

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

Кафедра программных систем

**Практическая работа 4**

**Реализация объектно-ориентированной программы**

Выполнил: Мордвинцев Максим Витальевич

Группа № К3120

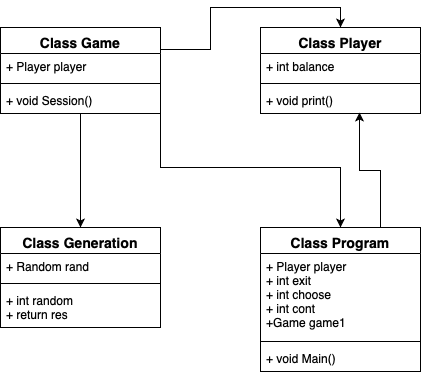
Проверила: Казанова П.П.

Цель работы:

Разработать диаграмму классов и программно реализовать иерархию классов

Диаграмма классов:

Диаграмма классов данного программного продукта представлена на рисунке 1.



*Рисунок 1 – диаграмма классов*

Программная реализация иерархии классов:

Класс Player

class Player

{  
  
        public int balance;  
  
        public Player(int balance)  
  
        {  
            this.balance = balance;  
        }  
        public int Balance()  
        {  
            return (this.balance);  
        }  
    }

Класс Game

class Game

{  
        public Player player;  
        public Game(ref Player player)  
        {  
            this.player = player;  
        }  
        public void Session()  
        {  
            Console.Write("\n@@@\nБаланс: {0}\n", player.balance);  
            Console.Write("Ваша ставка: ");  
            int stav = int.Parse(Console.ReadLine());  
            if (stav > player.balance)  
            {  
                Console.Write("Ошибка! Значение ставки не может превышать текущей суммы на счету\n");  
            }  
            else  
            {  
                Console.Write("\nСделайте выбор: \n0 - чет\n1 - нечет\n");  
                int choose = int.Parse(Console.ReadLine());  
                if ((choose != 0) & (choose != 1))  
                {  
                    Console.Write("Введено неверное значение\n");  
                }  
                else  
                {  
                    Generator generator = new Generator();  
                    int rand = generator.random();  
                    if (rand % 2 == choose)  
                    {  
                        player.balance += stav;  
                        Console.Write("\n---\nВы угадали!\nБаланс:{0}\n", player.balance);  
                        Console.Write("Число:{0}\n---\n", rand);  
                    }  
  
                    else  
                    {  
                        player.balance -= stav;  
                        Console.Write("\n---\nВы не угадали!\nБаланс:{0}\n", player.balance);  
                        Console.Write("Число:{0}\n---\n", rand);  
  
                    }  
                }  
            }

**Класс Generator**

{  
        Random rand;  
        public Generator()  
        {  
            rand = new Random();  
        }  
        public int random()  
        {  
            int res = rand.Next(1, 10);  
            return (res);  
        }  
    }

Класс Program

class Program

{  
            Console.Write("\n@@@Введите начальное значение баланса:\n");  
            int exit = 0;  
            Player Player = new Player(int.Parse(Console.ReadLine()));  
              
            do  
            {  
                Console.Write("\n@@@Главное меню@@@\nВарианты действий: \n0 - играть\n1 - запрос баланса \n2 - выйти\n");  
                Console.Write("\nВыберите действие: ");  
                int action = int.Parse(Console.ReadLine());  
                switch (action)  
                {  
                    case 0:  
                        int cont = 1;  
                        while ((Player.balance > 0) & (cont == 1))  
                        {  
                            Game game1 = new Game(ref Player);  
                            game1.Session();  
                            Console.Write("\nЖелаете продолжить игру? \n-0 - выйти в меню, \n-1 - продолжить, \n-2 - закончить игру\n");  
                            cont = int.Parse(Console.ReadLine());  
                            if ((cont != 0) & (cont != 1))  
                            {  
                                Console.Write("\n---\nВведено неверное значение\n---\n");  
                                break;  
                            }  
                        }  
                        if (Player.balance == 0) Console.Write("\nУ Вас больше нет средств!\n");  
                        exit = 1;  
                        break;  
                    case 1:  
                        Console.WriteLine(Player.Balance());  
                        Console.WriteLine("\n-0 - выход в главное меню, \n-1 - закончить игру\n");  
                        exit = int.Parse(Console.ReadLine());  
                        break;  
                    case 2:  
                        exit = 1;  
                        break;  
                    default:  
                        Console.WriteLine("\nВведите корректное значение\n");  
                        break;  
                }  
            }  
            while (exit != 1);  
  
  
        }

Вывод:

В ходе выполнения практической работы была составлена диаграмма используемых в данном программном продукте классов и программно реализована иерархия классов.

Результатом выполнения всех практической работы является рабочий прототип программы «чёт-нечёт»