|  |  |
| --- | --- |
| 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 компьютерной логической игры Баше**  Пояснительная записка  ОГУ 09.02.07. 3024. 059 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 22ИСП-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л. К. Басин  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

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

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту \_\_\_\_\_Басину Льву Константиновичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

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

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

1. Тема работы: \_\_\_Разработка компьютерной логической игры Баше\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_\_июня\_\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Программно смоделировать выигрышную стратегию компьютера в логической игре Баше для двух игроков (компьютер-человек) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
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. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Л.К. Басин\_\_\_\_\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 059 ПЗ

Разраб.

Басин Л. К

Провер.

Михайличенко Ж.В

Реценз.

Н. Контр.

Утверд.

Разработка компьютерной логической игры Баше

Лит.

Листов

29

22ИСП-1

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

В первой главе курсовой работы приведён анализ требований и проектирование программного продукта, что включает в себя: анализ предметной области компьютерной логической игры «Баше», техническое задание на разработку программного обеспечения, а также построение моделей IDEF0.

Во второй главе курсовой работе рассмотрена реализация и тестирование программного продукта, которое включает в себя: обоснование программных средств реализации, разработку пользовательского интерфейса, алгоритмизацию и программирование и тестирование.

В третьей главе курсовой работы приведены рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения, включающее в себя руководство пользователя и план внедрения и сопровождения.

Пояснительная записка содержит 29 страницы, в том числе 7 рисунков, 3 таблицы, 11 источников, 1 приложение.

Разработка приложения выполнена в среде программирования Visual Studio 2022, c типом проекта Windows Forms на платформе .NET Framework.

**Содержание**

[Введение 5](#_Toc169187382)

[1 Анализ требований и проектирование программного продукта 6](#_Toc169187383)

[1.1 Анализ предметной области 6](#_Toc169187384)

[1.2 Техническое задание на разработку 7](#_Toc169187385)

[1.3 Построение моделей 10](#_Toc169187386)

[2 Реализация и тестирование программного продукта 13](#_Toc169187387)

[2.1 Обоснование программных средств реализации 13](#_Toc169187388)

[2.2 Разработка пользовательского интерфейса 14](#_Toc169187389)

[2.3 Алгоритмизация и программирование 16](#_Toc169187390)

[2.4 Тестирование 18](#_Toc169187391)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения 20](#_Toc169187392)

[3.1 Руководство пользователя 20](#_Toc169187393)

[3.2 План внедрения и сопровождения 22](#_Toc169187394)

[Заключение 24](#_Toc169187395)

[Список использованных источников 25](#_Toc169187396)

[Приложение А 26](#_Toc169187397)

# Введение

Индустрия компьютерных игр взошла на свет сравнительно недавно, примерно 30 лет назад, но уже успела превратиться в огромную отрасль с огромными доходами, достигающими нескольких миллиардов долларов в год. Такой внезапный рост популярности легко объяснить: это произошло благодаря широкому распространению компьютерных технологий, а также появлению интернета. Это позволило компьютерным играм стать более доступными для обычных пользователей по сравнению с другими видами развлечений. Достаточно иметь лишь компьютер и наличие доступа в интернет, чтобы можно было скачать игру, не выходя из дома. К тому же, для выбора подходящей игры пользователю не требуется особых знаний, в то время как для других видов развлечений может потребоваться понимание особенностей и экипировки. На сегодняшний день компьютерные игры уже не рассматриваются только как средство отдыха и развлечений. Например, с использованием игровых технологий создаются специализированные комплексы для обучения специалистов по различным профессиям, от лесорубов до пилотов реактивных самолетов.

Однако, в России и в других странах постсоветского пространства игровая индустрия развита не так хорошо. Это связано с тем, что культура компьютерных игр пришла к нам слишком поздно и практически не развивалась. Несмотря на значительный спрос, в стране слишком мало разработчиков, способных конкурировать с зарубежными компаниями.

Поэтому развитие технологий в данном направлении можно считать одним из наиболее перспективных, особенно для такой большой сверхдержавы, как Россия.

Объектом исследования является разработка компьютерной игры.

Предмет исследования: технологии разработки компьютерной логической игры.

Цель работы – разработка компьютерной логической игры «Баше» средствами среды Visual Studio 2022, с типом проекта Windows Forms на платформе .NET Framework.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. ознакомление с предметной областью курсовой работы;

2. создание технического задания;

3. построение моделей в виде IDEF0;

4. ознакомление и обоснование выбора программных средств реализации;

5. разработка пользовательского интерфейса;

6. разработка игры;

7. создание руководства пользователя;

8. защита курсовой работы.

# Анализ требований и проектирование программного продукта

## Анализ предметной области

Игра «Баше» представляет собой логическую игру, в которой игроки поочередно берут предметы из кучи. Цель игры - оставить сопернику последний предмет или предметы, чтобы выиграть.

Необходимо понять основные правила игры, включая количество предметов в начальной кучке, диапазон количества предметов, которые можно взять за один ход, и условия победы или поражения. Игра начинается с заданного количества предметов, разделенных на несколько кучек. Игроки делают ходы по очереди, выбирая одну из кучек и убирая из неё любое количество предметов (от одного и более). Игрок обязан взять хотя бы один предмет. Человек, сделавший последний ход и оставивший противнику последний предмет, выигрывает.

Анализ предметной области компьютерной логической игры «Баше» включает в себя:

1. Начальное состояние: если начальное количество предметов в каждой кучке является степенью двойки (один, два, четыре, восемь и т.д.), то игрок, который ходит первым, может гарантированно выиграть при правильной игре.
2. Логика игры: игрок должен стремиться создавать ситуации, когда сумма количества предметов в каждой кучке будет являться степенью двойки. Это позволит контролировать ходы соперника и выиграть.
3. Развитие логического мышления: игра «Баше» развивает логическое мышление, способность анализа и принятия решений на основе математических концепций. Игроки вынуждены думать стратегически, предвидеть ходы соперника и планировать свои действия заранее.
4. Варианты правил: существуют несколько вариантов правил для игры «Баше», которые могут изменять сложность игры и стратегии, необходимые для победы.
5. Анализ возможных ходов: для успешной игры в «Баше» игрок должен уметь анализировать возможные ходы, предвидеть развитие игры и выбирать оптимальные действия. Это требует умения рассчитывать вероятности и принимать решения на основе математических расчетов.

Чтобы всегда выигрывать, нужно знать «проигрышные числа»: три, шесть, девять, 12, 15 и так далее. Это значит, что, если на столе лежит такое количество камешков, надо отдать первый ход сопернику и в процессе игры оставлять ему каждый раз «проигрышное» количество камней. Если же на столе лежит не «проигрышное» число камней, то надо ходить первым и сразу же оставлять сопернику «проигрышное» количество камней. Таким образом, выигрыш всегда предопределен тем, кто ходит первым и сколько камешков лежит на столе в начале игры.

