System pomiarowy

M.M. & M.P. (Java)

Otrzymany kod programu

```java
public class EuclideanDistance {
  private static double p1x;
  private static double p1y;
  private static double p2x;
  private static double p2y;
  private static double p3x;
  private static double p3y;
  private static double pow_1_2x;
  private static double pow_1_2y;
  private static double pow_1_3x;
  private static double pow_1_3y;
  private static double pow_2_3x;
  private static double pow_2_3y;
  private static double d_1_2;
  private static double d_1_3;
  private static double d_2_3;

  public static void main(String[] args) {
    p1x = 5.21;
    p1y = -7.32;
    p2x = -0.13;
    p2y = 3.14;
    p3x = -4.71;
    p3y = -8.41;
    pow_1_2x = Math.pow(p1x - p2x, 2);
    pow_1_2y = Math.pow(p1y - p2y, 2);
    pow_1_3x = Math.pow(p1x - p3x, 2);
    pow_1_3y = Math.pow(p1y - p3y, 2);
    pow_2_3x = Math.pow(p2x - p3x, 2);
    pow_2_3y = Math.pow(p2y - p3y, 2);
    d_1_2 = Math.sqrt(pow_1_2x + pow_1_2y);
    d_1_3 = Math.sqrt(pow_1_3x + pow_1_3y);
    d_2_3 = Math.sqrt(pow_2_3x + pow_2_3y);
    System.out.println("Euclidean Distance rounded to 2 decimal places.");
    System.out.println("P1 = (" + p1x + ", " + p1y + ")");
    System.out.println("P2 = (" + p2x + ", " + p2y + ")");
    System.out.printf("Distance between P1 and P2: %.2f\n", d_1_2);
    System.out.println("P1 = (" + p1x + ", " + p1y + ")");
    System.out.println("P2 = (" + p3x + ", " + p3y + ")");
    System.out.printf("Distance between P1 and P2: %.2f\n", d_1_3);
    System.out.println("P1 = (" + p2x + ", " + p2y + ")");
    Run | Debug
    System.out.println("P2 = (" + p3x + ", " + p3y + ")");
    System.out.printf("Distance between P1 and P2: %.2f\n", d_2_3);
  }
}
```

Co mamy do zrobienia?

1. Ułatwienie czytelności kodu
2. Zastosowanie zasad programowania
3. Wyławienie zapachów w kodzie
4. Naprawa (leczenie) kodu

Co pzreszkadza?

```
18    public static void main(String[] args) {
19      p1x = 5.21;
20      p1y = -7.32;
21      p2x = -0.13;
22      p2y = 3.14;
23      p3x = -4.71;
24      p3y = -8.41;
25      pow_1_2x = Math.pow(p1x - p2x, 2);
26      pow_1_2y = Math.pow(p1y - p2y, 2);
27      pow_1_3x = Math.pow(p1x - p3x, 2);
28      pow_1_3y = Math.pow(p1y - p3y, 2);
29      pow_2_3x = Math.pow(p2x - p3x, 2);
30      pow_2_3y = Math.pow(p2y - p3y, 2);
31      d_1_2 = Math.sqrt(pow_1_2x + pow_1_2y);
32      d_1_3 = Math.sqrt(pow_1_3x + pow_1_3y);
33      d_2_3 = Math.sqrt(pow_2_3x + pow_2_3y);
34      System.out.println("Euclidean Distance rounded to 2 decimal places.");
35      System.out.println("P1 = (" + p1x + ", " + p1y + ")");
36      System.out.println("P2 = (" + p2x + ", " + p2y + ")");
37      System.out.printf("Distance between P1 and P2: %.2f\n", d_1_2);
38      System.out.println("P1 = (" + p1x + ", " + p1y + ")");
39      System.out.println("P2 = (" + p3x + ", " + p3y + ")");
40      System.out.printf("Distance between P1 and P2: %.2f\n", d_1_3);
41      System.out.println("P1 = (" + p2x + ", " + p2y + ")");
42      System.out.println("P2 = (" + p3x + ", " + p3y + ")");
43      System.out.printf("Distance between P1 and P2: %.2f\n", d_2_3);
44    }
```

Zmienne

Otrzymany kod programu

```
18    public static void main(String[] args) {
19      point1_x = 5.21;
20      point1_y = -7.32;
21      point2_x = -0.13;
22      point2_y = 3.14;
23      point3_x = -4.71;
24      point3_y = -8.41;
25      power_subtracted1and2_x = Math.pow(point1_x - point2_x, 2);
26      power_subtracted1and2_y = Math.pow(point1_y - point2_y, 2);
27      power_subtracted1and3_x = Math.pow(point1_x - point3_x, 2);
28      power_subtracted1and3_y = Math.pow(point1_y - point3_y, 2);
29      power_subtracted2and3_x = Math.pow(point2_x - point3_x, 2);
30      power_subtracted2and3_y = Math.pow(point2_y - point3_y, 2);
31      distance_points1and2 = Math.sqrt(power_subtracted1and2_x + power_subtracted1and2_y);
32      distance_points1and3 = Math.sqrt(power_subtracted1and3_x + power_subtracted1and3_y);
33      distance_points2and3 = Math.sqrt(power_subtracted2and3_x + power_subtracted2and3_y);
34      System.out.println("Euclidean Distance rounded to 2 decimal places.");
35      System.out.println("P1 = (" + point1_x + ", " + point1_y + ")");
36      System.out.println("P2 = (" + point2_x + ", " + point2_y + ")");
37      System.out.printf("Distance between P1 and P2: %.2f\n", distance_points1and2);
38      System.out.println("P1 = (" + point1_x + ", " + point1_y + ")");
39      System.out.println("P2 = (" + point3_x + ", " + point3_y + ")");
40      System.out.printf("Distance between P1 and P2: %.2f\n", distance_points1and3);
41      System.out.println("P1 = (" + point3_x + ", " + point3_y + ")");
42      System.out.println("P2 = (" + point2_x + ", " + point2_y + ")");
43      System.out.printf("Distance between P1 and P2: %.2f\n", distance_points2and3);
44    }
```

Jakie zasady programowania stosujemy?

