. Приёмка должна быть организованна на защите курсовой работы.

1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие.

Программа может быть установлена методом копирования архива, содержащего папку проекта. Запустить программу можно через файл с расширением \*.exe или через ярлык этого файла. Обучение пользователя производится просьбой прочитать документацию, приложенную к программе в архиве.

1. Требования к документированию.

– Документирование программного кода (комментарии);

– Пояснительная записка к курсовой работе;

– Руководство пользователя.

1. Источники разработки.

* Протокол закрепления тем курсовой работы по дисциплине «технология разработки программного обеспечения» от 19.02.2024;
* ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;
* Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.
* ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».
* ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».

## Построение модели

Проектирование АС является важным этапом в создании эффективных и оптимизированных систем. Представление АС в виде функциональных моделей помогает лучше понять структуру и взаимодействие компонентов системы, что облегчает их дальнейшую разработку и улучшение.

IDEF0 – это методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Преимуществом этой методологии является возможность создать чёткую и структурированную модель, которая легко воспринимается и понимается, а также обеспечивает высокую точность моделирования.

Программа Ramus Educational относится к CASE-средствам (Computer Aided System/Software Engineering) и является компьютерным инструментом для анализа, моделирования и разработки систем. Ramus Educational предназначено для использования в проектах, в которых необходимо описание процессов. С его помощью можно создавать визуальные диаграммы, используемые для наглядного отображения различных процессов.

Для создания функциональной модели IDEF0 разрабатываемого программного обеспечения необходимо, прежде всего, построить контекстную диаграмму, состоящую из единственного функционального блока А0 «Функционирование компьютерной логической игры «Запомни!»».

В качестве входных компонентов выявлены начальное количество карточек и ограничение на время запоминания. Правила игры были установлены в качестве управляющего элемента. В качестве механизмов выявлены пользователь и персональный компьютер. Выходными компонентами являются сообщение о проигрыше, сообщение о выигрыше и количестве очков.

Контекстная диаграмма EDEF0 показана на рисунке 1.

