

Московский Государственный Технический Университет
имени Н.Э. Баумана

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

Отчет к лабораторной работе №2
по курсу «Технологии программирования»
“Углубленное ознакомление работы классов в языке программирования Java”

Продолжительность работы: 4 ак. часа

Сдал

Ситниченко С. А. ИУ3-42Б

Принял

Руденкова Ю.С.

Лабораторная №2

Тема: Углубленное ознакомление работы классов в языке программирования Java

Цель работы: ознакомиться с реализацией классов на Java и научиться работать с ними. Написать программу, использующую ООП для решения поставленной задачи.

Класс Point3D

```
import java.util.Scanner;
public class Point3D {
    double x, y, z;

    Point3D() {
        x = 0;
        y = 0;
        z = 0;
    }

    Point3D(double user_x, double user_y, double user_z) {
        x = user_x;
        y = user_y;
        z = user_z;
    }

    Point3D(Scanner scan) {
        x = scan.nextDouble();
        y = scan.nextDouble();
        z = scan.nextDouble();
    }

    boolean equals(Point3D point) {
        return (x == point.x) && (y == point.y) && (z == point.z);
    }

    double distanceTo(Point3D point) {
        return Math.sqrt(Math.pow(x - point.x, 2) + Math.pow(y - point.y, 2) + Math.pow(z - point.z, 2));
    }
}
```

Рис. 1 Класс точки Point3D

В классе хранится три координаты точки в трёхмерном пространстве. Реализованы три конструктора – конструктор по умолчанию, создающий точку (0;0;0), конструктор, принимающий три числа и создающий точку с соответствующими координатами, а также конструктор, принимающий в

качестве параметра сканнер, чтобы упростить ввод координат с клавиатуры. Также реализованы методы сравнения точки с другой точкой на равенство значений координат и вычисления расстояния от одной точки до другой по прямой.

Класс Lab1

```
static double computeArea(Point3D point1, Point3D point2, Point3D point3) {  
    double AB = point1.distanceTo(point2);  
    double BC = point2.distanceTo(point3);  
    double CA = point3.distanceTo(point1);  
    double p = (AB + BC + CA) / 2;  
    return Math.sqrt(p * (p - AB) * (p - BC) * (p - CA));  
}
```

Рис. 2 Вычисление площади

Вычисление площади треугольника выполняется по формуле Герона с использованием методов Point3D.

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.println("First point: ");  
    Point3D A = new Point3D(scanner);  
    System.out.println("Second point: ");  
    Point3D B = new Point3D(scanner);  
    System.out.println("Third point: ");  
    Point3D C = new Point3D(scanner);  
  
    if ((A.equals(B)) && (A.equals(C))) {  
        System.out.println("Points are EQUAL. Can't calculate area.");  
        return;  
    }  
  
    double sqrt = computeArea(A, B, C);  
    System.out.println("Computed area: " + sqrt);  
}
```

Рис. 3 Main

Реализация пользовательского ввода, проверка на равенство точек, и вычисление площади треугольника, образованной пользовательскими точками.

Output

```
First point:
1 3 5
Second point:
2 4 6
Third point:
-1 7 4
Computed area: 3.9370039370059087

Process finished with exit code 0
```

Расчет площади треугольника по координатам вершин

| | | |
|----------|---------|---------|
| Ax 1 | Ay 3 | Az 5 |
| Bx 2 | By 4 | Bz 6 |
| Cx -1 | Cy 7 | Cz 4 |

Точность вычисления
Знаков после запятой: 2

РАССЧИТАТЬ

Площадь треугольника
3.94

Периметр треугольника
11.01

ССЫЛКА

СОХРАНИТЬ

ВИДЖЕТ

```
First point:
1 1 1
Second point:
1 1 1
Third point:
1 1 1
Points are EQUAL. Couldn't calculate area.

Process finished with exit code 0
```

Вывод:

Была разработана программа, где активно использовал ООП для решения поставленной задачи, а именно - работы с точками в трёхмерном пространстве: методы задания точки, вычисления расстояния между двумя точками и вычисления площади треугольника, образованной тремя точками, а также пользовательский интерфейс.