

# Aula Prática 9 - Classes de objetos

## Trabalho prévio

1. Desenvolva uma classe de objectos para representar retângulos imutáveis. Um retângulo é construído a partir de um valor de largura e um valor de altura.

```
class Rectangle {  
    final int width;  
    final int height;  
  
    Rectangle(int width, int height) {  
        //...  
    }  
    //...  
}
```

A. Com base na largura e altura desenvolva 3 funções para obter outras propriedades do retângulo, nomeadamente: área, perímetro, e comprimento da diagonal.

```
int area() {  
    //...  
}
```

B. Desenvolva um predicado (função booleana) para verificar se o retângulo é um quadrado.

C. Desenvolva uma função que produz um retângulo com dimensões resultantes de multiplicar as dimensões do retângulo por um dado fator inteiro.

```
Rectangle scale(int factor) {  
    //...  
}
```

D. Desenvolva uma funções para somar um retângulo com outro.

```
Rectangle sum(int width, int height) { //...  
  
Rectangle sum(Rectangle r) { //...
```

E. Desenvolva uma função para verificar se um retângulo “cabe” dentro de outro, i.e., se a largura e a altura são menores ou iguais.

F. Desenvolva uma função estática (static) para criar um retângulo quadrado, dado o comprimento do lado.

## Exercícios

2. Desenvolva uma classe para simular objetos calculadora de inteiros positivos. Quando uma operação resulta num número negativo, o resultado deverá ser zero.

```
class Calculator {  
    int value;  
    //...  
}
```

A. Desenvolva um método para somar um inteiro positivo ao valor guardado.

```
void plus(int n) {  
    //...  
}
```

B. Desenvolva um método para subtrair um inteiro positivo ao valor guardado.

C. Desenvolva um método para reiniciar a calculadora (valor a zero).

D. Desenvolva um método para multiplicar o valor guardado por um inteiro positivo, sem utilizar operador de multiplicação (\*).

E. Desenvolva um método para elevar o valor guardado a um dado expoente positivo, sem utilizar o operador de multiplicação (\*).

F. Desenvolva um método para dividir o valor guardado por um inteiro positivo (divisão inteira), sem utilizar o operador de divisão (/).

G. Desenvolva um método para calcular o resto da divisão inteira do valor guardado por outro número, sem utilizar o operador de resto da divisão (%).

3. Desenvolva uma classe para representar números racionais (frações). Os objetos deverão ser imutáveis, e deverá ser possível criar um número racional de duas formas: (a) dando apenas o numerador, ficando o denominador implicitamente igual a 1; (b) dando o numerador e o denominador.

A. Defina a operação toString() de forma a representar o número racional na forma "n/d".

```
public String toString() {  
    //...  
}
```

B. Defina um método que permita obter o número real (double) representado pela fração.

- C. Defina um método para obter o número racional multiplicado por um dado escalar.
- D. Defina um método para obter o resultado da multiplicação do número racional por outro número racional dado.
- E. Defina um método para obter o resultado da adição do número racional com outro número racional dado.
- F. Defina métodos para verificar se o número racional é igual, maior ou menor que outro número racional dado.

4. Desenvolva uma classe para imagens em tons de cinzento [0, 255], fazendo uso de Rectangle (Questão 1) para representar a dimensão. Se achar necessário, pode implementar incrementos à classe Rectangle.

```
class GrayscaleImage {  
    int[][] pixels; // @grayscaleimage  
    // ...  
}
```

- A. A imagem pode ser criada de duas formas, fornecendo:
  - a. a dimensão com um objeto Rectangle;
  - b. uma imagem a cores (ColorImage) para conversão;
- B. Sobre a imagem deverá ser possível saber:
  - a. número de píxeis;
  - b. dimensão;
  - c. se é quadrada;
- C. Defina um procedimento para desenhar um retângulo num determinado tom.
- D. Defina um procedimento para reduzir o tamanho da imagem (*crop*), fornecendo a nova dimensão (que terá que ser inferior à atual).
- E. Defina um procedimento para aumentar o número de píxeis da imagem dada uma nova dimensão (que terá que ser superior à atual), sendo que os novos píxeis ficarão a preto.

## Exercícios Extra

5. Desenvolva uma classe de objetos para representar coordenadas num tabuleiro (linha e coluna). As coordenadas deverão ser imutáveis, ou seja, não é possível alterar os dados de uma coordenada após o objeto ter sido criado. Porém, deverão ser disponibilizadas funções para produzir outras coordenadas a partir de uma coordenada. Cada coordenada localiza uma posição (linha, coluna) no tabuleiro, sendo o canto superior esquerdo a coordenada (0, 0). Para além das várias funções a serem desenvolvidas, será útil definir um conjunto de constantes para representar direções através de uma coordenada (ver exemplo).

```
class Coordinate {  
    static final Coordinate TOP = new Coordinate(-1, 0);  
    //...
```

- A. Obter uma representação textual da coordenada.
- B. Saber se a coordenada é igual a outra coordenada, ou seja, se os valores de linha e coluna são iguais em ambas.
- C. Saber se a coordenada é adjacente a outra coordenada (diagonais não incluídas).
- D. Saber se a coordenada é adjacente na diagonal a outra coordenada.
- E. Obter uma coordenada adjacente a partir de uma direção (coordenada, tipicamente definida como uma constante, p.e. TOP).
- F. Obter um vetor com todas as coordenadas adjacentes.