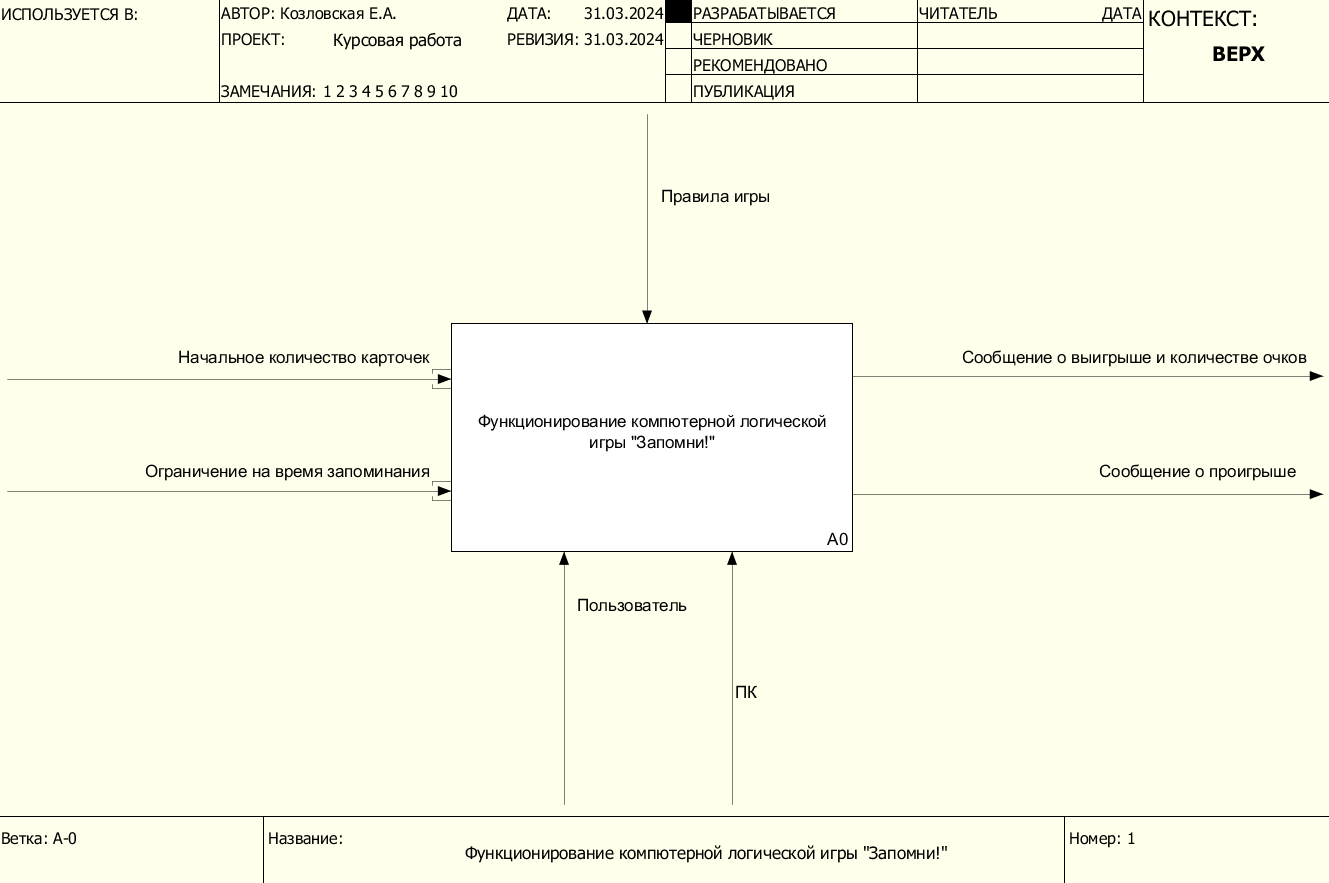


Рисунок 1 – Контекстная диаграмма EDEF0

Для того чтобы лучше понять функции системы необходимо провести декомпозицию, определив следующие функции:

A1 – Настройка сложности уровня. В качестве входного компонента выявлено желание продолжить игру. Правила игры установлены в качестве управляющего элемента. В качестве механизмов выявлены: пользователь, персональный компьютер. Выходными компонентами являются количество карточек, ограничение на время запоминания.

A2 – Реализация игры. В качестве входных компонентов выявлены количество карточек, ограничение на время запоминания, необходимость продолжить игру. Правила игры установлены в качестве управляющего элемента. В качестве механизмов выявлены пользователь и персональный компьютер. Выходными компонентами является выполненное задание.

A3 – Проверка результатов игры. В качестве входных компонентов выявлено выполненное задание. Правила игры установлены в качестве управляющего элемента. В качестве механизма выявлен персональный компьютер. Выходными компонентами являются сообщение о проигрыше, сообщение о выигрыше и количестве очков, необходимость продолжить игру, желание начать следующую игру.

Диаграмма декомпозиции первого уровня EDEF0 показана на рисунке 2.

