# MAB225 – Computação II – Aula Prática 3

**Objetivo:** programação orientada a objeto – herança, polimorfismo e o método construtor

Para todos os exercícios abaixo: teste as classes: crie instâncias, modifique os atributos e utilize os métodos.

1. Crie uma classe em Python para representar o tabuleiro de um jogo. Essa classe deve possuir os seguintes atributos e métodos:

    a. Método construtor - recebe o número de linhas e colunas do tabuleiro completo. Cria um atributo com número de linhas e outro com o número de colunas. Em seguida, cria um atributo que representa o tabuleiro em si. Esse atributo deve ser do tipo lista de listas com o caractere asterisco como elemento da sub-lista. A lista de listas deve ter tamanho igual ao número de linhas e colunas passado por argumento. (Ex.: Se as entradas forem 2,3 o atributo que representa o tabuleiro deve ser: [[‘*’, ‘*’, ‘*’] , [‘*’, ‘*’, ‘*’]] .)
    b. Método para modificar um caractere do tabuleiro - recebe linha e coluna do tabuleiro e verifica se os valores passados são maiores ou iguais a zero e menores que os atributos de número de linhas e colunas, respectivamente. Caso afirmativo, troca a posição referente à linha e coluna do tabuleiro pelo caractere passado por argumento.
    c. Método reiniciar - substitui todas as posições do tabuleiro diferentes de asterisco por asterisco.
    d. Método imprimir - mostra o tabuleiro em formato de matriz. Exemplo:

```
>>> t = Tabuleiro(4,4)
>>> t.imprimir()
* * * *
* * * *
* * * *
* * * *
>>>
```

2. Crie uma classe herdeira de Tabuleiro que implementa um jogo da velha. Defina os seguintes métodos e atributos:
    a. Sobrescreva o método construtor - não recebe nenhum argumento além da referência para o objeto (self) e chama o método construtor da classe pai de forma que o número de linhas e colunas seja igual a 3 e 3 (veja como chamar o método da classe pai nos slides 50 ou 51 da última aula – duas formas diferentes). Em seguida, cria um atributo para o jogador 1 igual ao caractere 'X' e outro para o jogador 2 igual ao caractere 'O'.
    b. Método para verificar vitória - recebe o jogador que desejamos verificar se ganhou e verifica todas as suas possibilidades de vitória. Retorna True se o jogador ganhou e False caso contrário.
    c. Método jogar - funciona em um loop com no máximo 9 jogadas. A cada iteração imprime o tabuleiro e pede para o jogador da vez digitar a posição que deseja jogar

(peça o número da linha e coluna separadamente). A primeira jogada deve ser do jogador 1 e as demais jogadas são alternadas. Verifique se a posição escolhida a cada rodada está vazia e se é uma posição dentro do tabuleiro. Se sim, utilize o método que modifica um caractere do tabuleiro (método herdado) para registrar a jogada. Se não, imprima uma mensagem de erro e volte a pedir a posição desejada para o mesmo jogador até que ele passe um valor correto. Utilize o método de verificar vitória a cada jogada. Se houve vitória, imprima uma mensagem parabenizando o jogador vencedor e termine o jogo. Informe também se o jogo acabar em velha. No final, utilize o método reiniciar (herdado) para limpar o tabuleiro.

3. [Opcional] Crie uma outra classe herdeira de Tabuleiro, agora para implementar um jogo da memória. Defina os seguintes métodos e atributos:
    a. Sobrescreva o construtor – pode receber o número de linhas e colunas como argumento, mas ambos possuem valor default igual a 4. Verifique se a multiplicação dos valores passados por argumento é par. Caso positivo, chama o método construtor da classe pai considerando o número de linhas e colunas definido pelo usuário. Caso contrário, chama o método construtor da classe pai considerando que o número de linhas e o número de colunas é igual a 4. Em seguida, chama o método da letra b e o método da letra c.
    b. Método para criar cópia de tabuleiro - Cria um atributo com uma cópia do tabuleiro. (Atenção ao criar uma cópia de uma lista de listas).
    c. Método embaralhar – define as posições dos pares do jogo da memória. Os pares devem ser números de 1 até (número de linhas * número de colunas)/2 . Utilize um método da biblioteca random para embaralhar. Possível implementação: crie uma lista com os números e uma lista de tuplas com as posições do tabuleiro (exemplo para um tabuleiro 2x2: *num* = [1, 1, 2, 2]; *pos* = [(0,0),(0,1),(1,0),(1,1)] ). Utilize o método shuffle da biblioteca random para trocar as posições das tuplas dentro da lista. Em seguida, utilize o método da questão 1.b (herdado) para adicionar os elementos da lista *num* no atributo do tabuleiro considerando as posições da lista *pos* depois do shuffle. Esse método deve modificar somente tabuleiro, a cópia deve permanecer igual a uma matriz de asteriscos.
    d. Sobrescrever o método imprimir – apresenta o tabuleiro ou a cópia em formato de matriz. Recebe a lista de listas que deve ser apresentada por argumento.
    e. Método jogar – funciona em loop até que todos os pares sejam encontrados. Apresenta na tela a cópia do tabuleiro (através do método imprimir) e pergunta para o usuário a posição que deseja ver (linha e coluna separadamente). Acrescente o elemento daquela posição à cópia do tabuleiro, o elemento da posição pedida e imprima (utilize o método imprimir). Repita o procedimento para uma segunda posição. Se os valores que aparecerem nas duas posições forem iguais, a cópia do tabuleiro deve manter os dois valores de forma que eles apareçam a cada rodada. Se não, ambas posições devem voltar a ser igual ao caractere asterisco. Em seguida, o método deve voltar para o início do primeiro loop para pedir novos pares.