Программный код компьютерной логической игры «Баше», разрабатываемый в рамках курсовой работы, должен соответствовать вышеперечисленному описанию предметной области.

Выигрышная стратегия для компьютера в программной коде создана исходя из «проигрышных чисел»: шесть, девять и так далее. При анализе количества фишек в кучке, компьютер находит такое число и отнимает две фишки, тем самым, не давая игроку совершить последний ход и, следовательно, выигрывает, забирая оставшиеся фишки. Чтобы выиграть игроку, он должен рассуждать точно также, как и компьютер, находя «проигрышные числа».

## Техническое задание на разработку

Техническое задание на разработку программного обеспечения – документ, который содержит подробное описание требований к программному продукту. Техническое задание на разработку компьютерной логической игры «Баше» 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) Наименование автоматизированной системы (АС): «Разработка компьютерной логической игры Баше»

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

в) Наименование разработчика: студент второго курса группы 22ИСП-1 Басин Л. К.;

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: развитии у игроков навыков логического мышления, анализа, принятия решений и стратегического планирования;

б) Назначение АС: развитие пространственного воображения, сообразительности, смекалки, находчивости, а также увлечение;

3 Характеристика объекта автоматизации.

* необходимость иметь физические предметы, определённое количество свободного пространства и второго человека для проведения игры;
* подсчёт очков проходит в уме или на бумаге.

4 Требования к АС.

а) Требования к функциям, выполняемым АС:

* ввод исходных данных;
* управление игровым процессом;
* визуализация;
* автоматические проверки на соответствие правилам.

б) Требования к видам обеспечения:

* требования к математическому обеспечению: использование теории игр для анализа стратегий игроков и определения оптимального поведения, применение алгоритмов для обработки информации о ходах игроков и принятия решений на основе этих данных;
* требования к информационному обеспечению: игра может быть создана для работы на различных операционной системе Windows. Документация по игре, правила и инструкции для пользователей также должны быть представлены в удобной форме.
  + - требования к программному обеспечению: операционные системы: Windows, Linux, язык программирования: C#, интегрированная среда разработки: Microsoft Visual Studio 2022, средства документирования: Microsoft Word, средство построения бизнес-процессов Ramus Educational, приложение для быстрого создания файлов справки Dr.Explain;
    - требования к техническому обеспечению: система не должна предъявлять высоких требований к аппаратным средствам и легко запускаться на любой модели современных компьютеров;
* требования к организационному обеспечению: запуск и игра в «Баше» согласно правилам. Пользование интерфейсом игры для лёгкого управления функциями игры и выполнения задач.

г) Общие технические требования к АС:

* требования к численности и квалификации пользователей АС: система предназначена для одного пользователя и должна быть проста в эксплуатации для человека;
* требования к эргономике и технической эстетике: лёгкое управление всеми функциями и понятный, удобный интерфейс для пользователей.

5 Состав и содержание работ по созданию АС.

В таблице 1 показаны этапы разработки компьютерной логической игры «Баше».

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 Анализ требований | 19.02.2024 – 05.03.2024 | Анализ предметной области, изучение программных аналогов, разработка технического задания на создание АС. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 2 Проектирование | 06.03.2024 – 20.03.2024 | Разработка диаграмм IDEF различного уровня, диаграмм вариантов использования. Выбор и обоснование технологий и инструментов разработки. Проектирование пользовательского интерфейса. |
| 3 Программирование | 21.03.2024 – 14.05.2024 | 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 | 15.05.2024 – 31.05.2024 | Проведение функционального, модульного, интеграционного и системного тестирования. Исправление ошибок и несоответствий. |
| 5 Внедрение | 01.06.2024 – 10.06.2024 | Установка и настройка системы. Подготовка сопровождающей документации (пояснительной записки к курсовой работе, руководства пользователя и других). Защита курсовой работы. |
| 6 Эксплуатация и сопровождение | С 10.05.2024 | Постоянная поддержка пользователей и решение их проблем; Обновление и модификация системы по мере необходимости |

6 Порядок разработки АС: **проводится поэтапно в соответствии с таблицей пять и постоянным контролем заказчика.**

7 Порядок контроля и приёмки АС:

а) Методы испытания АС:

* тестирование функций: проверка выполнения каждой функции системы в соответствии с требованиями;
* тестирование системы: проверка работы системы в целом, включая её совместимость с окружающей инфраструктурой;
* **тестирование сценариев использования**: проверка работы системы в различных сценариях использования, чтобы убедиться, что все функции работают корректно и без ошибок.
* приёмка: после успешной защиты, система готова к внедрению и полноценной эксплуатации заказчиком.

8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие:

* установка программного обеспечения: разработчик устанавливает программное обеспечение компьютере пользователя в соответствии с требованиями системы;
* тестирование и отладка: после установки и настройки системы проводится тестирование на работоспособность, отладка возможных ошибок и сбоев;
* поддержка и сопровождение: после ввода системы в действие разработчик оказывает поддержку и сопровождение для решения возможных проблем и вопросов пользователей.

9 Требования к документированию:

* руководство пользователя содержит правила компьютерной игры «Баше»;
* руководство по сопровождению: описание процесса обновления и поддержки системы, информация о контактных лицах или службе поддержки для получения помощи;
* пояснительная записка к курсовой работе.

10 Источники разработки:

* Протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 года;
* 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. Схема алгоритмов, программ, данных и систем.»
* ГОСТ 34.601–90 «Информационные технологии. Комплекс стандартов на автоматизированное системы. Стадии и этапы создания автоматизированной системы»

## Построение моделей

Проектирование информационной системы играет ключевую роль в создании эффективных систем. Оно помогает структурировать данные и определять их взаимосвязи. Благодаря ему можно лучше понять бизнес-процессы, оптимизировать работу с данными. Правильно построенная информационная модель упрощает анализ и обработку данных и позволяет лучше понять бизнес-процессы.

Информационные модели можно реализовать в виде IDEF0 («Icam DEFinition for Function Modeling»). IDEF0 – это стандартный метод функционального моделирования, который позволяет описать функции системы с помощью диаграмм. Преимуществами являются: стандартизация бизнес-процессов, возможность документирования функциональных требований и взаимосвязей в наглядной форме и повышение эффективности и точности анализа бизнес-процессов. Существуют программы, в которых можно создавать визуальные диаграммы, одна из них Ramus Educational.

