Calculating the Perimeter of a Shape

Quiz, 7 questions

1 point

1. **For many of the questions in this quiz, you will need to have completed the Calculating the Shape of a Perimeter programming exercise, using the ProgrammingAssignmentRunner BlueJ Project on DukeLearnToProgram.com: http://www.dukelearntoprogram.com/course2/files.php.**

What is the perimeter of the shape made from the file **datatest1.txt** whose contents are shown below (just give to two decimal places)?

-3,3

-4,-3

4,-2

6,5

30.64

1 point

2. What is the average length of a side in the shape made from the file **datatest4.txt** whose contents are shown below (just give to two decimal places)?

-3, 9

-8, 7

-12, 4

-6, -2

-4, -6

2, -8

6, -5

10, -3

8, 5

4, 8

5.94

1 point

3. What is the longest side in the shape made from the file **datatest1.txt** whose contents are shown below (just give to two decimal places)?

-3,3

-4,-3

4,-2

6,5

9.21

1 point

4. What is the largest perimeter of a shape made from the shapes in files **dataset1.txt, dataset2.txt, dataset3.txt, dataset4.txt, dataset5.txt, and dataset6.txt** (just give to two decimal places)?

62.65

1 point

5. What is the name of the file that has the shape with the largest perimeter from the four files **example1.txt, example2.txt, example3.txt and example4.txt**?

- example1.txt
- example2.txt
- example3.txt
- example4.txt

1 point

6. The method getNumPoints returns the number of points in a Shape s.

Which one of the following is NOT a correct implementation of getNumPoints?

```
public int getNumPoints (Shape s) {
  int count = 0;
  for (Point p : s.getPoints()) {
    int newPoint = 1;
    count = count + newPoint;
  }
  return count;
}
```

```
public int getNumPoints (Shape s) {
  int count = 0;
  for (Point p : s.getPoints()) {
    count = count + count;
  }
  return count;
}
```

```
public int getNumPoints (Shape s) {
  int count = 0;
  int newPoint = 1;
  for (Point p : s.getPoints()) {
    count = count + newPoint;
  }
  return count;
}
```

```
public int getNumPoints (Shape s) {
  int count = 0;
  for (Point p : s.getPoints()) {
    count = count + 1;
  }
  return count;
}
```

1 point

7. Consider the following code for the function mysteryShape that has one parameter a Shape s and calls the function getNumPoints from the assignment.

```
public double mysteryShape (Shape s) {
  double tmp = 0;
  for (Point p : s.getPoints()) {

    if (p.getX() > 0) {

      if (p.getY() < 0) {
        tmp = tmp + 1;
      }
    }
  }
  return tmp / getNumPoints(s);
}
```

Which one of the following best describes the purpose of this function?

- The function computes the **sum** of those points from the Shape s that have a **positive X** and a **negative** Y.
- The function computes the **sum** of those points from the Shape s that have a **positive X** or a **negative Y**.
- The function computes the **percentage** of those points from the Shape s that have a **positive X** or a **negative Y**.
- The function computes the **percentage** of those points from the Shape s that have a **positive X** and a **negative Y**.

I, **Ning Zheng**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Submit Quiz