

Representação de um grafo usando uma estrutura `ArrayList`

Neste exemplo pretende-se usar um `ArrayList` para representar os arcos de um grafo. Cada elemento do `ArrayList` irá conter um triplo que define um arco do grafo: Origem, Destino e Distância.

Considere que o grafo é dirigido ou orientado, ou seja, cada arco tem um sentido. Por exemplo, o arco definido pelo triplo $(A, B, 5)$ é um arco unidirecional que permite ligar o vértice A ao vértice B e tem uma distância de 5 unidades. Para definir uma ligação direta e bidirecional entre os nós A e B será necessário definir dois arcos: $(A, B, 5)$ e $(B, A, 5)$.

1. Declare as classes `Vertice` e `Edge` para representar os vértices e os arcos do grafo.
2. Declare a classe `Graph` para representar um grafo. Um grafo é formado por uma coleção do tipo `ArrayList` cujos elementos são objetos do tipo `Edge`.
3. Na classe `Graph` declare o método de instância `getVertices()` que devolve uma coleção contendo todos os vértices do grafo.
4. Na classe `Graph` declare o método de instância `getVerticesConnectedTo(Vertice v)` que devolve uma coleção contendo todos os nós ligados por um arco a partir do vértice v .

Graph representation using an `ArrayList` structure

In this example, we intend to use an `ArrayList` to represent the edges of a graph. Each element of the `ArrayList` will contain a triple that defines an edge of the graph: Source, Destination, and Distance.

Consider that the graph is directed, meaning each edge has a direction. For example, the edge defined by the triple $(A, B, 5)$ is a unidirectional edge that connects vertex A to vertex B and has a distance of 5 units. To define a direct and bidirectional connection between nodes A and B , it will be necessary to define two edges: $(A, B, 5)$ and $(B, A, 5)$.

1. Declare the classes `Vertex` and `Edge` to represent the vertices and edges of the graph.
2. Declare the class `Graph` to represent a graph. A graph is formed by a collection of type `ArrayList` whose elements are objects of type `Edge`.
3. In the `Graph` class, declare the instance method `getVertices()` which returns a collection containing all the vertices of the graph.
4. In the `Graph` class, declare the instance method `getVerticesConnectedTo(Vertex v)` which returns a collection containing all the nodes connected by an edge from the vertex v .