Функциональным блоком контекстной диаграммы является блок А0 «Функционирование компьютерной логической игры Баше». В качестве входных компонентов используются: право первого хода, количество фишек в кучке, в следствии чего, в выходных данных - сообщение о выигрыше и сообщение о проигрыше. В механизмах выявлены такие компоненты, как компьютер и пользователь, а в управлении правила игры.

Контекстная диаграмма IDEF0 представлена на рисунке 1.

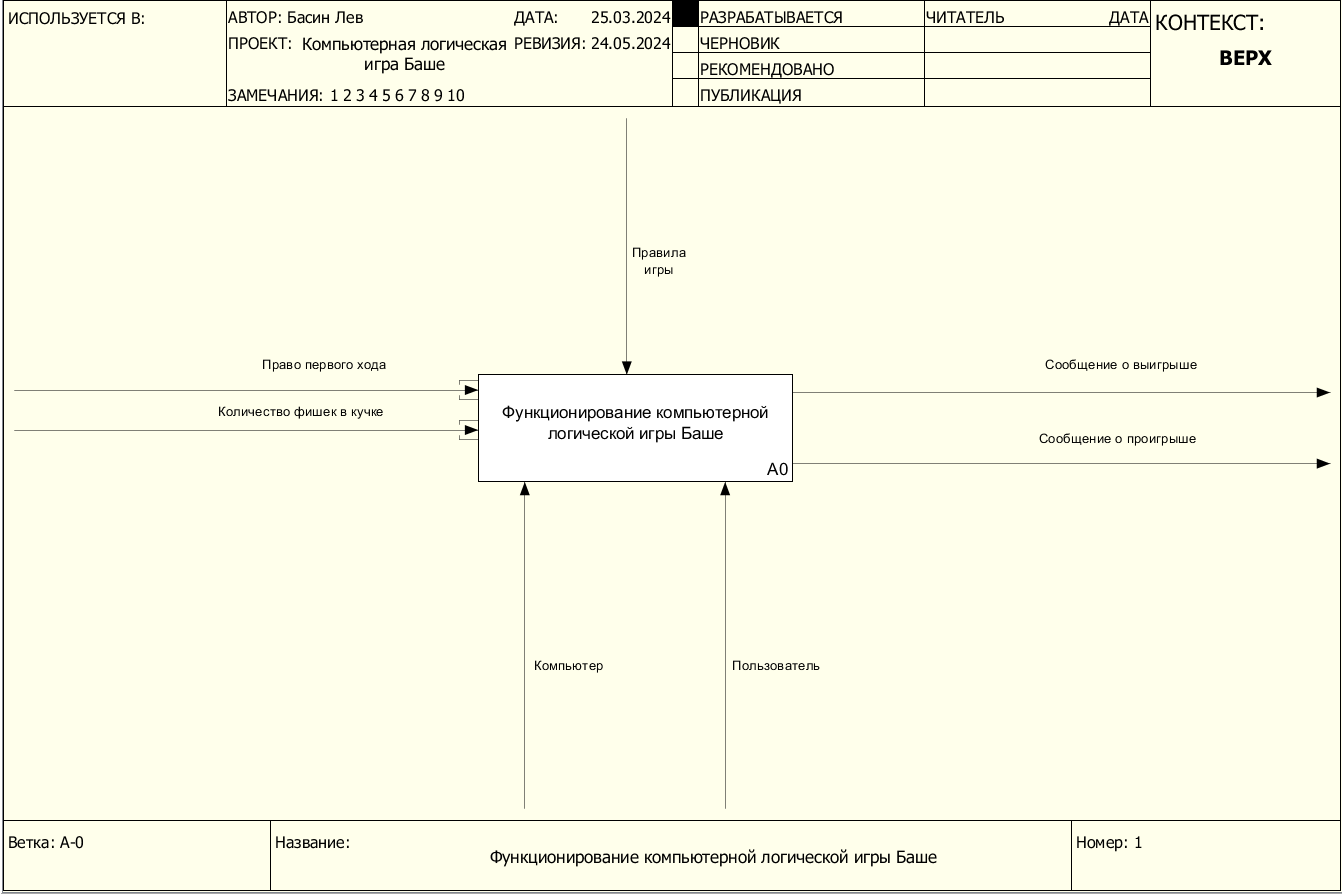


Рисунок 1 – Контекстная диаграмма IDEF0

Для того, чтобы лучше понять функции системы необходимо определить следующие функциональные блоки декомпозиции первого уровня:

А1 «Определение первого хода»,

А2 «Процесс игры»,

А3 «Определение победителя».

В управлении, во всех блоках используются правила игры. Механизмами функции для А1 и А2 стоят компьютер и пользователь, для А3 - компьютер. В качестве выходных компонентов у первого функционального блока используются: право первого хода, количество фишек в кучке, что являются входными данными для следующего блока «Процесс игры». Также в блок А2 «Процесс игры» включают дополнительные входные компоненты: количество взятых фишек. Выходными компонентами этого блока считаются: количество оставшихся фишек в кучке, которые входят в входные данные последнего блока «Определение победителя». На выходе в конце последнего блока «Определение победителя» пользователь сможет получить сообщение о выигрыше и сообщение о проигрыше.

Диаграмма декомпозиции первого уровня представлена на рисунке 2.