SOLID

- S - single responsibility
- O - open/closed
- L - liskov substitution
- I - interface segregation
- D - dependency inversion

Single responsibility (1)

```
18    public static void calculateDistance() {
19      power_subtracted1and2_x = Math.pow(point1_x - point2_x, 2);
20      power_subtracted1and2_y = Math.pow(point1_y - point2_y, 2);
21      power_subtracted1and3_x = Math.pow(point1_x - point3_x, 2);
22      power_subtracted1and3_y = Math.pow(point1_y - point3_y, 2);
23      power_subtracted2and3_x = Math.pow(point2_x - point3_x, 2);
24      power_subtracted2and3_y = Math.pow(point2_y - point3_y, 2);
25      distance_points1and2 = Math.sqrt(power_subtracted1and2_x + power_subtracted1and2_y);
26      distance_points1and3 = Math.sqrt(power_subtracted1and3_x + power_subtracted1and3_y);
27      distance_points2and3 = Math.sqrt(power_subtracted2and3_x + power_subtracted2and3_y);
28    }
29
30    public static void showPoints() {
31      System.out.println("P1 = (" + point1_x + ", " + point1_y + ")");
32      System.out.println("P2 = (" + point2_x + ", " + point2_y + ")");
33      System.out.println("P3 = (" + point3_x + ", " + point3_y + ")");
34    }
35
36    public static void printResult() {
37      System.out.printf("Distance between P1 and P2: %.2f\n", distance_points1and2);
38      System.out.printf("Distance between P1 and P3: %.2f\n", distance_points1and3);
39      System.out.printf("Distance between P2 and P3: %.2f\n", distance_points2and3);
40    }
```

```
42      public static void main(String[] args) {
43        point1_x = 5.21;
44        point1_y = -7.32;
45        point2_x = -0.13;
46        point2_y = 3.14;
47        point3_x = -4.71;
48        point3_y = -8.41;
49        System.out.println("Euclidean Distance rounded to 2 decimal places.");
50        showPoints();
51        calculateDistance();
52        printResult();
53      }
```

Single responsibility (2)

```java
    public static double calculateDistance(double p1x, double p1y, double p2x, double p2y) {
      double power_subtracted1and2_x = Math.pow(p1x - p2x, 2);
      double power_subtracted1and2_y = Math.pow(p1y - p2y, 2);
      double distance_points1and2 = Math.sqrt(power_subtracted1and2_x + power_subtracted1and2_y);
      return distance_points1and2;
    }

    public static void showPoints() {
      System.out.println("P1 = (" + point1_x + ", " + point1_y + ")");
      System.out.println("P2 = (" + point2_x + ", " + point2_y + ")");
      System.out.println("P3 = (" + point3_x + ", " + point3_y + ")");
    }

    public static void printResult(double distance) {
      System.out.printf("Distance between P1 and P2: %.2f\n", distance);
    }

    public static void main(String[] args) {
      point1_x = 5.21;
      point1_y = -7.32;
      point2_x = -0.13;
      point2_y = 3.14;
      point3_x = -4.71;
      point3_y = -8.41;
      System.out.println("Euclidean Distance rounded to 2 decimal places.");
      showPoints();
      printResult(calculateDistance(point1_x, point1_y, point2_x, point2_y));
      printResult(calculateDistance(point1_x, point1_y, point3_x, point3_y));
      printResult(calculateDistance(point2_x, point2_y, point3_x, point3_y));
    }
```

Single responsibility (3)

```java
public static double calculateDistance(Point point1, Point point2) {
  return Math.sqrt(Math.pow(point1.getX() - point2.getX(), 2) + Math.pow(point1.getY() - point2.getY(), 2));
}

public static void showPoints(Point point1, Point point2) {
  System.out.println("P1 = (" + point1.getX() + ", " + point1.getY() + ")");
  System.out.println("P2 = (" + point2.getX() + ", " + point2.getY() + ")");
}

public static void printResult(double distance) {
  System.out.printf("Distance between P1 and P2: %.2f\n", distance);
}

public static void euclideanDistance(Point point1, Point point2) {
  showPoints(point1, point2);
  printResult(calculateDistance(point1, point2));
}

public static void main(String[] args) {
  System.out.println("Euclidean Distance rounded to 2 decimal places.");
  euclideanDistance(point1, point2);
  euclideanDistance(point1, point3);
  euclideanDistance(point2, point3);
}
```

Single responsibility (3)

```java
public class Point {
  private double x;
  private double y;

  public Point() {}

  public Point(double x, double y) {
    this.x = x;
    this.y = y;
  }

  public void setX(double x) {
    this.x = x;
  }
  public double getX() {
    return x;
  }

  public void setY(double y) {
    this.y = y;
  }
  public double getY() {
    return y;
  }
}
```

Done !!!