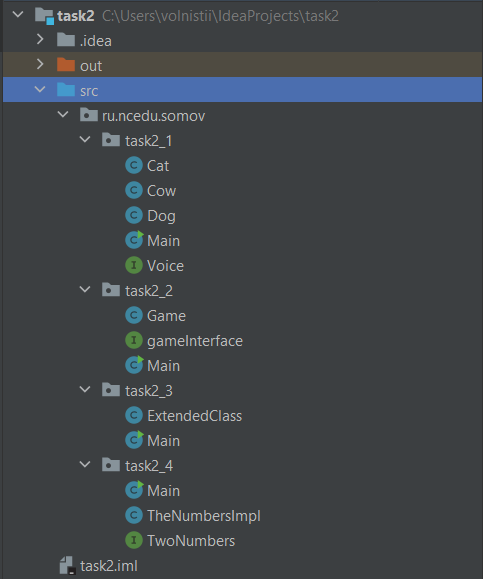
**Java Base Libraries**

**Цель работы:**

Сформировать навыки проектирования и реализации интерфейсов Java, закрепить знания в области разработки классов java и научиться переопределять методы eduals(), hashCode(), toString().

**Задание. Разработать программу с использованием интерфейсов и переопределить методы Java.**

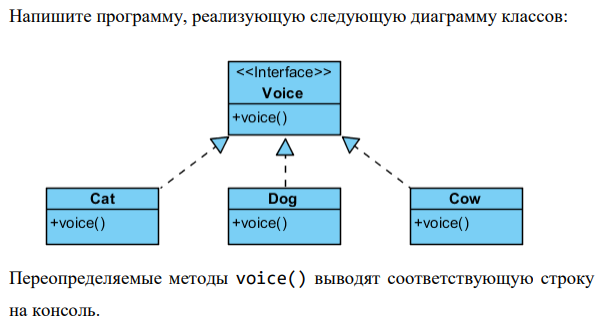
**Структура проекта:**



**Ссылка на github:**

<https://github.com/volnistii11/java/tree/codeReview>/task2

Задание 2.1.



Листинг задания 2.1

interface Voice

package ru.ncedu.somov.task2\_1;  
  
public interface Voice {  
 public void voice();  
}

class Cat

package ru.ncedu.somov.task2\_1;  
  
public class Cat implements Voice{  
 @Override  
 public void voice() {  
 System.*out*.println("Мяяяяяяяяяяяяяяяяяяяяяуууууууу");  
 }  
}

class Dog

package ru.ncedu.somov.task2\_1;  
  
public class Dog implements Voice{  
 @Override  
 public void voice() {  
 System.*out*.println("Гав-гав");  
 }  
}

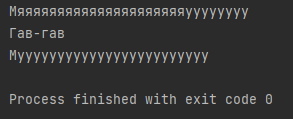
class Cow

package ru.ncedu.somov.task2\_1;  
  
public class Cow implements Voice{  
 @Override  
 public void voice() {  
 System.*out*.println("Муууууууууууууууууууууууу");  
 }  
}

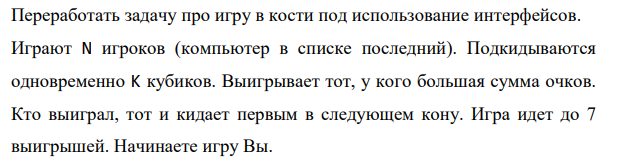
class Main (для проверки)

package ru.ncedu.somov.task2\_1;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 Voice voiceCat = new Cat();  
 voiceCat.voice();  
  
 Voice voiceDog = new Dog();  
 voiceDog.voice();  
  
 Voice voiceCow = new Cow();  
 voiceCow.voice();  
 }  
}

Демонстрация работы задания 2.1



Задание 2.2



Листинг задания 2.2

interface gameInterface

package ru.ncedu.somov.task2\_2;  
  
