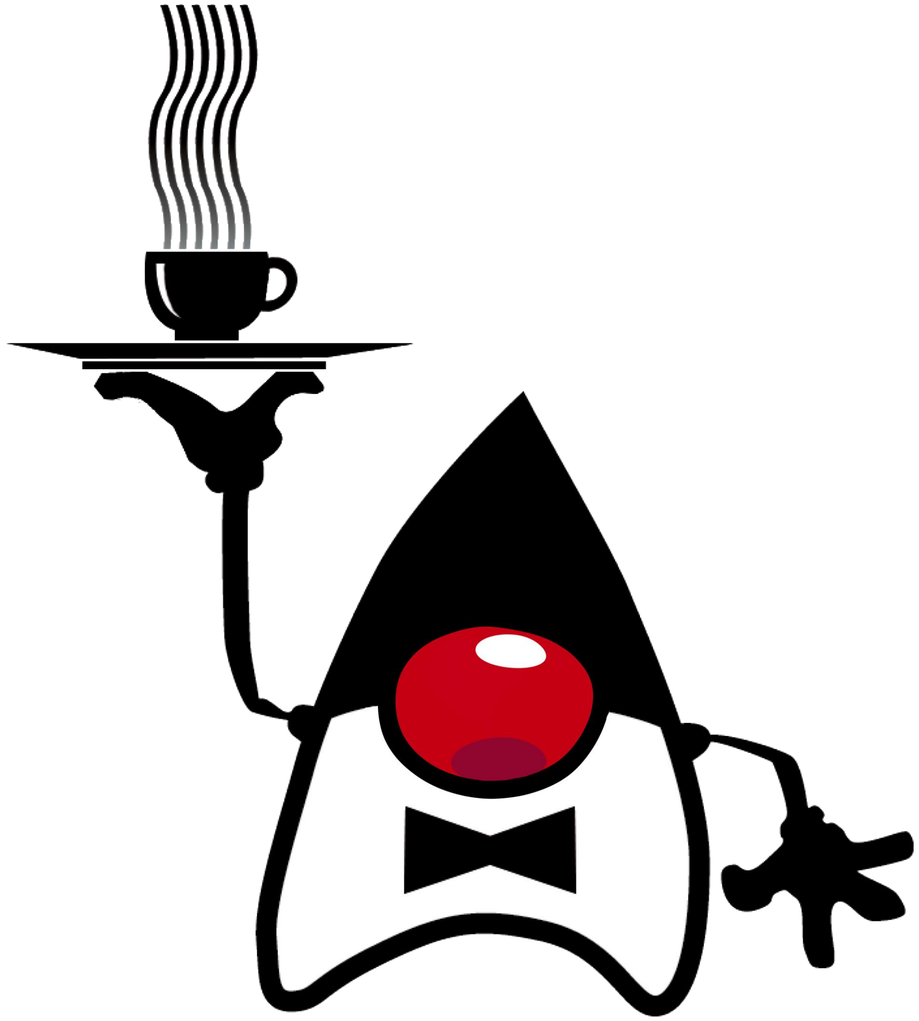
Trilha Java

Encontro 26 – Exemplo e Atividade 13



vnt/school
powered by venturus

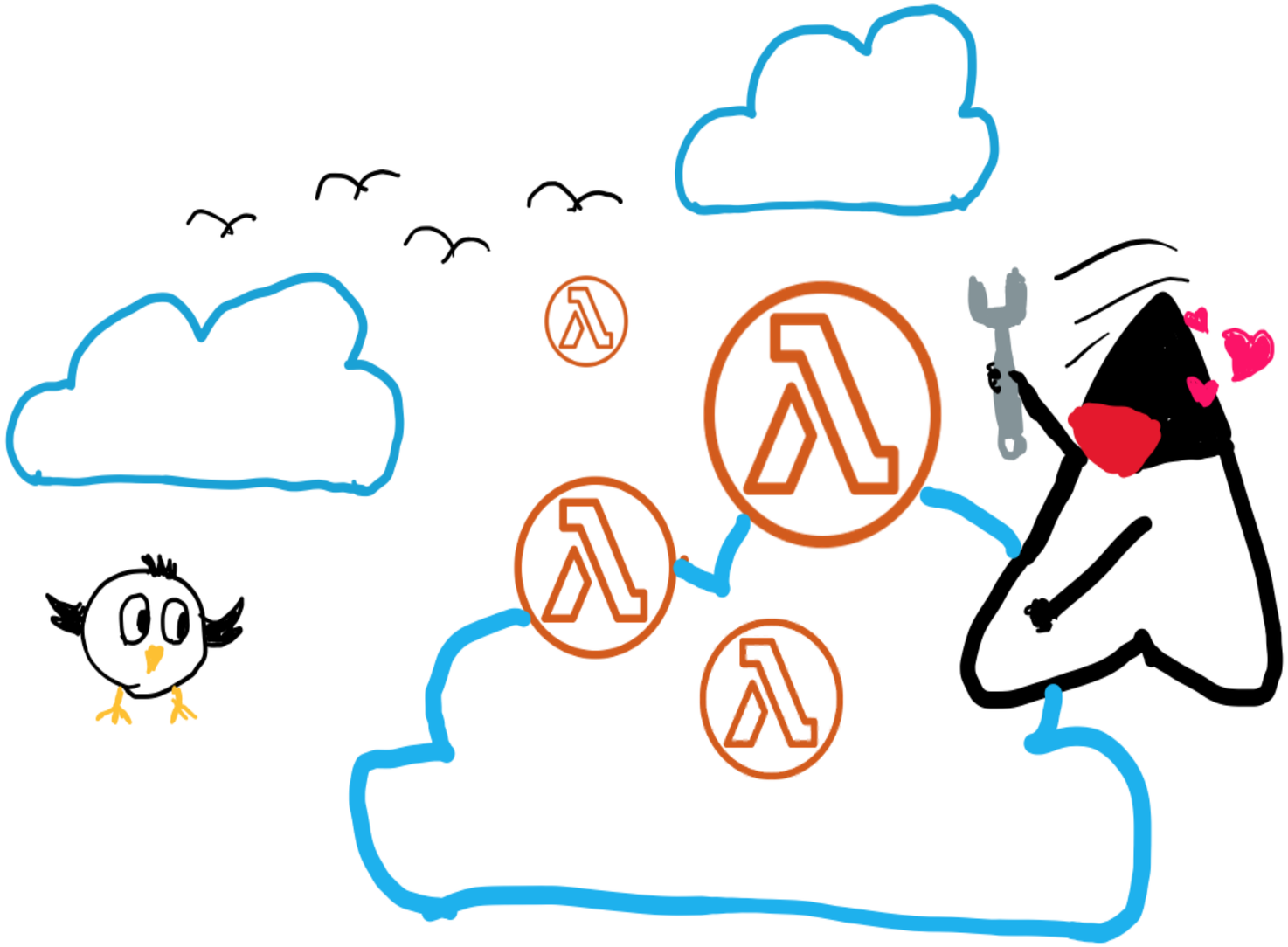
Recapitulação

1. Interface Funcional
2. Expressões Lambda
3. Predicate
4. Consumer
