Университет ИТМО

Практическая работа №1 и 2

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

Выполнил:

Гамеза Денис Владимирович

Группа № K3121

Проверила:

Филимонова Ирина Александровна

Санкт-Петербург

2020

**Цель работы:** понять принципы документирования требований к ПО, принципы разработки программ согласно процедурной парадигме, а также разработать программу, реализующую функциональные требования.

**Ход работы:**

Шаг 1. Описание исходных данных проекта.

Игрок делает ставку – чет или нечет. Проходит раунд игры. Если ставка сыграла, то к текущему счету игрока прибавляется сумма ставки, в противном случае – вычитается. Игра продолжается до тех пор, пока баланс игрока положительный. После каждого раунда игры игрок может забрать выигрыш и закончить игру.

Шаг 2. План разработки системы.

Понять принципы игры. Оценить возможные риски и необходимые ресурсы. Составить план возможных действий игрока. Составить план функционирования программы. Составить алгоритм программы, учитывая все возможные ситуации. Продумать интерфейс, с помощью которого пользователь будет взаимодействовать с системой.

Шаг 3. Перечень пользовательских и функциональных требований.

| **URS**  **ID** | **Описание UR** | **FRS**  **ID** | **Описание FR** |
| --- | --- | --- | --- |
| **Функция: *Игра чёт-нечёт*** | | | |
| 1 | Система обеспечивает возможность пользователю выбрать начальный баланс | 101 | Реализация ввода начального баланса |
| 2 | Система обеспечивает возможность пользователю делать ставку | 201 | Реализация ввода и сохранения ставки игрока |
| 202 | Проверка ставки (ставка не превышает баланса и больше нуля) |
| 3 | Система обеспечивает возможность пользователям сделать выбор | 301 | Реализация ввода и сохранения выбора игрока |
| 4 | Система обеспечивает возможность проведения раунда игры | 401 | Нахождение случайного числа |
| 402 | Проверка найденного числа на чётность |
| 403 | Сравнение выбора игрока с результатом проверки на чётность |
| 404 | Изменение баланса игрока |
| 405 | Вывод результата раунда |
| 5 | Система обеспечивает возможность закончить игру после завершения раунда | 501 | Продолжение игры (новый раунд) |
| 502 | Окончание игры |

Шаг 4. Спецификация требований к ПО.

1. Введение.
   1. Данная программа предназначена для развлечения.
   2. Данная программа может быть использована любым человеком.
2. Общее описание.
   1. Документацией для пользователей является таблица из шага 4, в которой указаны все возможные действия игрока и программы.
3. Функции системы.
   1. Функция «Выбор» позволяет игроку выбрать одну из двух значений: чётное или нечётное, а программа сохраняет выбор игрока и передаёт его в функцию «Проверка ставки».
   2. Функция «Нахождение случайного числа» генерирует случайное число в диапазоне от 1 до 100.
   3. Функция «Проверка на чётность» проверяет число, сгенерированное функцией «Нахождение случайного числа», на чётность и сохраняет результат.
   4. Функция «Проверка ставки» сравнивает выбор игрока в функции «Выбор» с результатом проверки в функции «Проверка на чётность».
   5. Функция «Конец раунда» осуществляет изменение баланса игрока и вывод результатов раунда. В случае нулевого баланса завершает игру.
   6. Функция «Конец игры?» предоставляет пользователю выбор: продолжить игру или выйти и забрать выигрыш.

Шаг 5. Диаграмма вариантов использования.

