cis112-week02

v2024-10-01

Content

Motivation

This week we focus on 2D geometric shapes:

- `Point`
- `Quadrilateral`
- `Rectangle`
- `Square`

Note that

- `Rectangle` is a special `Quadrilateral`
- `Square` is a special `Rectangle`

We implement `Rectangle` as subclass of `Quadrilateral`, and `Square` as subclass of `Rectangle`.

As usual we have `jUnit`s for testing.

Web resources

- Inheritance
- Interfaces

Introduction

2D Shapes

A 2D shape can be polygons or curved shapes such as circle. Any 2D shape has the following properties:

- `circumference`
- `area`
- `boundingBox`

where "bounding box" is the smallest rectangle that covers the shape entirely.

2D Polygons

We consider "convex" polygons in this lab. A *polygon* has *n* points.

A *quadrilateral* is a 4-point polygon. A *rectangle* and *square* are quadrilaterals, where the edges are parallel to the axes.

Outline

We develop some of the quadrilateral family of shapes.

(i) Abstract class `Polygon` defines the required methods.

(ii) Class `Quadrilateral` extends `Polygon`. Therefore, it must implement the methods called by `Polygon`.

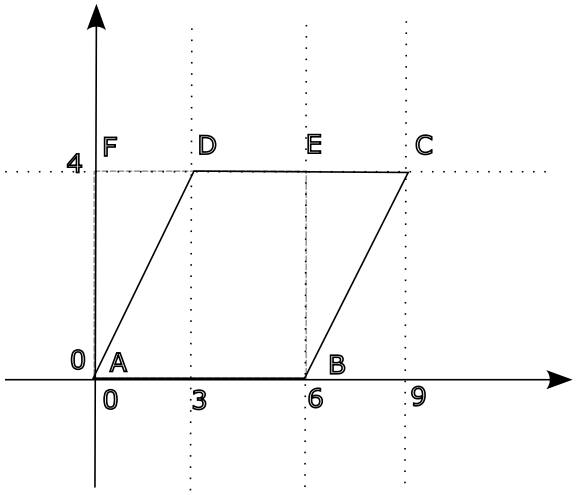
(iii) Class `Rectangle` extends `Quadrilateral`. It overrides methods `circumference`, `area` and `boundingBox` methods using properties of rectangle.

(iv) Similarly class `Square` extends `Rectangle`. It overrides methods `circumference` and `area` using properties of square.

Goal

Reminder - You need to write your code between the following marks.

```
// below ~~~~~~~~~~~~~~~~~~ V
// TODO here
// above ~~~~~~~~~~~~~~~~~~ A
```



Point used in jUnit tests.

Steps:

1. **jUnit.** Consider `CodeToBeTested` and `CodeToBeTested_jUnit`. So far, given the jUnit we improve the code. This time we do the opposite. `CodeToBeTested` is given. Add two more tests to `CodeToBeTested_jUnit`.
2. **Circumference.** Complete `circumference()` methods in `Quadrilateral`, `Rectangle`, and `Square`. Try to optimize performance by using their geometric properties.

   Make sure that your implementation passes the corresponding jUnit test cases.
3. **Area.** Note that `area()` method in `Quadrilateral` is given. `area()` methods in `Rectangle` and `Square` are commented out.

   Question. Run jUnit tests for `Rectangle` and `Square` and observe that area tests are passed. Explain.
4. Remove comments and complete `area()` methods in `Rectangle`, and `Square`. Try to optimize performance by using the geometric properties of the shapes.

   Make sure that your implementation passes the corresponding jUnit test cases.

Challenge

Steps:

1. Complete `boundingBox()` methods in `Quadrilateral`, `Rectangle`, and `Square`. Try to optimize performance by using the geometric properties of the shapes.

   Make sure that your implementation passes the corresponding jUnit test cases.

   Question. Do you need `boundingBox()` in `Square`?

eclipse short cuts

comment in/out (line-comments)

- Select the lines

- Use `control` + `/`

help

- Type first few letters of a command such as `if` or `for`
- Use `control` + `SpaceBar`
- Select the command from the list

System.out.println

- Type `syso`
- Use `control` + `SpaceBar`