public interface gameInterface {  
 public void bonesGame(int n, int k);  
}

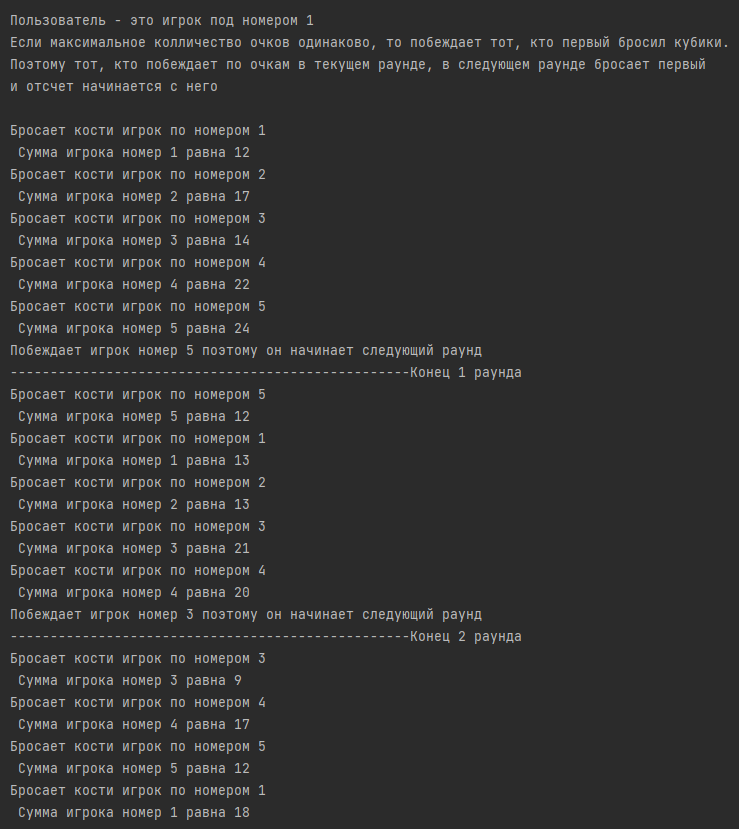
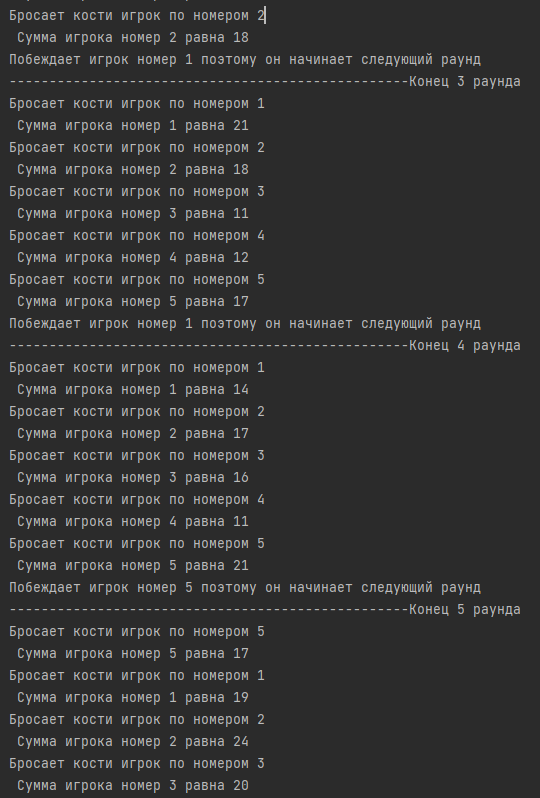
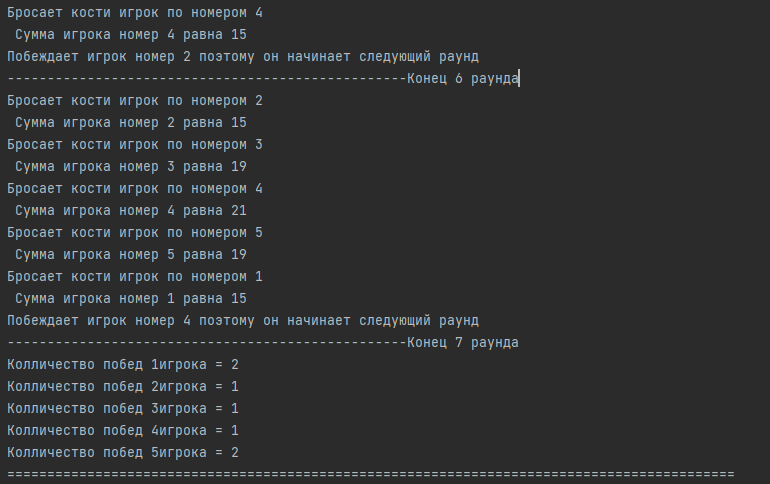
class Game

package ru.ncedu.somov.task2\_2;  
  
import java.util.Arrays;  
import java.util.Random;  
  
