

Relatório 1

Problem 1: The Round Table of the Knights

Grupo 10

Alexandra Brito nº50075, Cyrill Brito nº52875, Gonalo Pedras nº51702

Introduo:

Incircle  um circulo que esta inscrito dentro de um triangulo, como  demonstrado na figura 1, onde o raio corresponde  diviso da rea do triangulo pelo seu semperimetro :

$$r = \frac{A_{\text{triangulo}}}{s}$$

O centro (O_x, O_y) do tringulo  calculado atravs das formulas:

$$O_x = \frac{aA_x + bB_x + cC_x}{p} \text{ e } O_y = \frac{aA_y + bB_y + cC_y}{p}$$

Para saber se 3 pontos podem construir um triangulo  necessrio que as condies seguintes sejam verdadeiras:

$$\begin{aligned} a + b &> c \\ a + c &> b \\ c + b &> a \end{aligned}$$

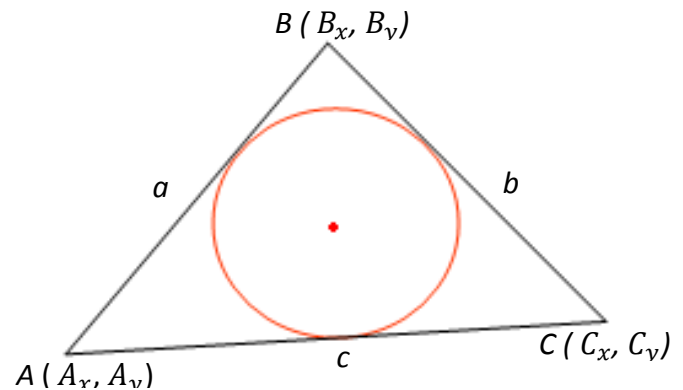


Figura 1

Diagrama UML:

Este diagrama UML foi criando atravs do addon sugerido pelo pelo porfessor, ObjectAid. Aqui est deostrado todas as clases do programa com respectivos atributos e funoes.

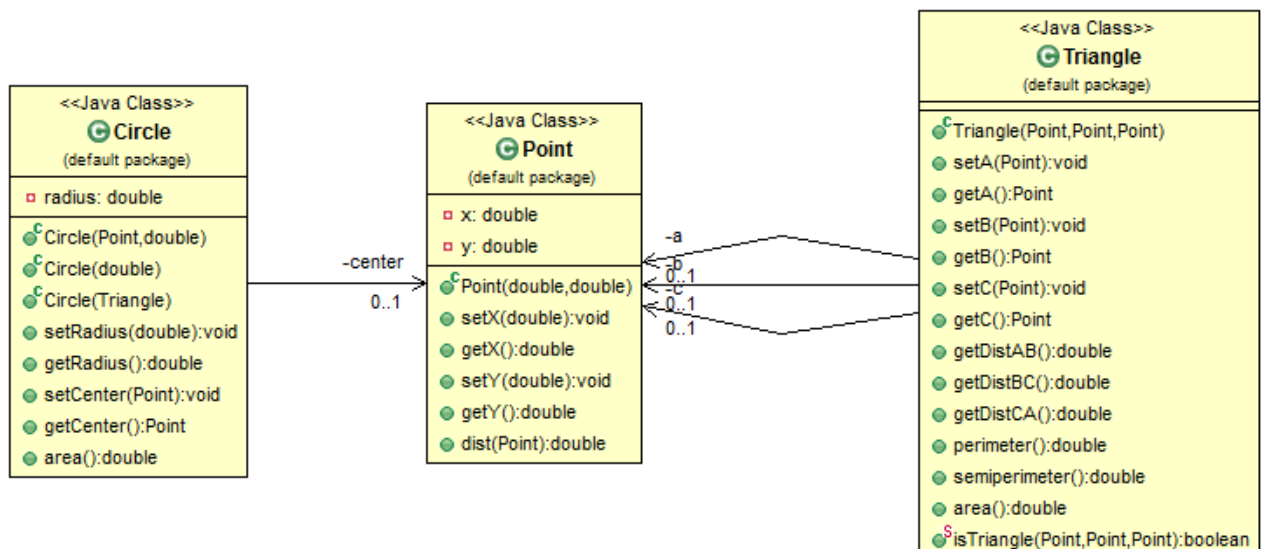


Figura 2

Classes:

❖ Classe Point

Usada para criar o centro do círculo. Não foi modificada desde de exercícios anteriores

❖ Classe Circle

Nesta classe criamos um novo construtor que recebe como parâmetro um objeto da classe triangulo, a partir do objeto recebido o construtor calcula o raio e o centro do círculo.

❖ Classe Triangle

Na classe triangulo criamos 3 pontos, que representam os 3 vértices do triangulo. Para a inicialização do triangulo criamos um construtor que recebe os 3 pontos como parâmetros. Visto que esses pontos são privados criamos get/set para cada ponto, desta forma podemos acessar aos pontos a partir de outras classes. Criamos também, getDistAB, getDistBC, e getDistCA que calcula a distancia entre os pontos. Criamos as funções area, perimeter e semiperimeter. Por fim criamos isTriangle que nos permite verificar se 3 pontos podem construir um triangulo.

Discussão:

Decidimos que era apropriado a criação de uma classe triangulo.

Para a criação do triangulo utilizamos a classe point, já criada em momentos anteriores, como seria necessário a utilização de coordenadas tal classe facilitou a criação dos pontos, que representam os vértices do triangulo.

Decidimos que faria mais sentido que a construção do círculo fizesse parte da classe circle, para isso fizemos o construtor nesta classe que aceita como parâmetros um triangulo, e a partir deste calcula o raio e centro do círculo.

Para verificarmos que 3 pontos poderiam ser do mesmo triangulo, foi criada uma função static na classe tiangle, esta função é static porque é necessário averiguar se os três pontos podem ser um triangulo, anteriormente á criação do objeto.

Teste unitários:

No primeiro teste foi criado um triângulo e a partir desse triângulo foi criado o innercircle e foi verificado se o centro e o raio estavam corretos.

```
@Test
public void test2() {
    Point a = new Point(0, 0);
    Point b = new Point(1, 1);
    Point c = new Point(2, 2);

    assertFalse(Triangle.isTriangle(a, b, c));
}
```

No segundo teste foram inseridos 3 pontos que não formavam um triângulo desta forma podíamos verificar se a função isTriangle estava funcional.

```
@Test
public void test1() {
    Point a = new Point(0, 0);
    Point b = new Point(0, 4);
    Point c = new Point(4, 0);

    assertTrue(Triangle.isTriangle(a, b, c));

    Triangle t = new Triangle(a, b, c);
    Circle circle = new Circle(t);

    assertEquals((int) circle.getCenter().getX(), 1);
    assertEquals((int) circle.getCenter().getY(), 1);
    assertEquals((int) circle.getRadius(), 1);
}
```