



**ALGORITMOS E ESTRUTURAS DE DADOS I**  
**LISTA DE EXERCÍCIOS 1B – POO BÁSICO COM C++**  
**PROF. FLÁVIO JOSÉ MENDES COELHO**

CLASSES, OBJETOS, ATRIBUTOS, MÉTODOS E CONSTRUTORES EM C++

1. Codifique uma *classe* chamada **Pedra** que representa uma pedra do jogo de dominó. Uma pedra armazena dois valores inteiros no intervalo de 0 à 6. Escreva uma função **main** para: (a) instanciar uma pedra; (b) atribuir valores inteiros para a pedra; (c) imprimir a pedra na tela (por exemplo, a pedra  deve ser mostrada na tela assim: [2|5]).
2. Estenda o programa da questão anterior adicionando à classe **Pedra** um método chamado **print** para imprimir a pedra na tela. Por exemplo, a pedra  deve ser impressa na tela assim: [2|5]. Escreva uma função **main** que: (a) instancie uma pedra; (b) atribua valores inteiros para a pedra; (c) imprima a pedra na tela empregando o método **print**.
3. Estenda o programa da questão anterior adicionando à classe **Pedra** um construtor padrão (*default*) que inicialize a pedra com valores nulos, isto é, ao ser instanciada, uma pedra ficará assim: [0|0]. Escreva uma função **main** que: (a) instancie uma pedra; (b) imprima a pedra na tela empregando o método **print**; (c) atribua novos valores inteiros para a pedra; (d) imprima a pedra novamente na tela.
4. Estenda o programa da questão anterior para incluir à classe **Pedra** um construtor com parâmetros que receba os dois valores da pedra como parâmetros. Escreva uma função **main** que: (a) instancie uma pedra inicializando-a com valores inteiros quaisquer; (b) imprima a pedra na tela empregando o método **print**; (c) atribua novos valores inteiros para a pedra; (d) imprima a pedra novamente na tela.
5. Estenda o programa da questão anterior para incluir à classe **Pedra** um método chamado **valida** (sem parâmetros) para verificar se a pedra é válida, isto é, se cada um de seus valores está dentro do intervalo de 0 à 6. Este método deve retornar o valor *true* se a pedra for válida, ou retornar *false*, em caso contrário. Escreva uma função **main** que: (a) instancie uma pedra inicializando-a com valores inteiros quaisquer; (b) imprima a pedra na tela caso a mesma seja válida, ou mostre a mensagem “pedra inválida”, em caso contrário. Empregue o método **print**.
6. Modifique o programa da questão anterior para que o método **print** da classe **Pedra** só imprima a pedra se ela for válida. Se a pedra for inválida, o método deverá imprimir a mensagem “inválida”. Escreva uma função **main** para testar a nova versão do método **print**.
7. Modifique o programa da questão anterior encapsulando a classe **Pedra**. Escreva métodos *get* para cada um dos valores de uma pedra. Escreva uma função **main** para instanciar um pedra e obter seus valores por meio dos métodos *get*, e imprima estes valores na tela.



8. Codifique um programa em C++ que: (a) crie e inicialize estaticamente um vetor de pedras de dominó; (b) escreva uma função que receba o vetor e seu tamanho como parâmetros, e imprima todas as pedras presentes no vetor.
9. Crie uma nova versão do programa da questão anterior para: (a) criar e inicializar dinamicamente um vetor de pedras de dominó utilizando o operador **new** do C++; (b) escreva uma função que imprima todas as pedras presentes no vetor; (c) desaloque o vetor da memória empregando o operador **delete**.
10. Crie uma nova versão do programa da questão anterior para: (a) criar e inicializar dinamicamente um vetor de ponteiros para pedras de dominó utilizando o operador **new** do C++; (b) escreva uma função que imprima todas as pedras presentes no vetor; (c) desaloque o vetor da memória empregando o operador **delete**.

[FIM-DA-LISTA]