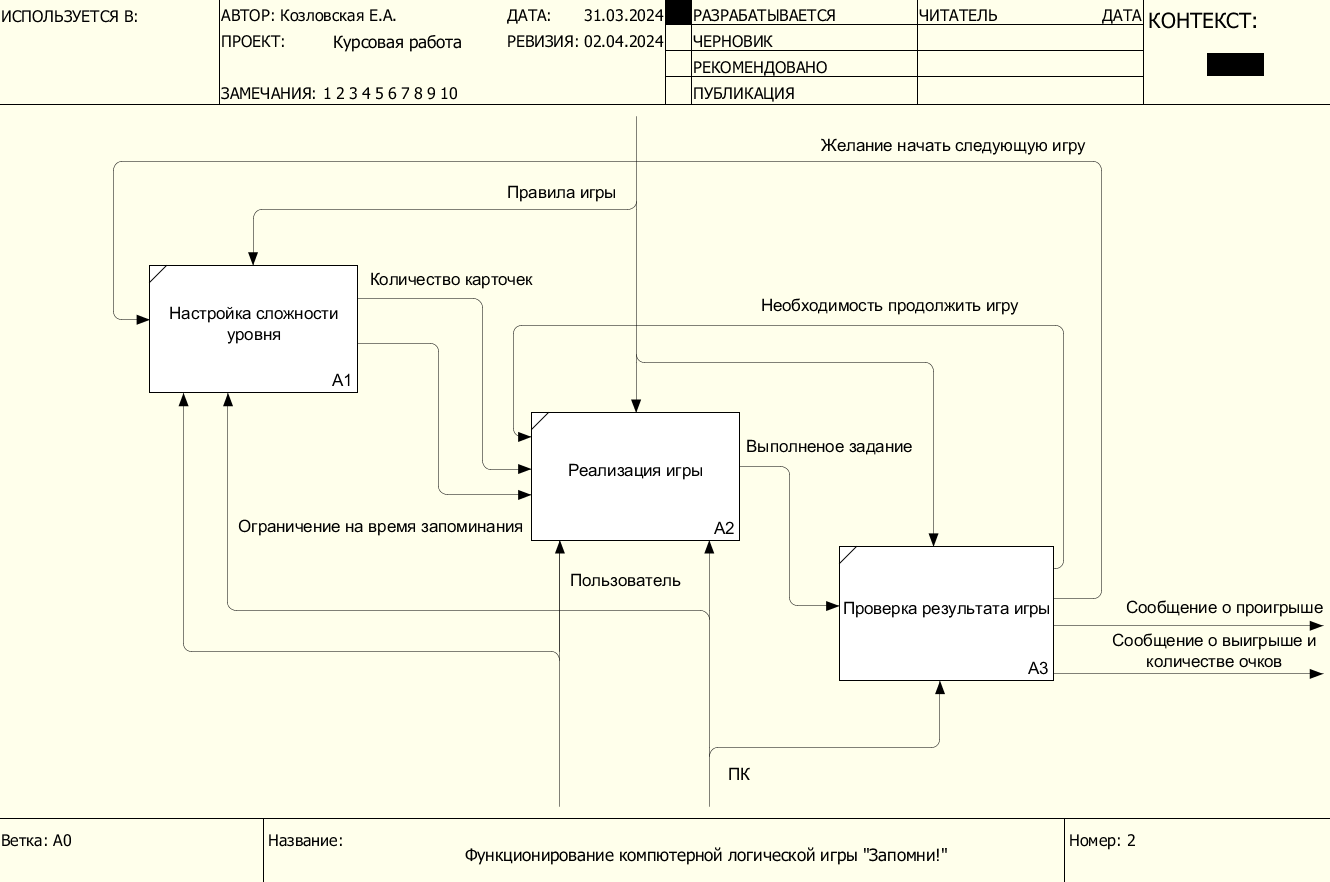


Рисунок 2 – Диаграмма декомпозиции первого уровня EDEF0

Все построенные функциональные диаграммы будут использованы при разработке программного обеспечения для тренировки памяти.

# Реализация и тестирование программного продукта

## Обоснование программных средств реализации

Выбор языка программирования C# и интегрированной среды разработки Visual Studio 2022 для реализации компьютерной логической игры «Запомни!» обоснован рядом ключевых преимуществ и возможностей, которые они предоставляют.

Язык программирования C# является одним из самых популярных и широко используемых языков в индустрии разработки программного обеспечения. Его объектно-ориентированный подход позволяет эффективно структурировать код, повышая его читаемость и управляемость. Благодаря строгой типизации и системе проверки типов на этапе компиляции, C# помогает предотвращать множество ошибок и повышает надежность разрабатываемых приложений. Он широко применяется в различных областях разработки программного обеспечения, включая веб-разработку, мобильные приложения, игровую индустрию, робототехнику и многое другое. Его синтаксис понятен и легко читаем, что делает процесс разработки более эффективным и продуктивным. C# имеет обширную стандартную библиотеку классов .NET, которая предоставляет множество готовых решений для решения широкого спектра задач.