*/\*\*  
 \* В классе Game происходит реализация игры в кости.  
 \* Играют N игроков (компьютер в списке последний). Подкидываются одновременно К кубиков.  
 \* Выигрывает тот, у кого большая сумма очков.  
 \* Кто выиграл, тот и кидает первым в следующем кону(тот, кто был за ним, становится вторым и т.д.).  
 \* Игра идет до 7 выигрышей. Начинаете игру Вы (Пользователь под номером 1).  
 \*/*public class Game implements gameInterface {  
  
 */\*\*  
 \*  
 \** ***@param*** *n - число игроков  
 \** ***@param*** *k - колличество кубиков  
 \* В методе bonesGame происходит вся реализация игры в кости.  
 \* В players\_results[i][0] записывается результат, который выбил i-ый игрок, бросив k кубиков.  
 \* В players\_results[i][1] записывается колличество побед i-ого игрока за 7 раундов  
 \*/* public void bonesGame(int n, int k) {  
 int[][] players\_results = new int[n][2];  
 int round;  
 Random rnd = new Random(System.*currentTimeMillis*());  
  
 System.*out*.println("\nПользователь - это игрок под номером 1");  
 System.*out*.println("Если максимальное колличество очков одинаково, то побеждает тот, кто первый бросил кубики. "  
 + "\nПоэтому тот, кто побеждает по очкам в текущем раунде, в следующем раунде бросает первый "  
 + "\nи отсчет начинается с него\n");  
 int winner = 0;  
 for (round = 0; round < 7; round++) {  
  
  
 int z = 0;  
  
  
 for (int i = winner; i < n; i++) {  
 System.*out*.println("Бросает кости игрок по номером " + (i + 1));  
 players\_results[i][0] = 0;  
 int j = 0;  
 while (j < k) {  
 players\_results[i][0] = players\_results[i][0] + 1 + rnd.nextInt(6);  
 j++;  
 }  
 System.*out*.println(" Сумма игрока номер " + (i + 1) + " равна " + players\_results[i][0]);  
 if (i == (n - 1)) {  
 i = -1;  
 }  
 if (z == (n - 1)) break;  
 z++;  
 }  
  
 int max = players\_results[0][0];  
 int max\_index = 0;  
  
 for (int i = 1; i < n; i++) {  
 if (players\_results[i][0] > max) {  
 max = players\_results[i][0];  
 max\_index = i;  
 }  
 }  
 players\_results[max\_index][1]++;  
 System.*out*.println("Побеждает игрок номер " + (max\_index + 1) + " поэтому он начинает следующий раунд");  
 System.*out*.println("--------------------------------------------------Конец " + (round + 1) + " раунда");  
 winner = max\_index;  
  
 }  
 System.*out*.println("Колличество побед " + (1) + "игрока = " + players\_results[0][1]);  
 System.*out*.println("Колличество побед " + (2) + "игрока = " + players\_results[1][1]);  
 System.*out*.println("Колличество побед " + (3) + "игрока = " + players\_results[2][1]);  
 System.*out*.println("Колличество побед " + (4) + "игрока = " + players\_results[3][1]);  
 System.*out*.println("Колличество побед " + (5) + "игрока = " + players\_results[4][1]);  
 }  
  
}

class Main (для проверки)

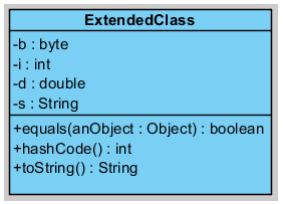
package ru.ncedu.somov.task2\_2;  
  
import java.text.ParseException;  
import java.util.Arrays;  
import java.util.Date;  
  
*/\*\*  
 \* Основной класс Main, где происходит тестирование остальных классов  
 \*/*public class Main {  
  
 public static void main(String[] args) throws ParseException {  
  
 gameInterface game = new Game();  
 game.bonesGame(5,5);  
 System.*out*.println("===========================================================================================");  
  
 }  
}

Демонстрация работы задания 2.2

Задание 2.3

Напишите программу, реализующую изображенный класс:



