Федеральное агентство связи

Ордена Трудового Красного знамени

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчет по лабораторной работе №2

по дисциплине «Объектно ориентированное программирование»

на тему «Основы объектно-ориентированного программирования»

Выполнил

Студент группы БФИ1701

Неретин И.Г.

Проверила:

Мосеева М.С.

Москва 2020

1. **Цель работы**

Цель работы – получить навыки работы с объектно-ориентированным подходом в программировании на языке Java.

1. **Задание**

Цель работы определила следующие задачи:

1. Создать класс трехмерной точки, унаследованный от имеющегося класса двумерной точки;
2. Переопределить методы сравнения точек;
3. Определить конструкторы инициализации и конструкторы по умолчанию для класса трехмерной точки;
4. Определить для этого класса интерфейсы взаимодействия (get и set методы) и изолировать данные от внешнего воздействия;
5. Создать метод distanceTo для класса трехмерной точки, который будет определять расстояние от одной точки до другой с точностью до двух знаков после запятой;
6. Создать метод computeArea главного класса, который будет определять площадь треугольника по трем точкам. При этом он должен обрабатывать ситуацию, когда две или более точки одинаковы, и выводить соответствующее сообщение;
7. Создать метод main, выполняющий обработку входных данных программы и ее выполнение;
8. Скомпилировать главный класс программы и класс трехмерной точки, и протестировать программу с разными входными данными.
9. **Порядок выполнения работы**

**Исходный код**

Листинг 1 – Point2d.java

|  |
| --- |
| */\*\*  \* двумерный класс точки.  \*\*/* public class Point2d {  */\*\* координата X \*\*/* private double xCoord;  */\*\* координата Y \*\*/* private double yCoord;  */\*\* Конструктор инициализации \*\*/* public Point2d (double x, double y) {  xCoord = x;  yCoord = y;  }  */\*\* Конструктор по умолчанию. \*\*/* public Point2d () {  //Вызовите конструктор с двумя параметрами и определите источник.  this(0, 0);  }  */\*\* Возвращение координаты X \*\*/* public double getX () {  return xCoord;  }  */\*\* Возвращение координаты Y \*\*/* public double getY () {  return yCoord;  }  */\*\* Установка значения координаты X. \*\*/* public void setX (double val) {  xCoord = val;  }  */\*\* Установка значения координаты Y. \*\*/* public void setY (double val) {  yCoord = val;  } } |

Листинг 2 – Point3d.java

|  |
| --- |
| */\*\*Класс трехмерной точки\*\*/* public class Point3d extends Point2d {  */\*\* координата Z \*\*/* private double zCoord;  */\*\* Конструктор инициализации \*\*/* public Point3d (double x, double y, double z) {  super(x, y);  zCoord = z;  }  */\*\* Конструктор преобразования \*\*/* public Point3d (Point2d point2d) {  super(point2d.getX(), point2d.getY());  zCoord = 0;  }  */\*\* Конструктор по умолчанию. \*\*/* public Point3d () {  this(0, 0, 0);  }  */\*\* Возвращение координаты Z \*\*/* public double getZ () {  return zCoord;  }  */\*\* Установка значения координаты Z \*\*/* public void setZ (double val) {  zCoord = val;  }  */\*\* Сравнение двух точек в трехмерном пространстве\*\*/* public boolean equals(Point3d AnotherPoint){  return (super.getX() == AnotherPoint.getX()) &&  (super.getY() == AnotherPoint.getY()) &&  (this.getZ() == AnotherPoint.getZ());  }  */\*\* Расстояние между двумя точками в трехмерном пространстве с точностью до  \* двух знаков после запятой \*\*/* public double distanceTo(Point3d point){  return (double) Math.*round*(Math.*sqrt*(  Math.*pow*(point.getX() - this.getX(),2)  + Math.*pow*(point.getY() - this.getY(),2)  + Math.*pow*(point.getZ() - this.getZ(),2)  ) \* 100) / 100;  } } |