Интегрированная среда разработки Visual Studio предоставляет полный набор инструментов для создания приложений любой сложности. Интегрированная среда разработки обеспечивает удобный интерфейс, поддержку множества языков программирования, инструменты для отладки, профилирования и тестирования кода, а также возможности для коллективной работы над проектами. Богатый функционал Visual Studio позволяет повысить производительность и качество своей работы.

Visual Studio, в свою очередь, предлагает удобную среду для создания и отладки приложений. С помощью Visual Studio можно легко создавать графические интерфейсы, выполнять автоматическое тестирование кода.

Важным преимуществом Visual Studio является его возможность создания кроссплатформенных приложений. Это позволяет писать код один раз и запускать его на различных платформах, включая iOS, Android и Windows.

Visual Studio обладает мощными инструментами для анализа кода, автоматического исправления ошибок и профилирования производительности приложений. Это помогает ускорить процесс разработки, повысить качество кода и оптимизировать работу приложений.

В целом, выбор языка программирования C# и интегрированной среды разработки Visual Studio обусловлен их мощными возможностями, удобством использования, высокой производительностью и широкой поддержкой со стороны сообщества. Эти инструменты помогут успешно реализовать компьютерную логическую игру «Запомни!» с типом проекта Windows Forms на платформе .NET Framework, обеспечивая ее эффективность, надёжность и высокое качество.

## Разработка пользовательского интерфейса

Программный интерфейс (API) – это набор определённых правил и спецификаций, которые определяют, как различные компоненты программного обеспечения могут взаимодействовать друг с другом. Он обеспечивает структурированный способ для приложений и сервисов обмениваться данными и функциональностью. Он определяет доступные методы, параметры и форматы данных, которые могут быть использованы для взаимодействия между различными программами.

Особенности программного интерфейса:

* Интерфейс описывает доступные функции и методы, а также формат и структуру передаваемых данных.
* Позволяет разработчикам использовать существующий функционал без необходимости знания всех деталей его реализации.
* Обеспечивает удобство и простоту взаимодействия различных программ и компонентов друг с другом.
* Может быть предоставлен как в виде библиотеки (набор функций), так и через удалённый доступ по сети (например, веб-сервисы).

Требования к программному интерфейсу:

1. Документирование – чёткое описание методов, параметров и форматов данных.

2. Надёжность – обеспечение стабильной работы интерфейса даже при нагрузке и различных условиях использования.

3. Простота использования – удобство работы с интерфейсом для разработчиков, минимум сложностей.

4. Эффективность – оптимальное использование ресурсов и быстрая обработка запросов.

5. Совместимость – возможность взаимодействия с разными операционными системами, языками программирования и аппаратными платформами.

Программный интерфейс компьютерной логической игры «Запомни!» должен соответствовать всем выше перечисленным требованиям. Структура интерфейса и его компоненты показаны на рисунке 3.



Рисунок 3 – Структура интерфейса и его компоненты

Компоненты интерфейса логической компьютерной игры «Запомни!»:

button1 «Начать игру»: при нажатии этой кнопки запускается таймер, генерируются карточки на поле DataGridView.

button2 «Сброс»: при нажатии этой кнопки все цветные карточки в поле DataGridView удаляется, а таймер обнуляется.

button3 «Подтвердить»: при нажатии этой кнопки происходит проверка правильности закрашивания карточек. Если все карточки закрашены верно, то выводится MessageBox с надписью о выигрыше. В ином случае выводится MessageBox с надписью о проигрыше.

