

### Laboratório de Introdução ao ambiente IntelliJ e JUnit (Parte 1/2)

#### Objetivos:

- Criação de uma aplicação demonstrativa das funcionalidades do IDE IntelliJ;
- Revisão e aplicação prática de conhecimentos de programação em Java.

#### 01 – Projeto – Criação do Projeto e Package

1. Crie um novo projeto no IntelliJ denominado lab0\_proj **sem usar a opção** "Create Project from Template".
2. Botão direito em cima da pasta **src** e crie um novo "Package" denominado **pt.pa.adts**.

#### 02 – Criação da Interface Shapeable

3. Botão direito em cima do package e crie a **interface** (New/Java Class/Interface] denominada **Shapeable** com os seguintes métodos:
  - **getArea()** – Devolve um double;
  - **getPerimeter()** – Devolve um double;
  - **getHypotenuse()** – Devolve um double;
  - **isPerfect()** – Devolve um boolean.

#### 03 – Criação da classe MyRectangle

4. Crie a classe **MyRectangle** que implementa a interface **Shapeable** contendo os seguintes atributos privados:
  - **x, y, width, height** (double);
  - **name** (string).
5. Utilize as funcionalidades do IDE (menu code/generate) para criar dois **construtores** para a classe:
  - **MyRectangle(String name, double x, double y, double width, double height)**
  - **MyRectangle(String name, double width, double height)** // x,y ficam a zero
6. Crie os **getters** and **setters** para todos os atributos da classe usando o menu code/generate:

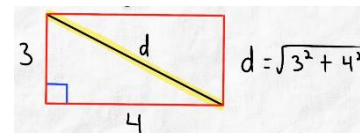
7. A sua classe continua a apresentar erros. Estes são mostrados pelo sublinhado a vermelho no editor porque ainda não implementou os métodos especificados na interface.

```
public class MyRectangle implements Shapeable{
```

Use as funcionalidades do IDE para gerar automaticamente o código das funcionalidades definidas na interface (menu code/generate/Implement Methods).

8. Implemente os métodos da classe sabendo que:

- Se pretende devolver o valor da hipotenusa do triângulo retângulo formado entre os dois lados do retângulo;
- Um retângulo é considerado perfeito se o comprimento do lado menor for superior ou igual a metade do comprimento do lado maior.



9. Faça o **Override** do método **toString** e adicione ao código gerado pelo IDE informação sobre o perímetro, área, hipotenusa e se a figura é considerada perfeita:

```
public String toString() {
    return "MyRectangle{" +
        .....
        .....
        //add getPerimeter()      to result
        //add getArea()          to result
        //add getHypotenuse()     to result
        //add isPerfect()         to result
        .....
    }
}
```

Exemplo:

```
MyRectangle{name='Nice Rect', x=5.0, y=10.0, width=6.0, height=8.0, area=48,00, perimeter=28,00, hypotenuse=10,00000000,
isPerfect=YES}
```

```
MyRectangle{name='Ugly Rect', x=0.0, y=0.0, width=3.8, height=9.0, area=34,20, perimeter=25,60, hypotenuse=9,76933979,
isPerfect=NO}
```

10. Crie a classe **Main** com o método **main()** e escreva o código que permita gerar o output apresentado no exemplo anterior.

## 04 – Criação da subclasse MySquare

- Crie a classe **MySquare**, subclasse da classe **MyRectangle** – tenha em atenção que um quadrado tem os lados todos iguais, sendo por isso um caso particular (especializado) de retângulo.
- Altere o método **isPerfect**. Um quadrado é considerado perfeito se o valor da sua hipotenusa for um número inteiro com um erro máximo de 0.00000001.
- Copie para a classe o código do método **toString** da classe **MyRectangle** e ajuste-o às características da classe **MySquare**.

## 05 – Análise dos resultados da aplicação

---

14. Substitua o método **main** pelo código seguinte e analise os resultados obtidos.

```
public static void main(String[] args) {  
    MyRectangle rectangle = new MyRectangle("Rect X ", 10.0, 20.0, 3.0, 4.0);  
    System.out.println(rectangle);  
  
    MyRectangle otherRectangle = new MyRectangle("Rect Y ", 3.4, 8.3);  
    System.out.println(otherRectangle);  
  
    MySquare square = new MySquare("Q-Lux", 5.0, 6.0, 10.0);  
    System.out.println(square);  
  
    MySquare otherSquare = new MySquare("Q-Dark", 7.071);  
    System.out.println(otherSquare);  
  
    MySquare perfectSquare = new MySquare("Q-Perfect", 7.0710678118654752440084436210485);  
    System.out.println(perfectSquare);  
}
```