

Lista de Exercícios 03: Revisão Para Avaliação

1. Em linguagem de programação C++, elabore uma classe denominada `JogadorFutebol` que possui os atributos `nome`, `altura`, `peso`, `idade` e `posicao`. Em relação aos membros de dados de um objeto desta classe, o atributo `nome` deve ser da classe `string` (incluir biblioteca `string`), a `altura` deve ser armazenada em centímetros, o `peso` em quilos e a `idade` em anos. Para a posição do jogador, considere os seguintes valores: 0 para goleiro, 1 para defensor (zagueiro e laterais), 2 para meio de campo, 3 para atacante e -1 se a posição estiver indefinida. A classe `JogadorFutebol` deve possuir as seguintes funcionalidades:

- Método construtor sem parâmetros que inicializa os atributos `nome`, `altura`, `peso`, `idade` e `posicao` com os valores "", 170, 70, 18 e -1, respectivamente;
- Método construtor com parâmetros, em que cada um dos parâmetros deve corresponder ao valor desejado para os atributos do objeto ao ser instanciado. Assim, este método deve possuir cinco parâmetros de entrada;
- Métodos acessores `get` e `set` com o objetivo de garantir acesso seguro aos atributos privados da classe. Os atributos `altura`, `peso` e `idade` devem ser positivos. Caso um valor diferente dos esperados para definir a posição do `JogadorFutebol` for utilizado como argumento do método `setPosicao`, deve ser atribuído o valor -1 para o atributo `posicao`;
- Método que calcula e retorna o valor do Índice de Massa Corporal (IMC) do `JogadorFutebol`. O cálculo do IMC é dado pelo `peso` dividido pela `altura` ao quadrado, em que o `peso` é dado em **Kg** e a `altura` em **metros**;
- Método que imprime em tela os valores de todos os atributos do objeto, o IMC do `JogadorFutebol` e a sua condição em relação ao IMC, conforme a tabela logo a seguir. **Observação.** Para exibir a situação do objeto em relação ao seu IMC, pode ser utilizado **método privado** (método auxiliar da classe) que recebe como argumento o valor calculado do IMC e exibe em tela o texto correspondente à situação do objeto.

Resultado	Situação
Abaixo de 17	Muito abaixo do <i>peso</i>
Entre 17 e 18,5	Abaixo do <i>peso</i>
Entre 18,5 e 24,9	<i>Peso normal</i>
Entre 25 e 29,9	Acima do <i>peso</i>
Entre 30 e 34,9	Obesidade I
Entre 35 e 39,9	Obesidade II (severa)
Acima de 40	Obesidade III (mórbida)

2. Desenvolva uma classe denominada `Triangulo` em linguagem de programação C++. Um objeto da classe `Triangulo` deve possuir os atributos `altura` e `base`, que devem ser dados em centímetros, e `nome` que deve ser do tipo de dado `char`. `'A'`, `'B'`, `'C'` são alguns exemplos de nomes que podem ser atribuídos ao atributo `nome`. A classe `Triangulo` deve ter as seguintes características:

- Método construtor sem parâmetros que inicializa os atributos `altura` e `base` com o valor de 10 centímetros e `nome` como sendo igual a `' '` (caractere vazio);
- Método construtor com parâmetros que inicializa todos os atributos do objeto por meio de valores informados como argumentos do método. Assim, o método construtor deve possuir três parâmetros;
- Métodos acessores `get` e `set` para cada um dos atributos da classe. Seguindo o princípio do encapsulamento, deve-se evitar que os atributos da classe assumam valores inválidos. Lembre-se que os valores de `altura` e `base` de um triângulo devem ser positivos;
- Método que calcula e retorna o valor da área do `Triangulo`;
- Método que exibe em tela o estado do objeto da classe, apresentando os valores de seus atributos e o valor da área do `Triangulo`;
- Método destrutor que imprime em tela o seguinte texto: “Destruicao do Triangulo A”, em que `'A'` deve ser o `nome` do objeto.

3. Em linguagem de programação C++, elabore uma classe denominada `Carro` que possui os atributos `velocidade`, `ligado` e `placa`, sendo este último um objeto da classe `string` (biblioteca `string`). Sempre que um objeto da classe `Carro` é instanciado, a sua velocidade deve ser igual a 0 Km/h e o seu atributo `ligado` igual a `false`. Os carros desta classe possuem velocidade controlada, não podendo ultrapassar a velocidade máxima de 100 Km/h. A classe `Carro` deve possuir as seguintes características:

- Método construtor com um único parâmetro que recebe uma `string`, contendo a identificação da placa do objeto da classe `Carro` a ser instanciado;
- Métodos públicos `getVelocidade` e `getPlaca` que retornam os valores dos atributos `velocidade` e `placa`, respectivamente, do objeto da classe `Carro`;
- Método público `desligarCarro`, que desliga o carro, fazendo com o que o atributo `ligado` assumo o valor `false`. Para que o carro seja desligado, ele deve estar parado. Ou seja, a sua velocidade deve ser igual a 0 Km/h;
- Método público `ligarCarro`, que faz com que o atributo `ligado` assumo o valor `true`;
- Método `acelerarCarro` que possui um único parâmetro, cujo valor deve ser entre 1 e 10 Km/h. Este método deve incrementar a velocidade do veículo de acordo com o valor informado em seu parâmetro. **Observações.** Não é possível efetuar incrementos de velocidade acima de 10 Km/h. Portanto, se um valor maior do que este for passado para o método como argumento, o valor de 10 Km/h deve ser assumido. Não ocorrerá aumento de velocidade do veículo se o valor passado como argumento for menor do que 1 Km/h. A velocidade do carro apenas é incrementada se o veículo estiver ligado, não podendo ultrapassar o valor de 100 Km/h;
- Método `freiarCarro`, que possui um único parâmetro, cujos valores estão entre 1 e 10 Km/h. Este método deve reduzir a velocidade do veículo de acordo com o valor informado em seu parâmetro. **Observações.** Não é possível efetuar decréscimos de velocidade com valores acima de 10 Km/h. Portanto, se um valor maior do que este for passado para o método como argumento, o valor de 10 Km/h deve ser assumido. Não ocorrerá alteração de velocidade do veículo se o valor passado como argumento for menor do que 1 Km/h. A velocidade do carro não pode ser decrementada se o veículo estiver parado (velocidade igual a 0 Km/h) e não pode assumir valores negativos.
- Método `imprimir`, que exibe em tela a placa do veículo, sua velocidade e se ele se encontra ligado ou desligado.