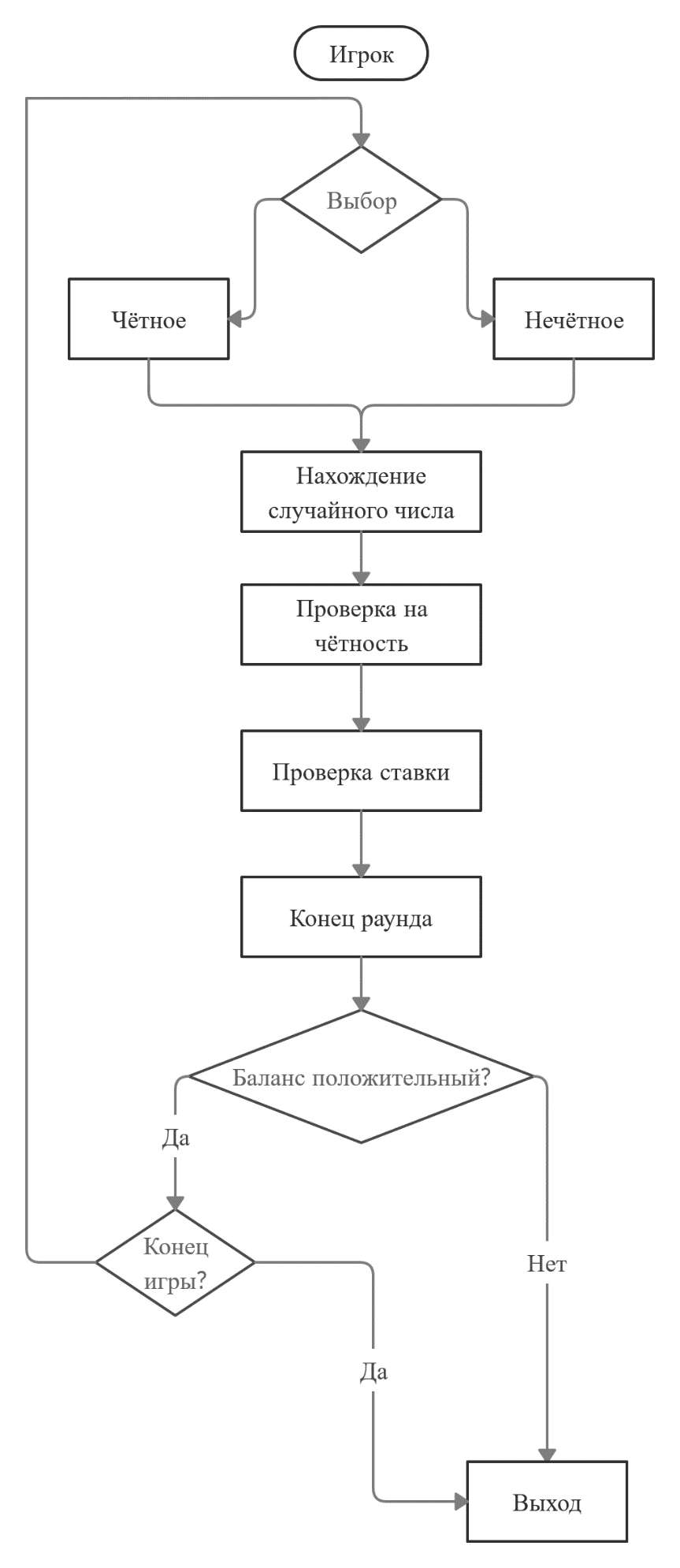


Рисунок 1 - Диаграмма вариантов использования

Шаг 6. Диаграмма взаимодействия, соответствующая потокам событий вариантов использования.