DataGridView1: поле генерации цветных карточек.

numericUpDown1: выбор количества карточек.

numericUpDown2: выбор времени на запоминание.

textBox1: здесь описаны правила логической компьютерной игры «Запомни!».

radioButton1 «Красный»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в красный цвет.

radioButton2 «Оранжевый»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в оранжевый цвет.

radioButton3 «Жёлтый»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в жёлтый цвет.

radioButton4 «Зелёный»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в зелёный цвет.

radioButton5 «Синий»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в синий цвет.

radioButton6 «Фиолетовый»: после выбора этой кнопки, нажимая на карточку поля DataGridView1, карточка окрашивается в фиолетовый цвет.

label1: «Количество карточек».

label2: «Время запоминания».

label3: выводится обратный отсчёт таймера, а также сообщение о закончившемся времени и начале закрашивания карточек.

label4: «Правила игры».

timer1: запускается при нажатии кнопки «Начать игру», оставшееся время отображается в label3.

## Алгоритмизация и программирование

Алгоритм — это чётко определённая последовательность действий или инструкций, предназначенная для решения определённой задачи или класса задач.

Основные свойства алгоритма

1. Определённость. Каждый шаг алгоритма должен быть чётко определён, без каких-либо двусмысленностей.
2. Конечность. Алгоритм должен завершиться после конечного числа шагов. Это не означает, что алгоритм короткий, но гарантирует, что он не будет выполняться вечно.
3. Массовость. Алгоритм должен быть применим к широкому классу задач, а не только к одной конкретной задаче.
4. Результативность. После завершения алгоритма нужно получить результат, который соответствует цели, для которой алгоритм был разработан.
5. Эффективность. Хотя это свойство не является обязательным, хорошие алгоритмы обычно оптимизированы таким образом, чтобы они были как можно более эффективными в плане использования ресурсов и времени выполнения.

Алгоритмы удобно представлять в виде блок-схем согласно ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».

Для компьютерной логической игры «Запомни!» блок-схема алгоритма представлена на рисунках 4, 5 и 6.

Start

1

Рисунок 4 – Блок-схема алгоритма компьютерной логической игры «Запомни!»

1

Запуск таймера

Создание строк с карточками

Создание столбцов с карточками

Закрашивание карточки цветом

Выбор количества карточек, время запоминания

Таймер закончился

Закрашивание карточки белым

2

+

-

Рисунок 5 – Продолжение блок-схема алгоритма компьютерной логической игры «Запомни!»

+

-

2

Проверка строк

Проверка столбцов

Закрашивание карточек по выбору пользователя

Совпадение с изначальным цветом

Вывод сообщения о проигрыше

3

Вывод сообщения о выигрыше

End

3

Рисунок 6 – Продолжение блок-схема алгоритма компьютерной логической игры «Запомни!»

Для реализации функций логической компьютерной игры «Запомни!» разработаны следующие методы:

1. private List<Color> – лист цветов для рандомного закрашивания карточек;
2. private void button1\_Click(object sender, EventArgs e) – данный метод начинает игру, а именно создаёт выбранное количество цветных карточек, запускает таймер и по его окончанию закрашивает все карточки в белый.
3. private void dataGridView1\_CellClick (object sender, DataGridViewCellEventArgs e) – закрашивание белых клеток в цвета при помощи выбора radioButton.
4. private void button2\_Click(object sender, EventArgs e) – данный метод отчищает поле dataGridView.
5. private void button3\_Click(object sender, EventArgs e) – данный метод осуществляет проверку правильно закрашенных клеток и выводит MessageBox с надписью о выигрыше, если все клетки закрашены в нужный цвет, или с надписью о проигрыше, если хотя бы одна клетка окрашена не в тот цвет.
6. private void Timer\_Tick(object sender, EventArgs e) – данный метод выполняет работу таймера, а также запоминает изначальный цвет клеток до окрашивания в белый.

## Тестирование

Тестирование АС – процесс проверки и оценки качества программного обеспечения, который позволяет выявить ошибки и дефекты в работе системы.

Целями тестирования АС являются:

* Выявление дефектов.
* Подтверждение соответствия требованиям.
* Проверка производительности.
* Улучшение качества.

