

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Практическая работа 1-2**

**Спецификация требований и создание вариантов использования. Реализация алгоритма решения задачи согласно процедурной парадигме**

Выполнил: Ефимов

Павел Леонидович

Группа № К3120

Проверила: Казанова П.П.

Санкт-Петербург

2020

**Цель работы:**

Понять принципы составления требований к ПО. Уяснить принципы разработки программ согласно процедурной парадигме. Разработать программу, реализующие функциональные требования.

**Ход работы:**

**Исходные данные:**

1.Игрок выбирает, на что он делает ставку – чет или нечет. Делает ставку.

2.Запускается сеанс игры и если ставка сыграла, то величина ставки приплюсовывается к текущему счету клиента, в противном случае – списывается.

3.Игра продолжается до тех пор, пока баланс игрока положительный. После определенного розыгрыша игрок может закончить игру и забрать выигрыш.

**План разработки системы:**

1.На основе исходных данных сформировать образ проекта.

2.Предварительно оценить возможные риски и необходимые ресурсы.

3.Составить план, в котором отразить основные опорные точки процесса разработки.

4.Определить основную функциональность, которую должна предоставлять система.

5. На основе функциональных требований создать модель прецедентов.

**Перечень пользовательских и функциональных требований:**

Данная модель показывает функции системы, их окружение и связи между прецедентами и актерами. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. Общие требования к поведению проектируемой системы представлены в Таблице 1.

Таблица 1 - Перечень пользовательских и функциональных требований

|  |  |  |  |
| --- | --- | --- | --- |
| **URS ID** | **Описание UR** | **FRS ID** | **Описание FR** |
| **Функция: Игра «Чет-Нечет»** | | | |
| 1 | Система обеспечивает возможность пользователю начать игру, сделав выбор («Четное» или «Нечетное» значение). | 101 | Взаимодействие с пользователем в виде вывода ожидаемых от него действий на экран. |
| 102 | Вывод в консоль баланса игрока. |
| 103 | Реализация выбора игрока после трёх ставок. |
| 2 | Система обеспечивает возможность пользователю вводить значение ставки. | 201 | Получение программой значения, которое вводит игрок. |
| 202 | Сохранение программой значения ставки. |
| 3 | Система обеспечивает пользователю видеть результаты игры. | 301 | Реализация выбора случайного числа. |
| 302 | Проверка совпадает ли четность числа с предположением игрока. |
| 303 | Изменение количества очков игрока. |
| 304 | Вывод результата на экран. |
| 305 | Вывод текущего количества очков игрока на экран. |
| 4 | Система обеспечивает пользователю продолжить или завершить игру. | 401 | Повторение программой алгоритма игры. |
| 402 | Вывод сообщения об окончании игры и состояния счета игрока. |

**Спецификация требований к ПО:**

1.Введение

Данное приложение предназначено для развлекательных целей. Документ предназначен для чтения как разработчикам, так и потенциальным пользователям. Система предназначена для использования студентами.

2.Общее описание

Данное приложение работает в среде разработки Microsoft Visual Studio и на платформах, поддерживающих язык программирования C#. Программа ограничена языком программирования C# и консолью ввода-вывода.

3.Функции системы

Для запуска игры использована функция Main(), в которой реализован диалог с пользователем. После вызывается функция Bet(int value), в которой игрок делает свою ставку, вводит её размер (при этом проверяется правильность ввода числа, больше 0 и меньше баланса, также нельзя вводить значения не соответствующие целочисленным). Вызывается функция Chance(), в которой игрок выбирает на что ставить «Чёт» или «Нечёт». Вызывается функция RandomNumber(), которая, генерируя случайное число, возвращает остаток от деления на 2. Если результат функции RandomNumber() совпадает со ставкой, то выводится сообщение о выигрыше и баланс увеличивается, в противном случае выводится сообщение о проигрыше и баланс уменьшается. В конце раунда вызывается функция Endgame(), в которой игрок выбирает продолжать игру или нет. Также есть проверка на баланс игрока, если баланс равен 0, то заканчивается игра.

**Диаграмма выбора использования:**

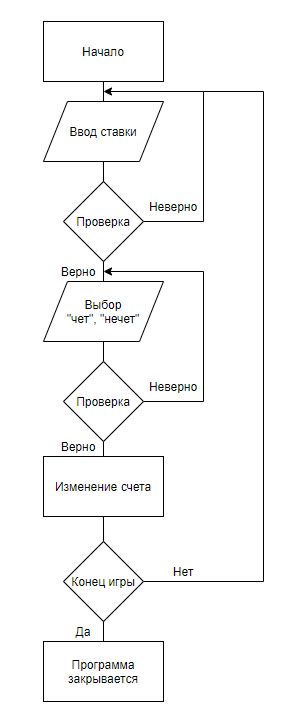


Рисунок – Диаграмма выбора действия

**Реализация функций:**

Была создана функция Bet(int) для ввода ставки игрока. Код функции изображен на рисунке 1.

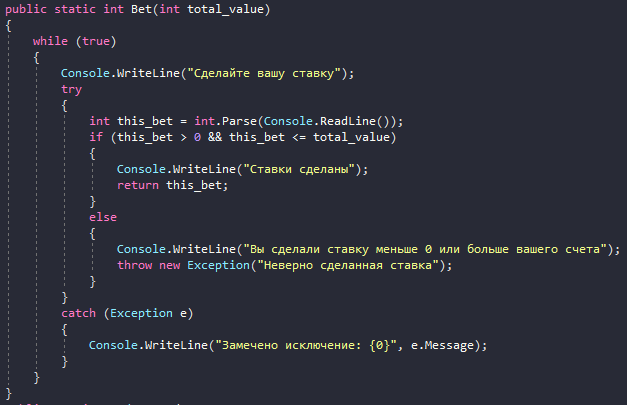


Рисунок – функция Bet(int)

Создана функция RandomNumber(), для создания случайной величины. Код функции изображен на рисунке 3.

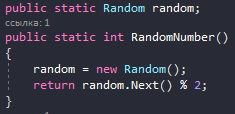


Рисунок – функция RandomNumber()

Создана функция Chance(), для выбора игрока ставки «чет», «нечет». Код изображен на рисунке 4.

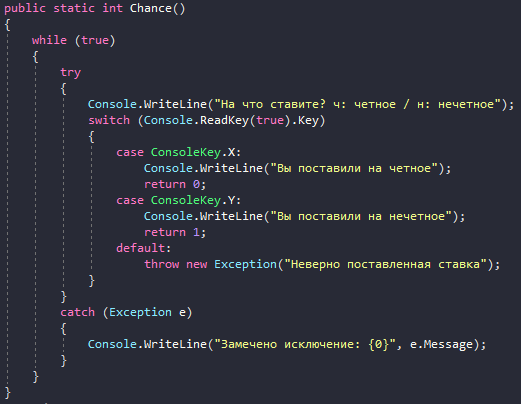


Рисунок – функция Chance()

Создана функция Endgame() для возможности закончить игру. На рисунке 5 изображен код.



Рисунок – функция Endgame()

**Вывод:**

В ходе выполнения практической работы, была реализована программа «Игра Чет-Нечет» согласно процедурной парадигме. Изучены принципы составления требований к ПО, уяснены принципы разработки программ согласно процедурной парадигме.