

Trabalho Prático 02 (TP02)

Instruções:

- i - O arquivo deve ser entregue em formato .ZIP ou .RAR seguindo a nomenclatura: “XXXX.KKK” onde XXXX é o número de sua matrícula e KKK a extensão do arquivo.
- ii - Cada um dos exercícios deve criado em um diretório com o seguinte nome: Exercicio_XX onde XX é o número da questão solucionada.
- iii - Para cada programa desenvolvido deverão ser entregues **SOMENTE** os arquivos de projeto e classes Java em seus respectivos pacotes.
- iv - O arquivo deve ser enviado via moodle limitado a data e hora de entrega definida no Plano de Ensino. Não serão aceitos trabalhos enviados por e-mail.

Questão 1. Grafo é uma estrutura de dados muito comum em computação, e os algoritmos sobre grafos são fundamentais para a área. Um grafo $G = (V; A)$ consiste em:

- um conjunto finito de pontos V . Os elementos de V são chamados de vértices de G .
- um conjunto finito A de pares não ordenados de V , que são chamados de arestas de G . Uma aresta a em A é um par não ordenado $(v; w)$ de vértices v, w em V , que são chamados de extremidades de a .

Uma aresta a em A é chamada de incidente com um vértice v em V , se v for uma extremidade de a . Um vértice v em V diz-se vizinho de outro vértice w em V se existir uma aresta a em A incidente com v e w . Um grafo pode ser representado por listas de adjacência ou por uma matriz de adjacência, como é ilustrado na Figura 1.

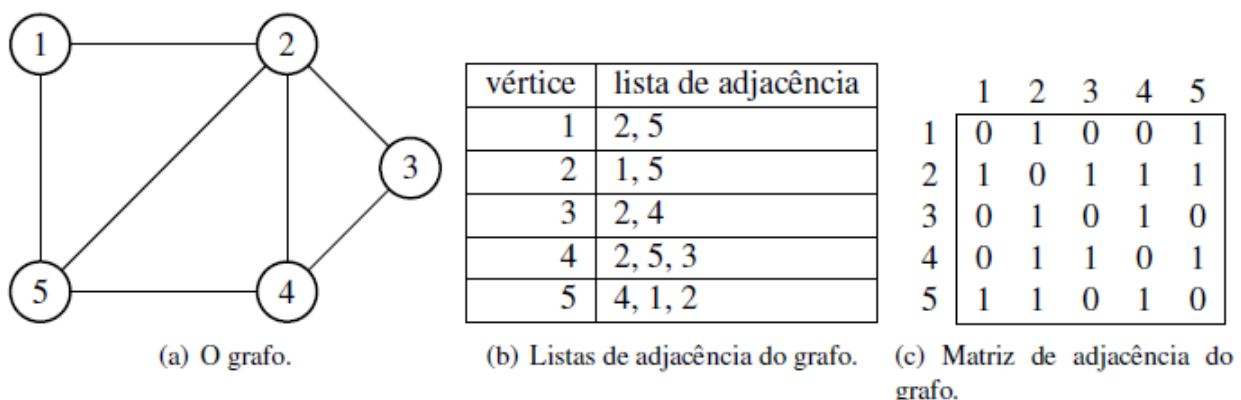


Figura 1: Exemplo de representações de grafos.

Escreva uma classe para representar grafos. Escolha entre a representação por listas de adjacência ou por matriz de adjacência. A classe deve oferecer uma operação para determinar se dois vértices são vizinhos, e outra operação para determinar a lista de todos os vértices que são vizinhos de um dado

vértice. Considere que cada vértice é representado por um número inteiro. Escreva um aplicativo para testar a classe.

Questão 2. Escreva um programa completo para jogar o jogo da velha. Para tanto crie uma classe *Jogo Da Velha*:

- A classe deve conter como dados privados um array bidimensional 3×3 para representar a grade do jogo.
- Crie uma enumeração para representar as possibilidades de ocupação de uma casa na grade (vazia, jogador 1 ou jogador 2).
- O construtor deve inicializar a grade como vazia.
- Forneça um método para exibir a grade.
- Permita dois jogadores humanos.
- Forneça um método para jogar o jogo; todo movimento deve ocorrer em uma casa vazia; depois de cada movimento, determine se houve uma derrota ou um empate.

A enumeração proposta pode ser definida utilizando o **enum**¹ a seguir:

```
1 public enum StatusPlace {  
2     EMPTY, PLAYER_ONE, PLAYER_TWO;  
3 }
```

Listing 1: Proposta para representar as possibilidades de ocupação de uma casa na grade.

¹Para o entendimento de enumeradores utilizando a linguagem Java você poderá consultar a seguinte url: <http://www.devmedia.com.br/tipos-enum-no-java/25729>.