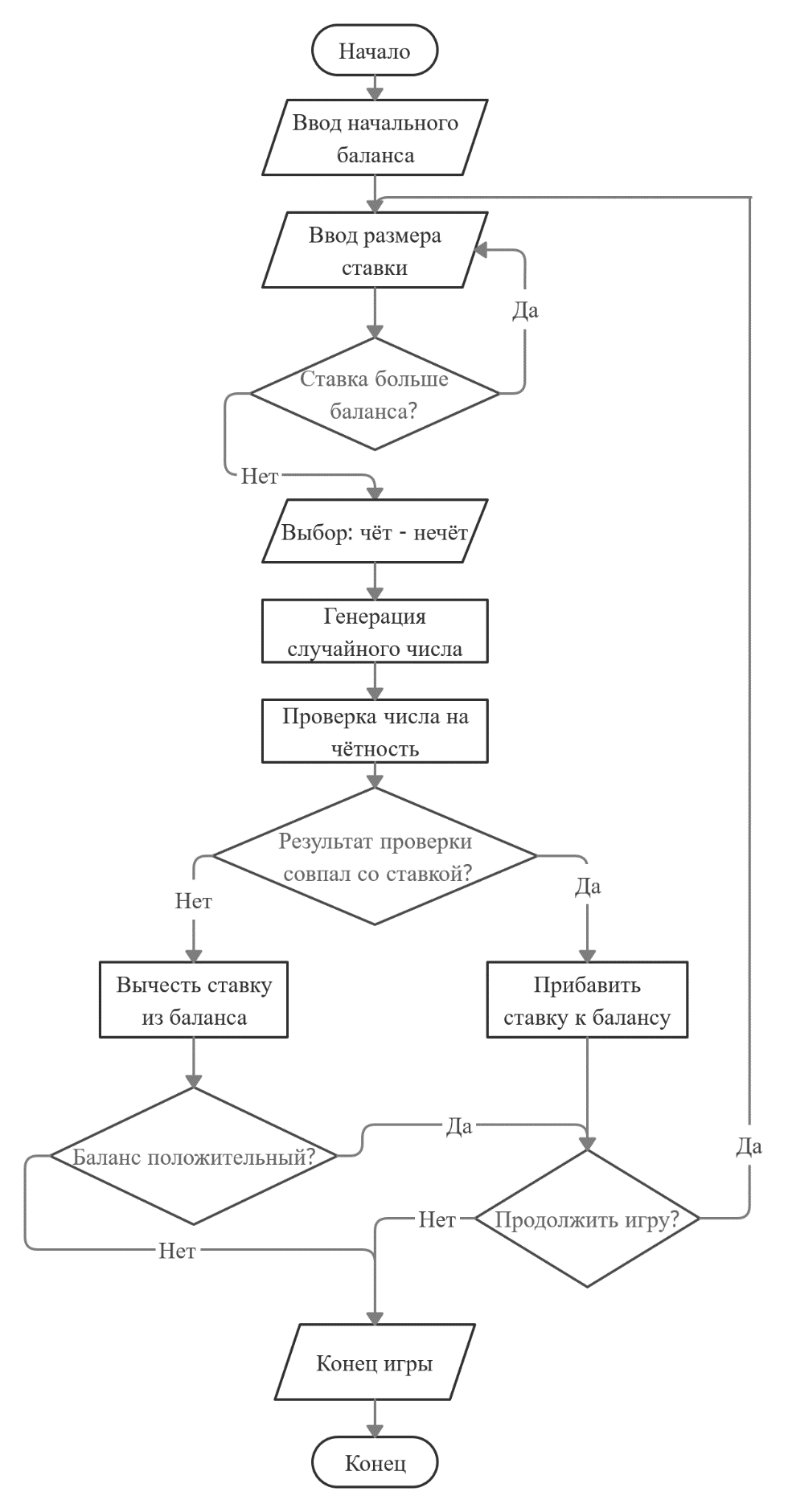


Рисунок 2 – Диаграмма взаимодействия

Шаг 7. Реализация функций.

Для ввода размера ставки реализована функция makebet. Функция считывает с консоли размер ставки, а также проверяет, является ли ставка положительной и не превышает ли ставка баланс.

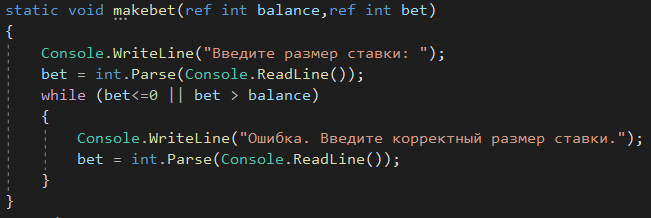


Рисунок 3 - Реализация функции makebet

Для ввода выбора игрока (четное или нечётное) реализована функция makechoice.