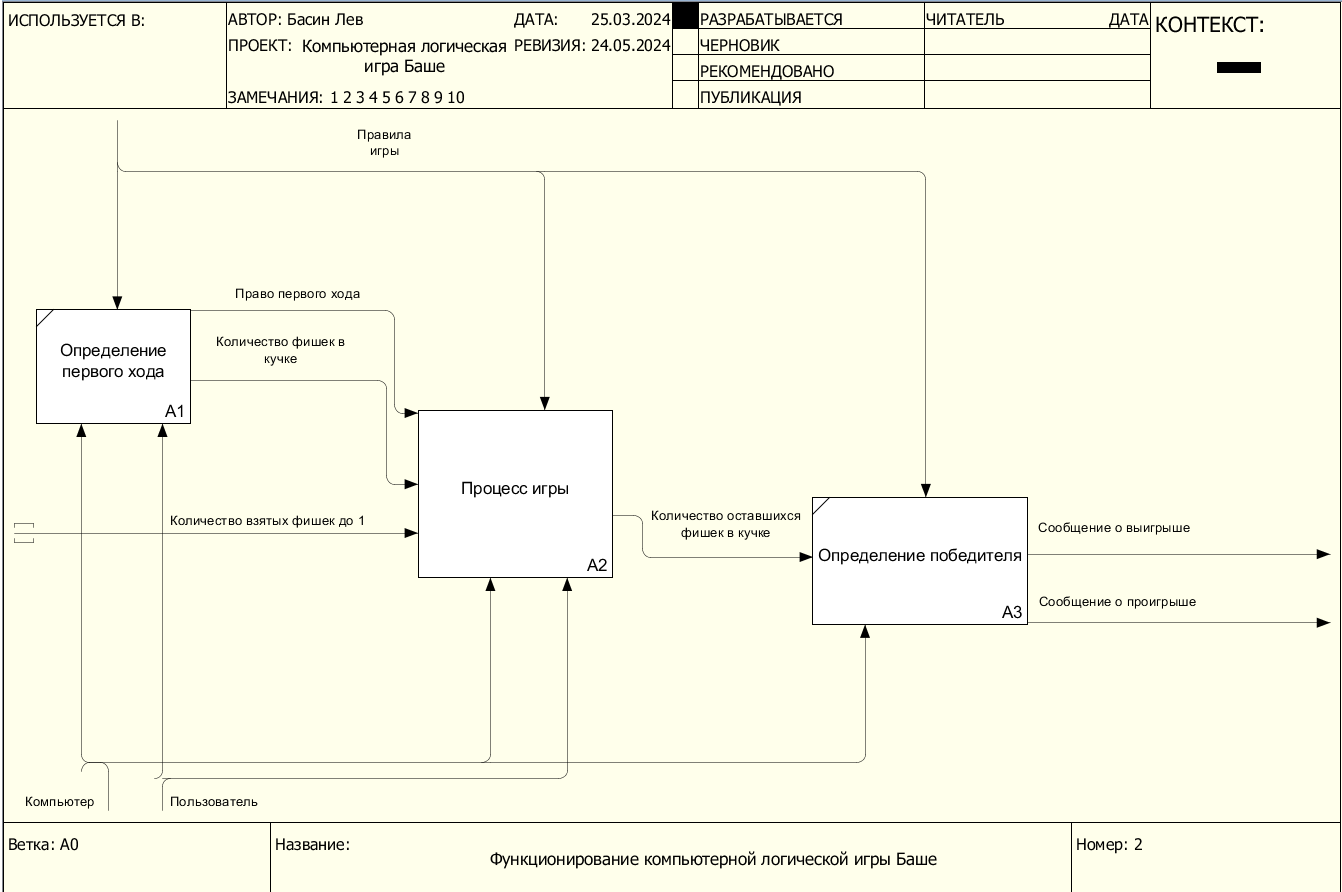


Рисунок 2 – Диаграмма декомпозиции IDEF0 первого уровня

Все построенные функциональные диаграммы будут использованы при разработке программного обеспечения.

# Реализация и тестирование программного продукта

## Обоснование программных средств реализации

Для разработки компьютерной логической игры «Баше» решено использовать язык программирования C# и интегрированную среду разработки Visual Studio 2022, тип проекта Windows Forms на платформе .NET Framework.

Язык C# отлично подходит как для начинающих, так и для опытных разработчиков. Преимуществами программирования на данном языке программирования являются:

* + - * простота изучения. C# является дружественным для новичков языком программирования. Синтаксис похож на обычный английский, что делает его лёгким для понимания;
      * язык поддерживает различные парадигмы программирования, включая объектно-ориентированное и функциональное программирование, что делает его мощным инструментом для разработки разнообразных приложений;
      * C# является кроссплатформенным языком программирования благодаря .NET Core, позволяя создавать приложения для различных операционных систем;
      * строгая типизация языка помогает выявлять ошибки на раннем этапе разработки, обеспечивая стабильность и надёжность приложения.

Ключевую роль в разработке приложений на C# играет Visual Studio 2022. Среда обеспечивает мощные инструменты для создания, отладки и управления проектами. Интегрированная среда разработки становится надёжным партнером для разработчиков благодаря:

* + - * расширенным инструментам отладки, так как Visual Studio 2022 предоставляет широкий спектр инструментов для отладки приложений, что облегчает обнаружение и исправление ошибок;
      * мощному интеллектуальному автозаполнению, поскольку с помощью функций интеллектуального автозаполнения и предсказания кода разработчики могут значительно увеличить производительность и точность своей работы;
      * Visual Studio 2022 рекомендует обширную библиотеку расширений и плагинов, которые позволяют настраивать среду разработки под конкретные потребности и повышают производительность разработчиков.

Подводя итоги, можно отметить, что программирование на C# с использованием Visual Studio 2022 открывает широкие возможности для создания качественных приложений и развития профессиональных навыков в области разработки ПО. Следовательно, выбор данного ПО вполне обоснован для разработки компьютерной логической игры Баше в рамках курсовой работы.

## Разработка пользовательского интерфейса

В условиях использования компьютерных информационных технологий актуальны вопросы организации взаимодействия человека с техническими и программными средствами. Такое взаимодействие обеспечивает программный интерфейс.

Программный интерфейс (API) представляет собой набор правил и протоколов, которые обеспечивают взаимодействие между различными компонентами программного обеспечения.

Программный интерфейс обладает рядом особенностей, что делает его неотъемлемой частью современных разработок.

* + - * API определяет стандартные способы взаимодействия между компонентами программного обеспечения, что облегчает их интеграцию и использование;
      * программный интерфейс скрывает внутреннюю реализацию компонентов, предоставляя только необходимый функционал. Это позволяет разработчикам использовать компоненты без необходимости знания всех деталей их работы;
      * способность расширения или поправок без внесения изменений в саму реализацию компонентов, что упрощает обновление и развитие программного обеспечения (ПО);
      * возможность управлять доступом к функционалу и данным, обеспечивая безопасное взаимодействие между компонентами;
      * хорошая документация помогает разработчикам понять, как использовать его функции, параметры и возможности.

Требования к программному интерфейсу:

* **простота и интуитивная понятность**;
* потребность быть эффективным и не требовать излишних действий со стороны пользователя;
* **и**нтерфейс должен быть гибким, чтобы удовлетворить разнообразные потребности пользователей;
* необходимость быть надёжным и стабильным в работе;
* **хорошая производительность**. Интерфейс должен обладать хорошей производительностью, работать быстро и отвечать на запросы пользователя мгновенно;
* **хорошая документация.**

Следовательно, API играет важную роль в современной разработке ПО, обеспечивая эффективное взаимодействие между различными компонентами. Его особенности, требования и преимущества делают API неотъемлемой частью процесса создания качественного и инновационного ПО. Внимательное проектирование и использование API способствует улучшению разработки и интеграции приложений, повышению их масштабируемости и безопасности.

Программный интерфейс компьютерной игры «Баше» должен соответствовать всем вышеперечисленным требованиям. Структура интерфейса и его компоненты показаны на рисунке 3.