Листинг задания 2.3

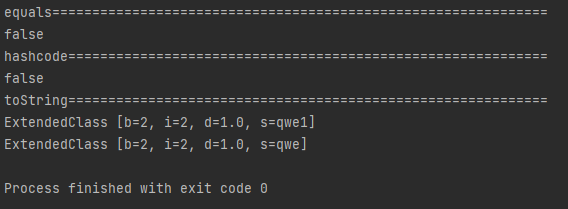
class ExtendedClass

package ru.ncedu.somov.task2\_3;  
  
import java.util.Objects;  
  
public class ExtendedClass {  
 private byte b;  
 private int i;  
 private double d;  
 private String s;  
  
 public ExtendedClass(byte b, int i, double d, String s) {  
 this.b = b;  
 this.i = i;  
 this.d = d;  
 this.s = s;  
 }  
  
  
 @Override  
 public boolean equals(Object obj) {  
 if (obj == this) {  
 return true;  
 }  
  
 if (obj == null || obj.getClass() != this.getClass()) {  
 return false;  
 }  
  
 ExtendedClass guest = (ExtendedClass) obj;  
  
 return Objects.*equals*(b, guest.b) && Objects.*equals*(i, guest.i) && Objects.*equals*(d, guest.d) && Objects.*equals*(s, guest.s);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(s,d,i,b);  
 }  
  
 @Override  
 public String toString(){  
 return "ExtendedClass [b=" + b  
 + ", i=" + i  
 + ", d=" + d  
 + ", s=" + s + "]";  
 }  
  
  
}

class Main (для проверки)

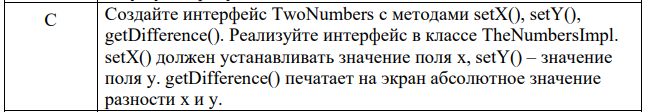
package ru.ncedu.somov.task2\_3;  
  
import ru.ncedu.somov.task2\_2.Game;  
import ru.ncedu.somov.task2\_2.gameInterface;  
  
import java.text.ParseException;  
  
*/\*\*  
 \* Основной класс Main, где происходит тестирование остальных классов  
 \*/*public class Main {  
  
 public static void main(String[] args) throws ParseException {  
  
  
 System.*out*.println("equals==============================================================");  
 ExtendedClass a = new ExtendedClass((byte) 2,2,1,"qwe1");  
 ExtendedClass b = new ExtendedClass((byte) 2,2,1,"qwe");  
 System.*out*.println(a.equals(b));  
  
  
 System.*out*.println("hashcode============================================================");  
 if (a.hashCode() == b.hashCode()) {  
 System.*out*.println(true);  
 } else {  
 System.*out*.println(false);  
 }  
  
  
 System.*out*.println("toString============================================================");  
 System.*out*.println(a.toString());  
  
 }  
}

Демонстрация работы задания 2.3



Задание 2.4

Выполните задание по варианту. Выбор осуществляется по первой букве фамилии. Классы, реализуемые в задании, не имеют метода main() и перед использованием должны быть инстанциированы (должен быть создан объект класса) в методе main() основного класса.



Листинг задания 2.4

interface TwoNumbers

package ru.ncedu.somov.task2\_4;  
  
*/\*\*  
 \* Создайте интерфейс TwoNumbers с методами setX(), setY(),  
 \* getDifference(). Реализуйте интерфейс в классе TheNumbersImpl.  
 \* setX() должен устанавливать значение поля x, setY() – значение  
 \* поля y. getDifference() печатает на экран абсолютное значение  
 \* разности x и y.  
 \*/*public interface TwoNumbers {  
 public void setX(double x);  
 public void setY(double y);  
 public double getDifference();  
}

class TheNumbersImpl

package ru.ncedu.somov.task2\_4;  
public class TheNumbersImpl implements TwoNumbers{  
 private double x;  
 private double y;  
  
  
 @Override  
 public void setX(double x) {  
 this.x = x;  
 }  
  
 @Override  
 public void setY(double y) {  
 this.y = y;  
 }  
  
 @Override  
 public double getDifference() {  
 return Math.*abs*(x-y);  
 }  
}

class Main (для проверки)

package ru.ncedu.somov.task2\_4;  
  
import ru.ncedu.somov.task2\_3.ExtendedClass;  
  
import java.text.ParseException;  
  
*/\*\*  
 \* Основной класс Main, где происходит тестирование остальных классов  
 \*/*public class Main {  
  
 public static void main(String[] args) throws ParseException {  
 TwoNumbers xy = new TheNumbersImpl();  
  
 xy.setX(10.2231);  
 xy.setY(10.3311);  
  
 System.*out*.println(xy.getDifference());  
  
 }  
}

Демонстрация работы задания 2.4