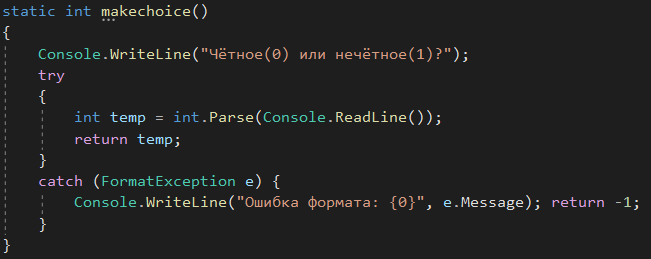


Рисунок 4 - Реализация функции makechoice

Для нахождения четности случайного числа реализована функция randomint. С помощью класса Random функция находит случайное число и возвращает остаток от деления этого числа на 2. Если число чётное – вернётся 0, если нечётное – 1.

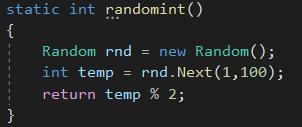


Рисунок 5 - Реализация функции randomint

Для проведения одного раунда игры реализована функция check. Функция проверяет, совпало ли предположение игрока с чётностью случайного числа, после чего изменяет баланс игрока. Если баланс становится неположительным, функция возвращает значение, завершающее игру.

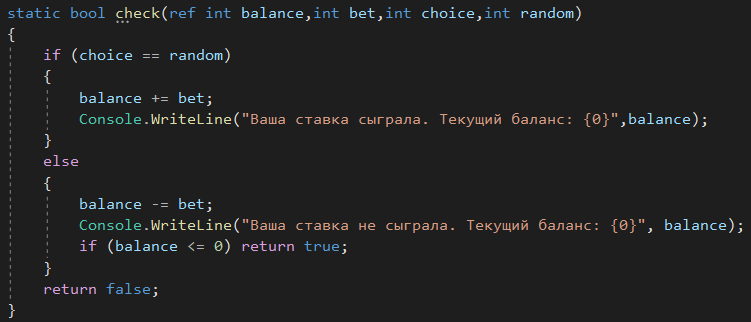


Рисунок 6 - Реализация функции check

В функции main происходит ввод стартового баланса игрока и взаимодействие всех вышеприведённых функций. Также в функции main реализован механизм продолжения или завершения игры после каждого раунда по желанию игрока, если его баланс положительный.



Рисунок 7 - реализация функции main

Результат работы игры приведён ниже.

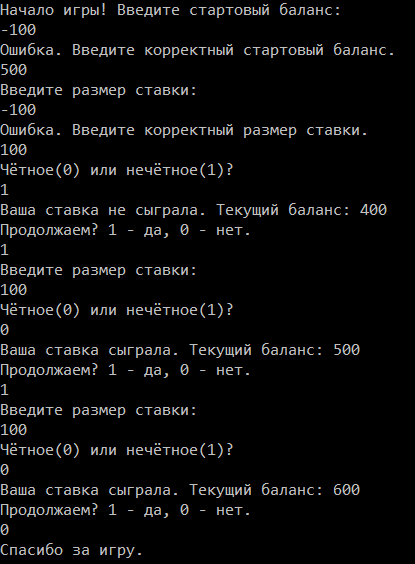


Рисунок 8 - Результат работы программы

**Вывод:** было описано приложение «Игра чётное-нечётное» согласно процедурной парадигме, созданы план разработки с перечнем требований и функционалом, таблица функционала приложения, а также диаграмма использования ПО. Конечным результатом выполнения практической работы является реализованное на языке C# приложение «Игра чётное-нечётное».