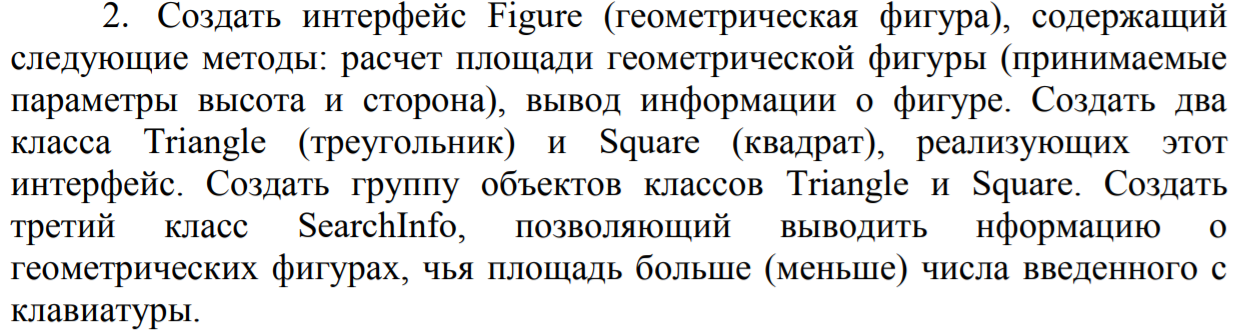
**Индивидульная практическая работа № 2. «Наследование и скрытие данных. Абстрактные классы и интерфейсы»**



**Текст программы:**

**package** com.main;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** com.models.\*;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Scanner in = **new** Scanner(System.***in***);

System.***out***.print("Input value: ");

**int** value = in.nextInt();

ArrayList<Figure> figures = **new** ArrayList<Figure>();

Figure square = **new** Square();

Figure triangle = **new** Triangle();

figures.add(square);

figures.add(triangle);

System.***out***.println("Less: ");

**for** (Figure figure : figures)

SearchInfo.*findBySquareLess*(figure, 5, 5, value);

System.***out***.println("Lager: ");

**for** (Figure figure : figures)

SearchInfo.*findBySquareLarger*(figure, 5, 5, value);

}

}

**package** com.main;

**import** com.models.Figure;

**public** **class** SearchInfo {

**static** **public** **void** findBySquareLess(Figure figure, **double** altitude, **double** side, **double** value) {

**double** square = figure.calcSquare(altitude, side);

**if** (square < value) {

figure.printFigureInfo();

// Hashcode - unique number (nearly)

System.out.println(String.format("Detected. Expected < %s. Gotten value: %s. Hashcode: %d", value, square, figure.hashCode()));

}

}

**static** **public** **void** findBySquareLarger(Figure figure, **double** altitude, **double** side, **double** value) {

**double** square = figure.calcSquare(altitude, side);

**if** (square > value) {

figure.printFigureInfo();

// Hashcode - unique number (nearly)

System.out.println(String.format("Detected. Expected > %s. Gotten value: %s. Hashcode: %d", value, square, figure.hashCode()));

}

}

}

**package** com.models;

**public** **interface** Figure {

**double** calcSquare(**double** altitude, **double** side);

**void** printFigureInfo();

}

**package** com.models;

**public** **class** Square **implements** Figure {

@Override

**public** **double** calcSquare(**double** altitude, **double** side) {

**if** (altitude == side && (altitude > 0 && side > 0))

**return** altitude \* side \* 2;

**else**

**return** -1;

}

@Override

**public** **void** printFigureInfo() {

System.***out***.println("This is square");

}

}

**package** com.models;

**public** **class** Triangle **implements** Figure {

@Override

**public** **double** calcSquare(**double** altitude, **double** side) {

**if** (altitude > 0 && side > 0)

**return** 0.5 \* altitude \* side;

**else**

**return** -1;

}

@Override

**public** **void** printFigureInfo() {

System.***out***.println("This is triangle");

}

}

**Результат выполнения:**