Листинг 3 – Lab2Classes.java

|  |
| --- |
| */\*\*  \* Программа "Треугольники" принимает на вход череду вещественных чисел через пробел,  \* из которых формирует точки по трем координатам.  \*  \* Первые три числа распознаются, как координаты X, Y, Z первой точки, следующие  \* 3 числа - координаты второй точки и т.п.  \*  \* 3 точки (или 9 чисел) формируют треугольник. Программа вычисляет и выводит  \* площадь треугольника с некоторой погрешностью, или выводит сообщение, если  \* невозможно вычислить площадь.  \*  \* Процедура повторяется для каждого полного треугольника (формирующих его 9 чисел).  \* Лишние числа, не формирующие точки или треугольники, не учитываются.  \* \*\*/* public class Lab2Classes {  */\*\* Входная точка программы \*\*/* public static void main(String[] args){  Point3d[] pointsArray = *formPointsArray*(args);  *outputAllPoints*(pointsArray);  if (pointsArray.length < 3){  System.*out*.println("Not enough points. You need 3 points (9 coordinates) at least");  return;  }   double answer;  int numberOfTriangle;  for (int i = 0; i < pointsArray.length / 3; i+=1){  answer = *computeArea*(pointsArray[i\*3],pointsArray[(i\*3)+1],pointsArray[(i\*3)+2]);  numberOfTriangle = i;  System.*out*.print("Triangle no." + numberOfTriangle + ": ");  if (answer != 0){  System.*out*.println("area = " + answer);  }  else{  System.*out*.println("Cannot calculate area");  }  }  }   */\*\* Функция, вычисляющая площадь треугольника по трем точкам \*\*/* public static double computeArea(Point3d p1, Point3d p2, Point3d p3){  //если какие-то из точек равны, то нет необходимости вычислять площадь  if (p1.equals(p2) || p2.equals(p3) || p3.equals(p1)){  System.*out*.println("Two or more points are identical. Calculation was cancelled");  return 0;  }  //стороны треугольника  double a = p1.distanceTo(p2);  double b = p2.distanceTo(p3);  double c = p3.distanceTo(p1);  double p = (a + b + c) / 2;  return Math.*sqrt*(p\*(p-a)\*(p-b)\*(p-c));  }   */\*\* Вывод информации о точке на экран \*\*/* public static void outputPoint(Point3d point, String name){  System.*out*.println("Point " + name + ": X = "  + point.getX() + "; Y = " + point.getY() + "; Z = " + point.getZ());  }   */\*\* Функция вывода информации обо всех точках из массива точек \*\*/* public static void outputAllPoints(Point3d[] points){  char[] alphabet = new char[] {'A','B','C'};  String nameOfPoint;  System.*out*.println("Created points: ");  for (int i = 0; i < points.length; i++){  if (i < alphabet.length){  nameOfPoint = alphabet[i] + "";  }  else{  int numberOfPoint = i / alphabet.length;  nameOfPoint = alphabet[i % alphabet.length] + Integer.*toString*(numberOfPoint);  }  *outputPoint*(points[i],nameOfPoint);  }  System.*out*.println("End");  }   */\*\* Функция создания массива точек из поступивших на вход значений \*\*/* public static Point3d[] formPointsArray(String[] args){  //инициализация  int i;  int pointsCount = 0;  int pointArrayLength = (args.length - (args.length % 3)) / 3; //сколько наборов из трех координат получится  int length = args.length - (args.length % 3); //количество чисел, составляющих полный набор для точки  //массив из трех чисел, в который будут переписываться координаты X, Y, Z  double[] coords = new double[3];  Point3d[] points = new Point3d[pointArrayLength]; //массив точек   //создание точек из имеющихся наборов из трех чисел  for (i = 0; i < length; i++){  coords[i % 3] = Double.*parseDouble*(args[i]);  if ((i+1) % 3 == 0){  //создаем новую точку  points[pointsCount] = new Point3d(coords[0],coords[1],coords[2]);  pointsCount++;  }  }   System.*out*.println(pointsCount + " points was created");  return points;  } } |