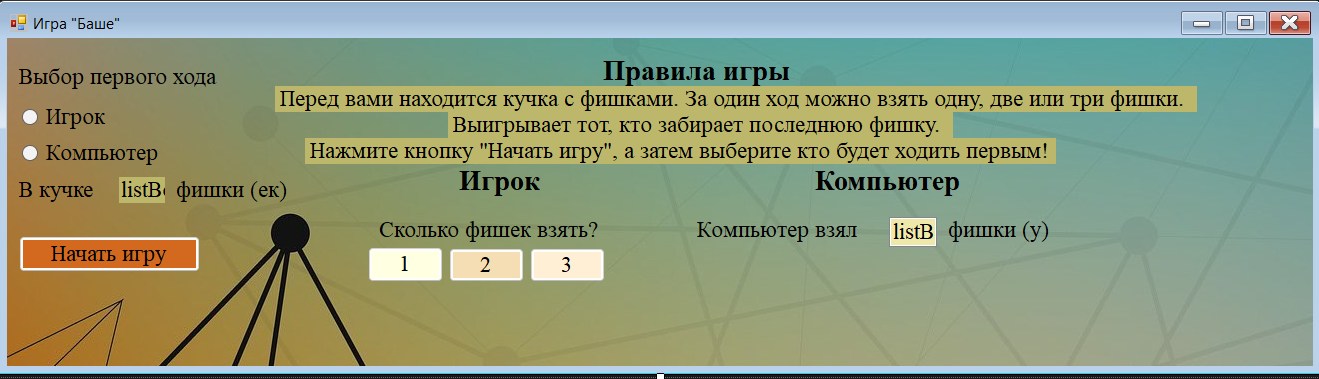


Рисунок 3 – Структура интерфейса и его компонентов

На главной странице приложения находятся четыре кнопки: «Начать игру», «1», «2», «3» и две radioButton «Игрок» и «Компьютер». Первая кнопка генерирует количество фишек в кучке в диапазоне от 10 до 50 и даёт возможность нажать на одну из двух radioButton. RadioButton «Игрок» и «Компьютер» используются для того, чтобы пользователь мог решить кто в игре будет ходить первым. При нажатии на любую radioButton, пользователю открывается доступ к следующим трём кнопкам «1», «2» и «3». Если игрок нажмёт на radioButton «Компьютер», то из кучки будут взяты от одного до трёх фишек от лица компьютера. При нажатии на другую radioButton пользователь сможет взять фишки первым. Кнопки «1», «2» и «3» определяют сколько пользователь возьмёт фишек из кучки. Исходя из названия кнопок, нажимая на кнопку «1» будет взята одна фишка, «2» - две фишки, «3» - три фишки. Также на главной странице сверху есть правила игры, слева - listBox с количеством фишек в кучке и справа - listBox с количеством фишек, которое взял компьютер. При победе компьютера или игрока появляется окно messageBox с информацией о победителе, рисунок 4.

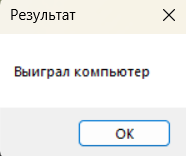


Рисунок 4 – MessageBox с результатом игры

## Алгоритмизация и программирование

Алгоритм — это чёткое описание последовательности действий, которые необходимо выполнить ля решения поставленной задачи.

Основные свойства алгоритмов:

* Конечность (результативность): определяет возможность получения результата за конечное число шагов;
* Дискретность: любой процесс алгоритма можно разбить на определённые этапы или шаги;
* Определённость: каждое правило алгоритма должно быть чётким и однозначным, а также выполниться в заданной последовательности;
* Массовость: получение результата при любых исходных данных.

Алгоритмы удобно представлять в виде блок-схем, согласно, ГОСТ 19.701-90 «Единая система программной документации. Схема алгоритмов, программ, данных и систем». Для компьютерной логической игры «Баше» блок-схема алгоритма хода компьютера представлена на рисунке 5.

нет

Начало хода

В кучке фишек больше нуля?

В кучке одна фишка?

В кучке две фишки?

В кучке три фишки?

Отнять из кучки одну фишку

Отнять из кучки две фишки

Отнять из кучки три фишки

да

нет

да

да

нет

да

нет

Победил компьютер

Конец хода

Отнять из кучки случайное количество фишек от одного до трёх

В кучке шесть или девять фишек?

Отнять из кучки две фишки

нет

да

Рисунок 5 - Блок-схема алгоритма хода компьютера для игры «Баше»

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

1) private void Button7\_Click(object sender, EventArgs e) - старт игры;

2) private void Button1\_Click(object sender, EventArgs e) - метод взятия одной фишки из кучки;

3) private void Button2\_Click(object sender, EventArgs e) - метод взятия двух фишек из кучки;

4) private void Button3\_Click(object sender, EventArgs e) - метод взятия трёх фишек из кучки;

5) private void radioButton1\_CheckedChanged(object sender, EventArgs e) - начало хода игроком;

6) private void radioButton2\_CheckedChanged(object sender, EventArgs e) - начало хода компьютером;

7) private void listBox1\_SelectedIndexChanged(object sender, EventArgs e) - вывод информации о количестве фишек в кучке;

8) private void listBox2\_SelectedIndexChanged(object sender, EventArgs e) - вывод информации о том, сколько фишек взял компьютер;

9) private static double Comp(double first) – ход компьютера.

## Тестирование

Тестирование информационной системы - очень важное направление, без которого невозможен был бы запуск ни одного сайта, мобильного приложения или игры.

Тестирование информационной системы (ИС) – проверка её на соответствие требованиям, предъявляемым заказчиком и записанном в техническом задании.

Тестирование стоит рассматривать как своеобразный фильтр, который помогает выявить слабые места в коде, структуре данных и функциональности продукта. Этот процесс позволяет разработчикам устранять обнаруженные дефекты на ранних этапах разработки. Это экономит время, ресурсы и обеспечивает высокий уровень удовлетворенности конечных пользователей.

Следующие виды тестирования будут использованы при создания компьютерной логической игры «Баше»:

* Функциональное тестирование - проверка поведения программы при соответствии функциональным требованиям. В него также входит тестирование различных функций и возможностей программы для определения, соответствуют ли они заявленным требованиям.
* Компонентное (модульное) тестирование - тестирование отдельных компонентов или модулей программы для проверки их работоспособности в отрыве от других компонентов.
* Интеграционное тестирование - проверка взаимодействия между различными компонентами или модулями программы после их объединения, чтобы убедиться, что они работают вместе корректно.
* Нагрузочное тестирование - оценка производительности и устойчивости программы при различных нагрузках, например, при большом количестве одновременных запросов или пользователей.
* Тестирование эргономики решений - проверка удобства использования интерфейса программы для обеспечения удовлетворения потребностей и ожиданий пользователей.
* Тестирование надёжности - оценка стабильности и надёжности программы путем проверки её работоспособности в различных условиях и обстоятельствах, а также выявление потенциальных проблем или ошибок.

