

# ALP

FICHA DE EXERCÍCIOS  
ATIVIDADE LETIVA

Programação Orientada a Objetos

UNIDADE CURRICULAR

Orientação a Objetos

FICHA

- 1) Defina uma classe **Retangulo** para representar um retângulo. A classe deve conter:
  - a. Duas variáveis do tipo number chamadas **largura** e **altura** que especificam a largura e a altura do retângulo com valores padrão 1.
  - b. Uma variável do tipo string chamada **cor** que especifica a cor do retângulo. Para este exercício em particular, assuma que TODOS os retângulos possuirão a mesma cor. A cor padrão é branco.
  - c. Defina dois construtores:
    - i. Um que cria um retângulo com a largura e altura especificadas.
    - ii. Um sem argumentos que cria um retângulo padrão.
  - d. Métodos **get** e **set** para as três variáveis da classe. A exceção é para a propriedade **cor** que não deverá ser possível reescrever o seu valor
  - e. Um método chamado **getArea** que retorna a área do retângulo.
  - f. Um método chamado **getPerimetro** que retorna o perímetro do retângulo.
  - g. Crie duas instâncias da classe **Retangulo**
    - i. Uma com os valores padrão
    - ii. Outra com a largura de 10 e altura de 5
  - h. Imprima na consola para ambas as instâncias: as larguras, alturas, cores, áreas e perímetros.
- 2) Defina uma classe **Circulo** para representar círculos. Esta classe deverá conter os seguintes atributos e métodos:
  - a. Uma variável number chamada **raio** para o valor do raio atual do círculo;
  - b. Métodos **get** e **set** para aceder e redefinir o valor da variável **raio**;
  - c. Um construtor que recebe o valor do raio como argumento;
  - d. Um método **getArea** que calcula e retorna a área do círculo;
  - e. Um método **getCircunferencia** que calcula e retorna a circunferência do círculo;

- f. Um método **aumentarRaio** que recebe um number e o usa como percentagem para aumentar o raio do círculo;
  - g. Crie uma instância da classe **Circulo** com o valor do raio dado pelo utilizador. Imprima o raio, área e circunferência. Depois peça ao utilizador uma percentagem para aumentar o raio do círculo. Reimprima todas as variáveis.
- 3) Defina uma classe chamada **Ventoinha** com o propósito de representar uma ventoinha do mundo real.



- Para esta classe deverá criar 3 constantes com os nomes de BAIXA, MEDIA e RAPIDA com os valores 1, 2 e 3. Estas constantes serão usadas para definir e representar a velocidade atual da ventoinha. A classe **Ventoinha** deverá possuir ainda os seguintes campos de dados e métodos:
- a. Uma variável number chamada **velocidade**, que define e mostra a velocidade atual da ventoinha (o padrão é BAIXA);
  - b. Uma variável booleana que mostra se a ventoinha está ligada ou desligada. O padrão é desligada, ou seja, **false**;
  - c. Métodos **get** e **set** para obter/redefinir valores das variáveis privadas;
  - d. Construtor sem argumentos que cria nova ventoinha com valores padrão;
  - e. Um método chamado **toString** que retorna uma descrição textual da ventoinha com: velocidade e se está ligada ou desligada.
  - f. Crie duas instâncias da classe **Ventoinha**.
  - g. Mostre todos os dados da primeira ventoinha. Aumente a velocidade da segunda ventoinha para MÉDIA e ligue-a. Finalmente, mostre todos os dados da segunda ventoinha.
- 4) Os dados são pequenos poliedros gravados com determinadas instruções. O dado mais clássico é o cubo (seis faces), gravado com números de um a seis.



Para este exercício deverá criar uma classe **Dado** para representar um dado e que nos permitirá rolá-lo e tirar valores que variarão de 1 a 6, ou seja, um dado de seis lados. A classe deverá conter os seguintes campos de dados e métodos:

- a. Uma variável constante `number` que define quantas faces o dado possui. O valor desta constante deverá ser 6;
  - b. Uma variável `number` chamada **`valorFace`** que guardará o valor atual da face do dado;
  - c. Um construtor sem argumentos que cria um novo objeto da classe **`Dado`** e define o valor da variável **`valorFace`** como 1;
  - d. Métodos **`get`** e **`set`** para a variável **`valorFace`**. O método **`set`** deve garantir que o valor fornecido como argumento esteja dentro da gama de valores esperada. Se o valor for inválido, exiba uma mensagem para a consola com o texto "Valor inválido para a face do dado";
  - e. Um método **`rolar`** que vai "rolar" o dado e guardar o resultado na variável **`valorFace`**. O valor sorteado deverá estar na gama de 1 até a quantidade de faces do dado;
  - f. Um método chamado **`getQuantidadeFaces`** que retorna um `number` representando a quantidade de faces do dado.
- 5) Crie uma classe **`Pais`** que armazena o nome do país, a sua população e a sua área (quilometro quadrado).
- a. Crie pelo menos 5 instâncias da classe;
  - b. Guarde todas as instâncias num array chamado **`países`**;
  - c. Defina métodos estáticos da classe para cálculo do país:
    - i. Com a maior área;
    - ii. Com a maior população;
    - iii. Com a maior densidade populacional (população por km<sup>2</sup>).
  - d. Ordene crescentemente e liste o array de objetos pelo nome do país.