**Результат работы**



Рисунок 1 – Компиляция двух исходных классов

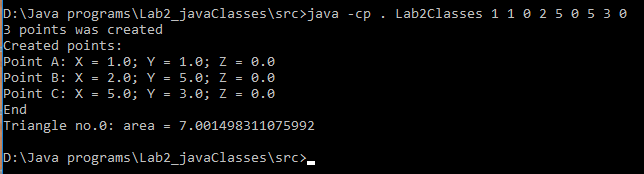


Рисунок 2 – Результат работы с девятью значениями, сформировавшими треугольник

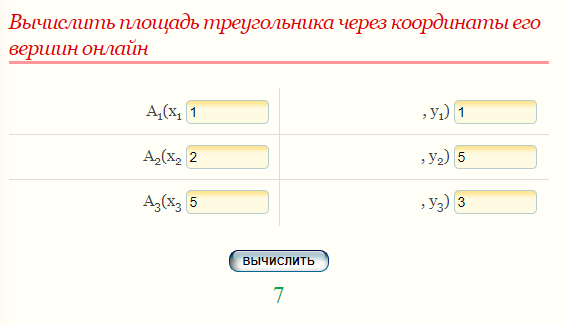


Рисунок 3 – Проверка на корректность работы

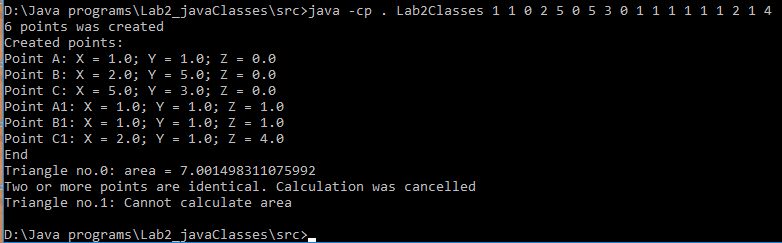


Рисунок 4 – Пример с двумя треугольниками, в одном из которых две точки совпадают

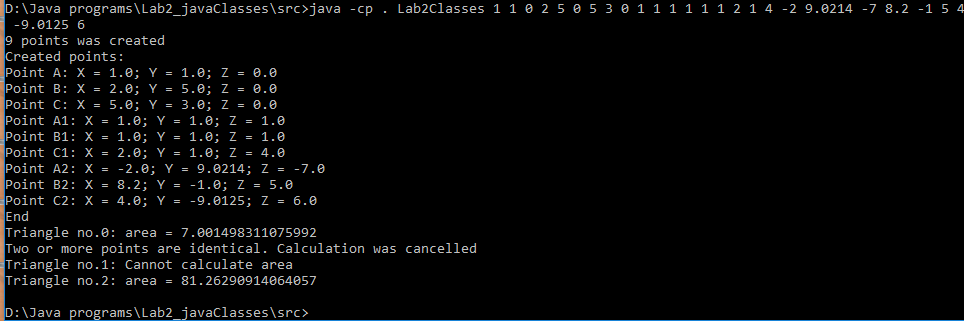


Рисунок 5 – Пример с вещественными числами и тремя треугольниками

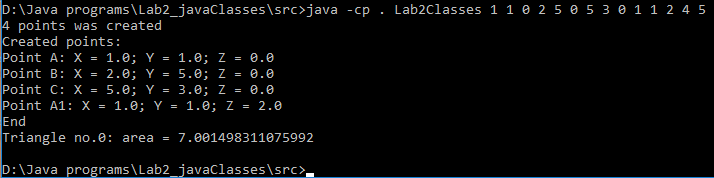


Рисунок 6 – Пример, в котором лишние числа и лишние точки были проигнорированы

1. **Заключение**

Объектно-ориентированный подход позволяет упростить процесс и сократить время разработки при помощи различных инструментов.

1. **Литература**
2. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа, 2006;
3. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004;
4. <https://www.google.ru/>