Эти различные виды тестирования помогают обеспечить качество программного обеспечения и его соответствие заявленным требованиям.

Результаты приёмки компьютерной логической игры «Баше» представлены в таблице 2.

Таблица 2 – Результат тестирования

|  |  |
| --- | --- |
| Вид тестирования | Оценка по 10 бальной шкале |
| Функциональное |  |

Продолжение таблицы 2

|  |  |
| --- | --- |
| Вид тестирования | Оценка по 10 бальной шкале |
| Компонентное (модульное) |  |
| Интеграционное |  |
| Нагрузочное |  |
| Эргономики интерфейса |  |
| Надёжности |  |

# Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения

## Руководство пользователя

1. Системные требования

Для стабильной и эффективной работы компьютерной логической игры «Баше» рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): 1.1 GHz

Количество ядер процессора (CPU): 4

Объем оперативной памяти (RAM): 4 GB

Объем свободного места на диске (HDD): 1 GB

Операционная система (OS): Windows

Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge

2. Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с расчётами в игре «Баше».

Перед началом работы, пожалуйста, ознакомьтесь с системными требованиями и лицензионным соглашением.

3. Основные понятия и термины

Перед началом работы с игрой «Баше» рекомендуем ознакомиться с основными понятиями и терминами:

Компьютер - электронно-вычислительная машина, обрабатывающая и хранящая информацию, производящая вычисления и передающая результаты этих вычислений под управлением заложенных программ.

Программа - последовательность машинных команд, предназначенная для достижения конкретного результата.

C# - объектно-ориентированный язык программирования общего назначения.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

Алгоритм — это четкая последовательность действий, выполнение которой дает какой-то заранее известный результат.

Ramus — программа для построения визуальных диаграмм, позволяющая наглядно отображать различные бизнес-процессы. Главное преимущество Ramus Educational заключается в том, что она поддерживает сразу две популярные методологии: DFD и IDEF0.

Баше́ — математическая игра, в которой два игрока из кучки, содержащей первоначально N предметов, по очереди берут не менее одного и не более М предметов. Проигравшим считается тот, кому нечего брать.

4. Установка

Для установки компьютерной логической игры «Баше» загрузить дистрибутив последней версии 1.0, доступный по адресу https://github.com/Leonolev/kursovaya1.1.git.

Перед установкой ознакомьтесь с системными требованиями и лицензионным соглашением.

В процессе скачивания игры не выключайте программу установки.

5. Настройка

Для начала работы в компьютерной логической игры «Баше» рекомендуем предварительно выполнить следующие настройки:

* Убедиться, что компьютер соответствует системным требованиям игры
* Установить файл игры.

При настройке необходимо учесть системные требования игры.

6. Запуск

Для запуска компьютерной игры «Баше» нажмите на ярлык «Баше» в меню Пуск либо наберите в командной строке.

При первом запуске программы вы увидите главный экран приложения с игрой.

7. Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса компьютерной игры «Баше» 1.0, режимов работы, предназначения окон и экранов, доступные операции.

Содержание раздела

* главное окно программы;
* настройки программы;
* режимы работы;
* горячие клавиши.

8. Главное окно программы

Главное окно программы компьютерной логической игры «Баше» позволяет выполнять операцию: взаимодействие с игровыми элементами приложения.

Главное окно программы представлено на рисунке 6.

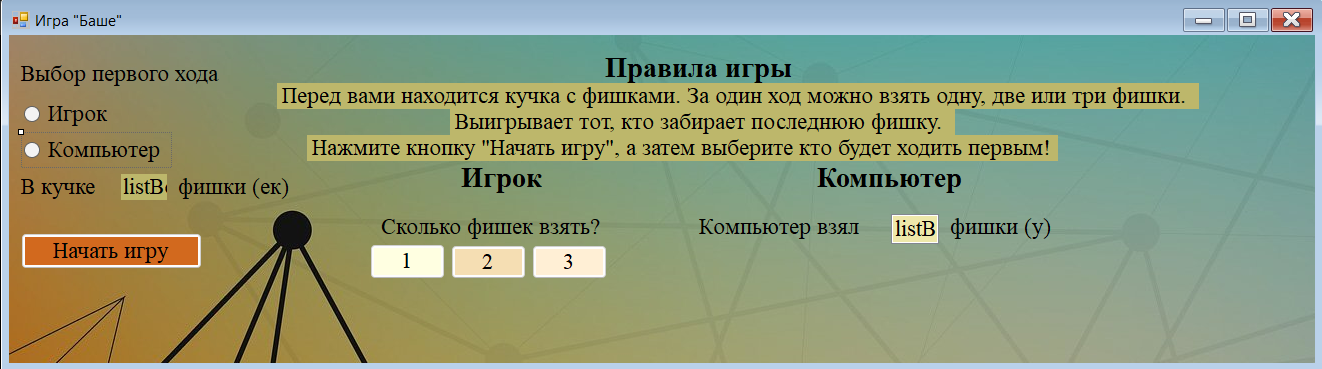


Рисунок 6 – Главное окно программы

9. Настройки программы

Окно Настройки программы позволяет редактировать параметры и глобальные настройки компьютерной логической игры «Баше».

10. Режимы работы

Пользовательский интерфейс компьютерной логической игры «Баше» обеспечивает работу только в режиме пользователя. Пользователю доступны все права. Режим работы программы представлен на рисунке 7.

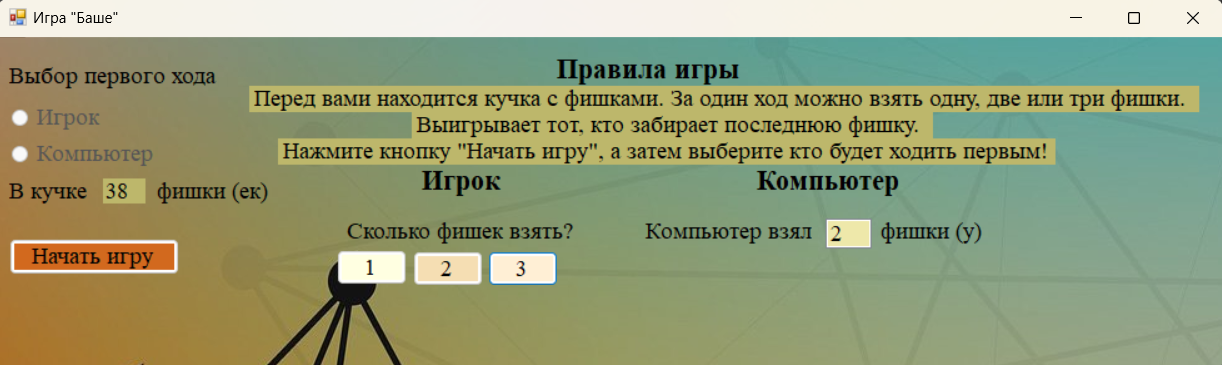


Рисунок 7 – Режим работы программы

11. Горячие клавиши

Раздел содержит все сочетания клавиш и способы управления при помощи мыши, поддерживаемые компьютерной логической игры «Баше».

