

DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Bônus](#) / [\[Bonus\] Parte 1 - Lista de exemplos](#)

Descrição [Visualizar envios](#)

[Bonus] Parte 1 - Lista de exemplos

Data de entrega: quarta, 4 Ago 2021, 23:59
Arquivos requeridos: ListaExemplos.java ([Baixar](#))
Tipo de trabalho: Trabalho individual

Parte 1 - Lista de exemplos

Nesta primeira parte do exercício, deve ser implementada uma lista de exemplos. A lista de exemplos neste exercício é um conjunto de dados formado por uma tabela. Cada linha da tabela é um exemplo.

A figura a seguir mostra uma lista de exemplos:

a1	a2	a3	rótulo da classe
1,0	0,9	1,1	1
1,1	0,8	1,0	1
4,0	8,0	0,9	2
1,2	1,1	0,9	1
4,0	6,0	0,1	2

Nessa tabela, cada linha é um exemplo. Os atributos a1, a2 e a3 serão chamados de atributos de entrada e o último atributo (última coluna) será chamado de rótulo da classe. Neste exercício, todos os atributos de entrada são do tipo double e o rótulo da classe será representado por um número inteiro.

Esses dados da lista de exemplos serão usados em [\[Bonus\] Parte 2 - Algoritmo de classificação](#) para que um algoritmo possa prever o rótulo da classe ao receber novos valores dos atributos de entrada (dados de teste).

Ordenação pela distância

Conforme será descrito mais a frente, uma funcionalidade que deverá ser implementada neste exercício é a ordenação dos exemplos pela distância com relação a um determinado conjunto de valores de atributos de entrada.

No caso deste exercício, a distância utilizada será a distância Euclidiana, que é calculada de acordo com a fórmula a seguir:

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Cálculo da distância Euclidiana entre x=[1,2,1] e y=[3,0,9]

$$\sqrt{(1 - 3)^2 + (2 - 0)^2 + (1 - 9)^2} = \sqrt{4 + 4 + 64} = 8,49$$

No caso da lista de exemplos, é possível calcular a distância de cada exemplo na tabela com relação um determinado conjunto de valores de atributos de entrada (o cálculo da distância considera apenas os atributos de entrada, no caso a1, a2 e a3). A tabela a seguir, mostra a distância entre cada exemplo na lista e os valores [5, 7, 1]:

a1	a2	a3	rótulo da classe	distância
1,0	0,9	1,1	1	7,30
1,1	0,8	1,0	1	7,32
4,0	8,0	0,9	2	1,42
1,2	1,1	0,9	1	7,02
4,0	6,0	0,1	2	1,68

A partir das distâncias, os exemplos podem ser ordenados pela distância (ordem crescente), conforme apresentado a seguir:

a1	a2	a3	rótulo da classe	distância
4,0	8,0	0,9	2	1,42
4,0	6,0	0,1	2	1,68
1,2	1,1	0,9	1	7,02
1,0	0,9	1,1	1	7,30
1,1	0,8	1,0	1	7,32

Classes a serem implementadas

Duas classes devem ser implementadas nesta parte 1 do exercício: Exemplo e ListaExemplos (as duas classes devem estar no pacote classificacao). A seguir, são apresentados os métodos necessários em cada uma das classes.

[acesso package] Classe **Exemplo**

- public Exemplo(double[] atributosEntrada, int rotuloClasse) - construtor que inicializa o exemplo com os valores dos atributos de entrada e o rótulo da classe.
- public double[] getAtributosEntrada() - retorna os valores dos atributos de entrada.
- public int getRotuloClasse() - retorna o valor do rótulo da classe.

[acesso public] Classe **ListaExemplos**

- public ListaExemplos(int maxQtd) - construtor que recebe como parâmetro a quantidade máxima (maxQtd) de exemplos que a lista pode possuir.
- public void adicionarExemplo(Exemplo exemplo) - adiciona um exemplo ao final da lista de exemplos.
- public void ordenarPelaDistancia(double[] atributosEntrada) - ordena os exemplos da lista em ordem crescente considerando a distância Euclidiana de cada exemplo com relação aos valores de atributos de entrada passados como parâmetro para o método. Uma explicação mais detalhada sobre como essa ordenação deve ser realizada é apresentada neste enunciado na seção Ordenação pela distância.
- public int getQtdExemplos() - retorna a quantidade de exemplos na lista.
- public Exemplo[] getExemplos() - retorna todos os exemplos armazenados na lista.
- public Exemplo[] getPrimeirosExemplos(int n) - retorna os primeiros *n* exemplos na lista.
- public Exemplo[] getUltimosExemplos(int n) - retorna os últimos *n* exemplos na lista.

Observação: Não é permitido o uso de funções de ordenação prontas (por exemplo, sort). Implemente um algoritmo de ordenação. Também não use o termo "sort" em nenhuma parte do código.

Importante: Submeta apenas as classes Exemplo e ListaExemplos descritas no enunciado (todas as classes devem estar no **pacote classificacao**). As classes submetidas não podem realizar impressão de dados, utilizar import e java.util.

Casos de teste

Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada

- Sequência de operações no formato:
 - instanciarListaExemplos [qtdMax]
 - lista [índice lista] adicionarExemplo [qtd_atributos_entrada] (valores dos atributos de entrada) [rótulo da classe]
 - lista [índice lista] ordenarPelaDistancia [qtd_atributos_entrada] (valores dos atributos de entrada) [rótulo da classe]
 - lista [índice lista] getExemplos
 - lista [índice lista] getPrimeirosExemplos [n]

- lista [índice lista] getUltimosExemplos [n]

Saída

- Operações executadas e saídas obtidas

[VPL](#)

◀ [\[Aula\] Polimorfismo com sobreposição](#) Seguir para... [\[Bonus\] Parte 2 - Algoritmo de classificação](#) ▶



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

[Conheça a UFABC](#)

[Conheça o NTI](#)

[Conheça o Netel](#)

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos



[Universidade Federal do ABC](#) - [Moodle](#) (2020)

[Obter o aplicativo para dispositivos móveis](#)