Виды тестирования, которые будут использоваться при приёмке АС «Запомни!»:

1. Функциональное тестирование – должно показать, что информационная система предоставляет пользователям ту функциональность, которая от неё ожидается, то есть система выполняет все свои функции корректно.
2. Компонентное тестирование – испытание отдельных программных компонентов (модулей, подсистем) АС, в ходе которых подтверждается корректность проведённых этими компонентами вычислений.
3. Интеграционное тестирование – испытание, направленное на выяснение проблем взаимодействия отдельных компонентов (программных и технических).
4. Нагрузочное тестирование – испытание АС в условиях нормальной нагрузки (количество запросов, которое система должна обработать за указанное количество времени).
5. Тестирование эргономики – испытание UI на предмет удобства и безопасности эксплуатации АС.
6. Тестирование надёжности – испытание работоспособности АС и возможности возвращаться к нормальному режиму работы после максимальной нагрузки, стрессов, предельных объёмов данных.

Результаты приёмки информационной системы для тренировки памяти «Запомни!» представлены в таблице 2.

Таблица 2 – Результаты приёмки

|  |  |
| --- | --- |
| Тип тестирования | Оценка по десятибалльной шкале |
| Функциональное тестирование |  |
| Компонентное (модульное) тестирование |  |
| Интеграционное тестирование |  |
| Нагрузочное тестирование |  |
| Тестирование эргономических решений (интерфейса) |  |
| Тестирование надёжности |  |

# Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения

## Руководство пользователя

1. Системные требования

Для стабильной и эффективной работы «Запомни!» рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): 2.4 GHz

Количество ядер процессора (CPU): 2

Объём оперативной памяти (RAM): 4 GB

Объём свободного места на диске (HDD): 20 GB

Операционная система (OS): Windows 10

Среда программирования: Visual Studio 2022

Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge

1. Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с «Запомни!».

Перед началом работы, пожалуйста, ознакомьтесь с [системными требованиями](file:///C:\Users\User\Downloads\Telegram%20Desktop\Курсовая%20работа%20Балакин%202024%20(2).docx#a9eb003e-8704-4d8f-8d92-8f59944ba6d6) и [лицензионным соглашением](file:///C:\Users\User\Downloads\Telegram%20Desktop\Курсовая%20работа%20Балакин%202024%20(2).docx#0343637b-d224-4986-a470-e45222a3137f).

1. Основные понятия и термины

Перед началом работы с компьютерной логической игрой «Запомни!» рекомендуем ознакомиться с основными понятиями и терминами:

Visual Studio - интегрированная среда разработки Visual Studio является творческой стартовой площадкой, которую можно использовать для редактирования, отладки и сборки кода, а также для публикации приложения. В дополнение к стандартному редактору и отладчику, предоставляемых большинством интегрированных сред разработки, Visual Studio включает компиляторы, средства завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки программного обеспечения.

C# — это язык программирования, разработанный компанией Microsoft. Он является универсальным, объектно-ориентированным языком программирования, который широко используется для разработки различных приложений, включая веб-приложения, настольные приложения и игры. C# является частью платформы .NET и часто используется в сочетании с другими технологиями, такими как ASP.NET для веб-разработки и Unity для разработки игр.

Windows 11 — это операционная система Windows NT, разработанная компанией Microsoft.

Память - обозначение комплекса познавательных способностей и высших психических функций, относящихся к накоплению, сохранению и воспроизведению знаний, умений и навыков.

Windows Forms в Visual Studio — это инструмент для создания графического интерфейса пользователя (GUI) в приложениях на платформе .NET. С помощью Windows Forms можно создавать окна, элементы управления (например, текстовые поля, кнопки и т. д.) и связывать их события с обработчиками кода. Windows Forms предоставляет удобный способ создания интерактивных приложений с помощью методов перетаскивания и размещения элементов на форме.

1. Установка

Для установки «Запомни!», пожалуйста, загрузите интегрированную среду программирования Visual Studio 2022.

В процессе установки, пожалуйста, разархивируйте архив, установить необходимые библиотеки.

Разархивируйте файл с программой и запустите файл с расширением .sln с помощью MS Visual Studio 2022.

1. Запуск

Для запуска «Запомни!» нажмите на ярлык программы «Remember!.sln» в папке с игрой. Откроется интегрированная среда разработки Visual Studio 2022, после этого нажмите на кнопку «Пуск».

1. Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса «Запомни!» 1.0: основных режимов работы, предназначение окон и экранов, доступные операции.

1. Главное окно программы

Главное окно программы «Запомни!» позволяет выполнять следующие операции:

* Выбор количества карточек
* Выбор ограничения на запоминание карточек

Главное окно программы показано на рисунке 6.

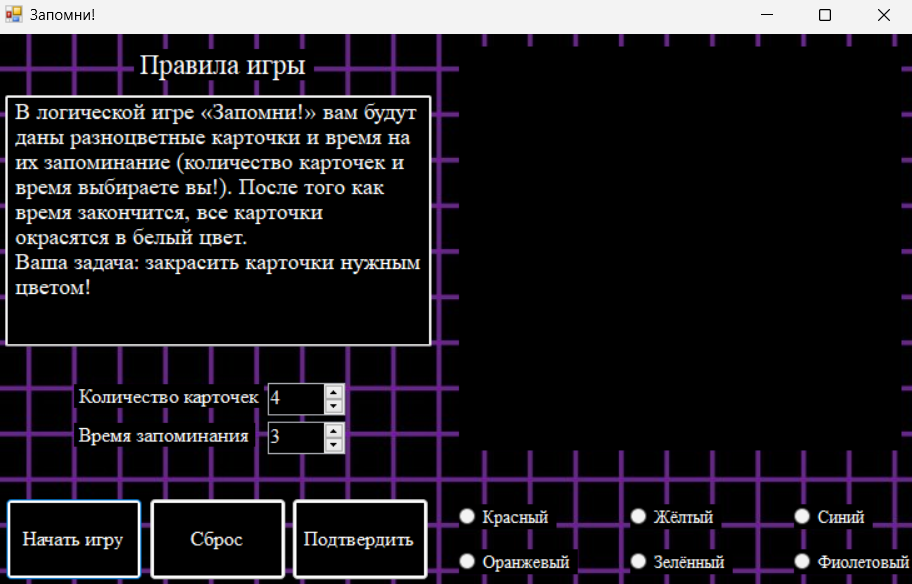


Рисунок 6 - Главное окно программы

1. Справка

Окно Справка позволяет узнать правила компьютерной игры «Запомни!».

Справка с правилами игры показана на рисунке 7.

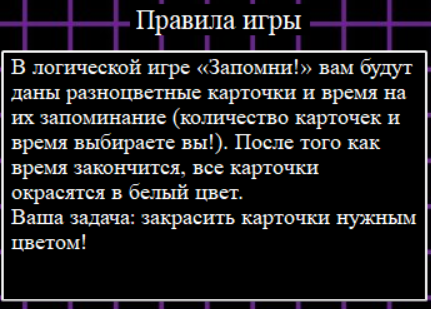


Рисунок 7 – Справка с правилами игры

Окно «Справка» находится на интерфейсе главной формы и его всегда видно.

1. Режимы работы

Пользовательский интерфейс «Запомни!» обеспечивает работу только в режиме пользователя и предоставляет полный функционал пользователю программы.

1. Устранение типовых проблем

Описание проблемы: Зависание программы.

Решение: Перезапуск программы или ПК

1. Частые вопросы (FAQ)

Для чего нужна ваша игра?

Ответ: Наша игра нужна для тренировки памяти.

Как пользоваться данной программой?

Ответ: Запускаете Visual Studio, запускаете программу в режиме выполнения, нажимаете кнопку «Начать игру».

1. Контактная информация

«Запомни!» разрабатывается и поддерживается компанией SkeltiK, являющейся правообладателем.