ЛКМ – управление игровыми элементами игры

Устранение типовых проблем

Если программа не работает, нужно переустановить программу.

Контактная информация

Техническая поддержка

Вы можете направить вопросы по функциональности программы Название Продукта следующими способами:

* Email: Basinlew2006@mail.ru
* Телефон: +7 9619209044

Сайт продукта

https://github.com/Leonolev/kursovaya1.1.git

## План внедрения и сопровождения

Внедрение информационной системы — процесс настройки ПО и технических средств под определённые условия использования, а также обучение пользователей работе с системой. Цель внедрения информационной системы (ИС) состоит в улучшении эффективности, автоматизации процессов и улучшении управления бизнес-операциями.

Сопровождение информационной системы – процесс улучшения, оптимизации и устранения дефектов ПО после передачи ИС в эксплуатацию. Цель сопровождения системы - обеспечение её бесперебойной работы, адаптация к изменяющимся потребностям бизнеса и обеспечение безопасности и надёжности операций.

Сопровождение информационной системы включает в себя:

* Модификацию ИС при сохранении её целостности;
* Поддержку текущего состояния ИС и её функциональные пригодности;
* Оптимизацию работы ИС;
* Адаптацию ИС к новым условиям;
* Поддержку пользователей (консультация)
* Устранение дефектов ИС.

В таблице 3 представлен план внедрения и сопровождения компьютерной логической игры «Баше».

Таблица 3 – План внедрения и сопровождения

|  |  |
| --- | --- |
| Мероприятия | Сроки (дней) |
| Подготовка оборудования и технического обеспечения | 2 |
| Инсталляция Visual Studio 2022 | 1 |
| Инсталляция ИС | 1 |
| Обучение персонала работе с ИС на эксплуатацию | 7 |
| Поддержка текущего состояния ИС, | 365 |
| Поддержка пользователей (консультация) | 365 |

# Заключение

В ходе выполнения курсовой работы была разработана компьютерная логическая игра «Баше» в интегрированной среде разработки Visual Studio 2022, с типом проекта Windows Forms на платформе .NET Framework. Был получен опыт написания программ на языке программирования C#.

В результате проделанной работы были выполнены поставленные, в начале курсовой работы, задачи и была достигнута заданная цель, а именно создание компьютерной логической игры «Баше».

Программный код написан с учётом конкретных требований постановки задачи, а интерфейс игры прост и удобен для использования. Тестирование программы подтвердило её корректную работу. Программа создана доступно и ясно, что даёт возможность легко модифицировать её.

В ходе работы были выполнены основные этапы для создания компьютерной логической игры от изучения предметной области игры и разработки технического задания до тестирования продукта.

Разработка приложения велась на языке программирования C# с использованием интегрированной среды разработки Visual Studio 2022. Это позволило получить эффективное и интуитивно понятное программное обеспечение. Программный интерфейс и алгоритмы игры были структурированы и описаны с помощью блок-схемы для лучшего понимания логики функционирования игры.

В ходе анализа требований и проектирования программного продукта были применены методы IDEF0 для построения моделей, что позволило эффективно объяснить функциональные возможности и взаимодействие различных компонентов игры. При разработке были использованы знания, полученные из различных источников, что облегчило выполнение поставленных задач.

В процессе разработки компьютерной логической игры «Баше» были изучены принципы объектно-ориентированного программирования (ООП), которые успешно применялись при создании различных функций игры. Использование парадигмы ООП позволило улучшить модульность, повысить понятность кода и обеспечить лёгкую расширяемость проекта.

В целом, курсовая работа по разработке компьютерной логической игры «Баше» позволила овладеть навыками проектирования, программирования и тестирования программного обеспечения. Полученный опыт и навыки будут важны для дальнейшей работы и личного развития.

# Список использованных источников

1. GitHub с курсовой работой и компьютерной логической игры «Баше» - https://github.com/Leonolev/kursovaya1.1.git (дата обращения 13.06.2024).
2. ГОСТ 19.701–90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначение условные и правила выполнения.» [Электронный ресурс] – https://is20-2019.susu.ru/wp-content/uploads/2021/04/gost\_19.701-90.pdf (дата обращения: 05.04.2024).
3. ГОСТ 19.701–90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.».
4. ГОСТ 34.601–90 «Стадии и этапы создания автоматизированной системы.».
5. Построение диаграммы бизнес-процесса с использованием методики IDEF0 [Электронный ресурс] - <https://spravochnick.ru/avtomatizaciya_tehnologicheskih_processov/postroenie_diagrammy_biznes-processa_s_ispolzovaniem_metodiki_idef0/> (дата обращения: 21.04.2024).
6. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург : Изд-во ОГУ, 2015. – 89 с.
7. Тестирование [Электронный ресурс] – https://studfile.net/preview/5049163/page:2/ (дата обращения: 11.06.2024).
8. Требования к программному интерфейсу [Электронный ресурс] - <https://studfile.net/preview/7851430/page:2/> (дата обращения 09.05.2024).
9. Уроки C# – MessageBox – Урок 8 [Электронный видео ресурс] – https://yandex.ru/video/preview/18112039786235575126 (дата обращения: 05.04.2024).
10. Что такое алгоритмы? [Электронный ресурс] – https://ru.hexlet.io/blog/posts/chto-takoe-algoritmy (дата обращения: 05.04.2024).
11. Что такое Ramus Education? [Электронный ресурс] –<https://softrare.space/ru/windows/ramus/> (дата обращения: 20.04.2024).

