Relatório 2

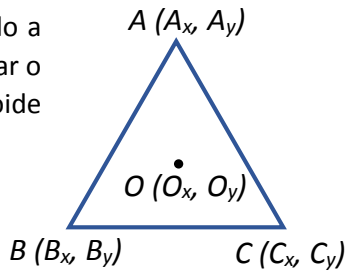
Problema 2 - Raiders of the lost figures

Grupo 10

Alexandra Brito nº50075, Cyrill Brito nº52875, Gonçalo Pedras nº51702

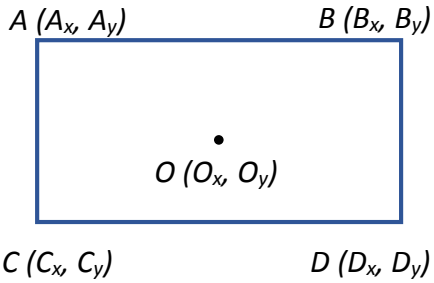
Introdução:

Neste problema tivemos que construir um triangulo ou um rectângulo a partir de uma string, e calcular o seu perímetro e centroid. Para calcular o perimetro de um polígno somamos as arestas do mesmo. Para o centroide O (O_x, O_y) do triangulo utilizamos as formulas:



$$O_x = \frac{A_x + B_x + C_x}{3} \qquad O_y = \frac{A_y + B_y + C_y}{3}$$

, e para o rectangulo utilizamos:



$$O_x = \frac{A_x+B_x+C_x+D_x}{4} \qquad O_y = \frac{A_y+B_y+C_y+D_y}{4} \quad .$$

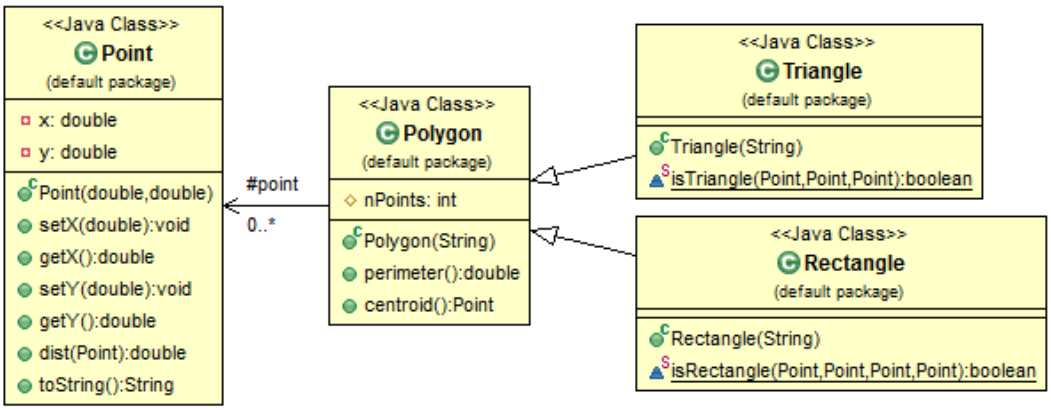
Foi nos fornecido pelo professor a classe *Main* que tivemos de incluir no programa.

Para este problema foi necessário a utilização de superclasses e subclasses.

Uma subclasse é uma classe que deriva de uma superclasse, ou seja, existe uma hierarquia de classes onde as subclasses herdam as propriedades da superclasse.

Diagrama UML:

Este diagrama UML foi criando através do addon sugerido pelo pelo porfessor, ObjectAid. Aqui está deostrado todas as clases do programa com respectivos atributos e funçoes.



Classes:

- Classe *Point*

Não doi mudificada desde de exercícios anteriores.

- Classe *Polygon*

Está é uma superclasse que tem tudo o que é comum em todos os polígonos. Eu seja os pontos, e as funções permeter e centroid.

- Classe *Triangle* e *Rectangle*

Estas são as subclasses da classe Polygon, estas classes apenas têm um construtor e a função isTriangle e isRectangle, que testam para ver se os pontos dados criam um triangulo ou um rectângulo respectivamente.

Teste unitários:

Para o problema também criamos uma classe UnitTests para testarmos se o problema estava a funcionar como se pretendia. Na classe criamos 7 unitTests, um para cada situação para termos a certeza que nada falhava.

1. Um exemplo funcional com um triângulo.
2. Um exemplo funcional com um retângulo.
3. Um exemplo onde os três pontos passados não criam um triângulo.
4. Um exemplo onde os três pontos passados não criam um retângulo.
5. Um exemplo onde a primeira palavra passada não era *triangle* nem *rectangle*.
6. Passamos apenas três ponto em vez de quatro.
7. Passamos apenas quatro ponto em vez de três.

```
public class UnitTests {

   public void test1() { // Working triangle..

   public void test2() { // Working rectangle..

   public void test3() { // Not a triangle..

   public void test4() { // Not a rectangle..

   public void test5() { // Wrong Class..

   public void test6() { // Wrong number points for rectangle..

   public void test7() { // Wrong number points for triangle..
}
```

Discusao:

Dada a função main, criamos a uma classe Polygon que vai ser a superclasse da classe *Triangle* e *Rectangle*. Nesta superclasse criamos 2 variáveis protegidas que podem ser alcançadas pelas suas subclasses:

- Array de Point *point*, que guarda todos os pontos do polígono
- Inteiro nPoints, que guarda o numero de pontos do polígono.

Também criamos 3 métodos:

- Polygon(construtor) - Recebe uma string como argumento vai inserir no array de Point as coordenadas dos pontos e vai guardar o número de pontos obtidos pela consola na variável nPoints
- perimeter – Calcula o perimetro do polígono
- centroid - Calcula o centroid do polígono

De seguida na classe Triangle fizemos uma ligeira alteração, escrevemos a classe como uma subclasse de Polygon e eliminamos as variáveis locais, das quais recebiam os pontos, mas como a super classe já recebe os pontos então podemos usar o array de pontos declarado em Polygon.

Fizemos um novo construtor, este que recebe como argumento uma string, que é enviada para a classe super para guardar os pontos no array e obter o número de coordenadas e verifica se há a existência do polígono com as coordenadas dadas, e, caso não houver envia um exceção.

Criámos também a classe Rectangle que fizemos basicamente o mesmo que a class Triangle. Fizemos um método que recebe uma string, que a envia para a super para obter o array de pontos e número de coordenadas e verifica se existe a existência de um retângulo com aquelas coordenadas.

Bibliografia:

http://www.mathopenref.com/coordcentroid.html

http://www.engineeringintro.com/mechanics-of-structures/centre-of-gravity/centroid-of-rectangle/

https://www.karlin.mff.cuni.cz/network/prirucky/javatut/java/javaOO/subclasses.html