Вы можете направить вопросы по функциональности программы «Запомни!» следующими способами:

* Email: kozlovskayaea@og-ti.ru
* Телефон: +7 9198639733

1. Продажи и сотрудничество

По вопросам лицензирования и сотрудничества, пожалуйста, используйте следующие каналы:

* Email: kozlovskayaea@og-ti.ru
* Телефон: +7 9198639733

Офис компании и адрес для корреспонденции

пр. Мира 15

Г. Орска

Оренбургской обл., РФ

## План внедрения и сопровождения

Внедрение – это сложный процесс интеграции программного продукта в финансово-управленческую деятельность компании, требующий от специалистов, участвующих в проекте, наличия специальных знаний, соответствующего опыта и особого подхода к реализации проекта.

Сопровождение АС – процесс улучшения, оптимизации и устранения дефектов после передачи АС в эксплуатацию.

Сопровождение информационной системы включает в себя:

* управление (создание/удаление/настройка доступа) учётными записями всех пользователей, поддержание правил разграничения доступа;
* установка обновлений программного обеспечения;
* изменение эксплуатационной документации и организационно-распорядительных документов по защите информации;
* корректировки базовой конфигурации информационной системы и системы её защиты;
* периодический контроль уровня защищенности информации;

Сопровождение информационной системы позволяет максимально оперативно решать вопросы, которые могут возникнуть во время её использования, при изменении самой системы (частично или полностью). Высококвалифицированные специалисты нашей компании помогут вам в решении этой задачи.

В таблице 3 представлен план внедрения и сопровождения.

Таблица 3 – План внедрения и сопровождения

|  |  |
| --- | --- |
| Мероприятия | Сроки (дни) |
| 1 | 2 |
| Первоначальный запуск на компьютерах пользователей | 1 |

Таблица 4 – продолжение таблицы 3

|  |  |
| --- | --- |
| 1 | 2 |
| Исправление ошибок в результате эксплуатации | 30 |
| Дальнейшая доработка программы | 365 |
|  | 1 |

# Заключение

Во время выполнения курсовой работы по дисц. были получены знания о проектировании и создании компьютерной логической игры в Windows Forms на языке C#, её тестировании, внедрении и сопровождении.

Были получены многочисленные знания о методах тренировки памяти. Были изучены такие методы как: метод ассоциаций, метод, метод фраз, метод повторения, метод игры в головоломки и кроссворды, метод медитации и концентрации. Также была изучена разница между ними и их принципами работы.

При проектировании приложения была изучена методика создании технического задания для информационных систем, создание контекстных диаграмм в приложении Ramus Education, и правильное изучение предметной области.

При непосредственном программировании и функционального создания информационной системы были изучены многие аспекты и функции языка C#, а также некоторые методы применения в Windows Forms.

Также были изучены моменты проектирования дизайна интерфейса программы, такие как его удобность, понятность и привлекательность для пользователя.

Были реализованы алгоритмы для тренировки памяти с помощью ранее перечисленных методов.

Было проведено многократное тестирование для устранения ошибок в программе.

В конце курсовой работы было изучено написание руководства пользователя, после чего было создано для компьютерной логической игры «Запомни!». Был составлен план внедрения и сопровождения компьютерной логической игры.

# Список использованных источников

1. Алгоритм: понятие в информатике и его свойства [Электронный ресурс] – Режим доступа: https://skysmart.ru/articles/programming/algoritm-ponyatie-v-informatike-i-ego-svojstva
2. Внедрение: понятие и его свойства [Электронный ресурс] - https://sozidatel-mirov.github.io/
3. ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».
4. ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;
5. ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».
6. Официальный сайт Visual Studio [Электронный ресурс]. – Режим доступа: Visual Studio: IDE и редактор кода для разработчиков и групп, работающих с программным обеспечением (microsoft.com)
7. Официальная документация по языку C# от компании Microsoft [Электронный ресурс]. – Режим доступа: Documentation comments - document APIs using /// comments - C# reference | Microsoft Learn
8. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург : Изд-во ОГУ, 2015. – 89 с.
9. Тестирование информационных систем: понятие в информатике и его виды [Электронный ресурс] – Режим доступа: https://daglab.ru/testirovanie-informacionnyh-sistem/

# Приложение А

*(обязательное)*

**Текст программы**

**Разрешается использовать шрифт Times New Roman 10 пт для экономии места**