# Приложение А

(обязательное)

**Текст программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Rebar;

namespace play

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

double first\_kuchka; // Создание переменной для работы с количеством фишек из кучки

bool vkl = false; // Создание переменной для блокировки и разблокировки кнопок выбора первого хода

bool vikl = false; // Создание переменной для блокировки и разблокировки кнопок выбора первого хода

// Генерация фишек в кучке от 10 до 50 и блокировка и разблокировка кнопок взятия фишек и выбора первого хода

private void button7\_Click(object sender, EventArgs e)

{

vikl = false;

listBox2.Items.Clear();

listBox1.Items.Clear();

Random rand = new Random();

first\_kuchka = rand.Next(10, 51);

listBox2.Items.Add(first\_kuchka.ToString());

if (vikl == true)

{

button1.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

}

radioButton1.Enabled = true;

radioButton2.Enabled = true;

radioButton1.Checked = false;

radioButton2.Checked = false;

vkl = true;

}

private void button1\_Click(object sender, EventArgs e)

{

first\_kuchka -= 1; // Взятие из кучки одной фишки

if (first\_kuchka == 1 || first\_kuchka == 2 || first\_kuchka == 3) // Определение победителя в игре на основе количества фишек в кучке

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл компьютер", "Результат");

}

else if (first\_kuchka == 0)

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл игрок", "Результат");

}

double res = Comp(first\_kuchka); // Запись в новую переменную количества фишек, взятые компьютером

double first\_kuchka\_save = first\_kuchka; // Сохранение информации о количестве фишек в кучке до взятия компьютером

first\_kuchka = res; // Присваивание информации о количестве фишек после взятия компьютером

double giveComp = first\_kuchka\_save - res; // Определение того, сколько фишек взял компьютер

listBox1.Items.Clear();

listBox1.Items.Add(giveComp.ToString()); // Запись в listBox информации о том, сколько фишек взял компьютер

listBox2.Items.Clear();

listBox2.Items.Add(res.ToString()); // Запись в listBox информации о том, сколько фишек осталось в кучке

// Блокировка определённых кнопок взятия фишек для более удобной работы

if (first\_kuchka == 0)

listBox2.Items.Clear();

if (first\_kuchka < 1)

{

button1.Enabled = false;

button2.Enabled = false;

button3.Enabled = false;

}

if (first\_kuchka == 1)

button2.Enabled = false;

if (first\_kuchka == 2)

button3.Enabled = false;

}

private void button2\_Click(object sender, EventArgs e)

{

first\_kuchka -= 2; // Взятие из кучки двух фишек

if (first\_kuchka == 1 || first\_kuchka == 2 || first\_kuchka == 3) // Определение победителя в игре на основе количества фишек в кучке

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл компьютер", "Результат");

}

else if (first\_kuchka == 0)

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл игрок", "Результат");

}

double res = Comp(first\_kuchka); // Запись в новую переменную количества фишек, взятые компьютером

double first\_kuchka\_save = first\_kuchka; // Сохранение информации о количестве фишек в кучке до взятия компьютером

first\_kuchka = res; // Присваивание информации о количестве фишек после взятия компьютером

double giveComp = first\_kuchka\_save - res; // Определение того, сколько фишек взял компьютер

listBox1.Items.Clear();

listBox1.Items.Add(giveComp.ToString()); // Запись в listBox информации о том, сколько фишек взял компьютер

listBox2.Items.Clear();

listBox2.Items.Add(res.ToString()); // Запись в listBox информации о том, сколько фишек осталось в кучке

// Блокировка определённых кнопок взятия фишек для более удобной работы

if (first\_kuchka == 0)

listBox2.Items.Clear();

if (first\_kuchka < 2)

{

button2.Enabled = false;

button3.Enabled = false;

}

if (first\_kuchka < 1)

button1.Enabled = false;

if (first\_kuchka == 2)

button3.Enabled = false;

}

private void Button3\_Click(object sender, EventArgs e)

{

first\_kuchka -= 3; // Взятие из кучки трёх фишек

if (first\_kuchka == 1 || first\_kuchka == 2 || first\_kuchka == 3) // Определение победителя в игре на основе количества фишек в кучке

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл компьютер", "Результат");

}

else if (first\_kuchka == 0)

{

listBox2.Items.Clear();

listBox2.Items.Add("0");

MessageBox.Show("Выиграл игрок", "Результат");

}

double res = Comp(first\_kuchka); // Запись в новую переменную количества фишек, взятые компьютером

double first\_kuchka\_save = first\_kuchka; // Сохранение информации о количестве фишек в кучке до взятия компьютером

first\_kuchka = res; // Присваивание информации о количестве фишек после взятия компьютером

double giveComp = first\_kuchka\_save - res; // Определение того, сколько фишек взял компьютер

listBox1.Items.Clear();

listBox1.Items.Add(giveComp.ToString()); // Запись в listBox информации о том, сколько фишек взял компьютер

listBox2.Items.Clear();

listBox2.Items.Add(res.ToString()); // Запись в listBox информации о том, сколько фишек осталось в кучке

// Блокировка определённых кнопок взятия фишек для более удобной работы

if (first\_kuchka == 0)

listBox2.Items.Clear();

if (first\_kuchka < 3)

button3.Enabled = false;

if (first\_kuchka < 2)

button2.Enabled = false;

if (first\_kuchka < 1)

button1.Enabled = false;

}

// Ход компьютера

private static double Comp(double first)

{

if (first > 0)

{

if (first == 1)

first -= 1;

else if (first == 2)

first -= 2;

else if (first == 3)

first -= 3;

else // Взятие случайного значения в диапазоне от 1 до 3 фишек из кучки

{

if (first % 3 == 0)

first -= 2;

else

{

Random rand = new Random();

first -= rand.Next(1, 4);

}

}

}

return first; // Возвращение информации об оставшиемся количестве фишек в кучке

}

private void Form1\_Load(object sender, EventArgs e)

{

}

// Определение первого хода, первый ход игрока

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

radioButton1.Checked = false;

if (vkl == true) //Разблокировка кнопок взятия фишек из кучки и блокировка кнопок определения первого хода

{

radioButton1.Enabled = false;

radioButton1.Checked = false;

radioButton2.Enabled = false;

button1.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

vkl = false;

vikl = true;

}

}

// Определение первого хода, первый ход компьютера

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

radioButton2.Checked = false;

if (vkl == true) //Разблокировка кнопок взятия фишек из кучки и блокировка кнопок определения первого хода

{

// Генерация и взятие из кучки определенного значения фишек от 1 до 3 компьютером

double firstStep = 0;

Random rand = new Random();

firstStep = rand.Next(1, 4);

first\_kuchka = first\_kuchka - firstStep;

listBox1.Items.Clear();

listBox1.Items.Add(firstStep.ToString()); // Запись в listBox информации о том, сколько фишек взял компьютер

listBox2.Items.Clear();

listBox2.Items.Add(first\_kuchka.ToString()); // Запись в listBox информации о том, сколько фишек осталось в кучке

radioButton1.Enabled = false;

radioButton2.Enabled = false;

radioButton2.Checked = false;

button1.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

vkl = false;

vikl = true;

}}}}