5. Function
6. Stream

vnt/school
powered by venturus

Stream

Expressões Lambda



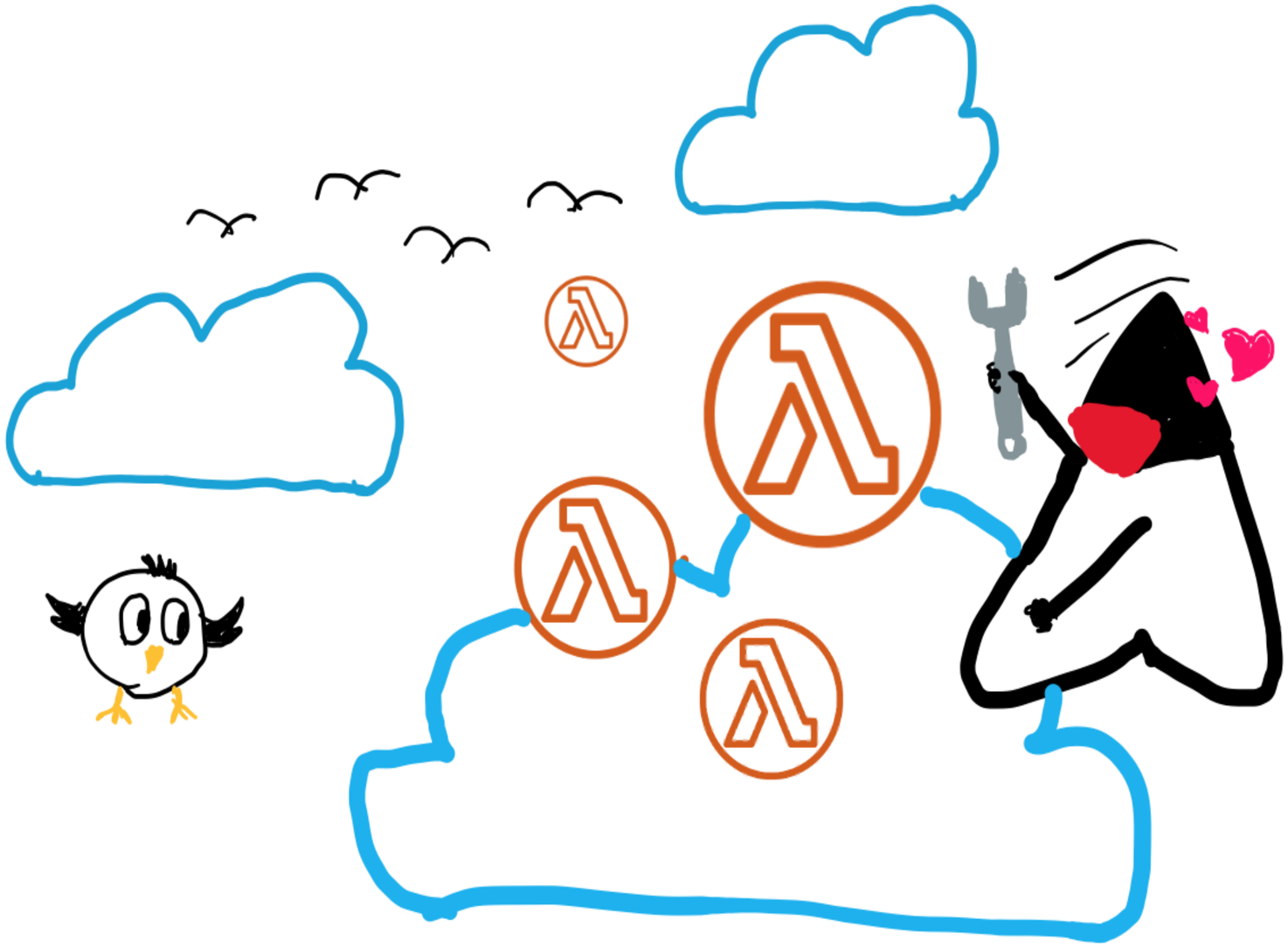
vnt/school

powered by venturus

Stream

Leia atentamente o conceito de **stream** apresentado nos próximos slides.

Em seguida, reproduza o código dado como exemplo.



vnt/school
powered by venturus

Stream

É uma **sequência de elementos** advinda de uma fonte de dados que oferece suporte a "**operações agregadas**".

Fonte de dados: coleção, array, função de iteração, recurso de E/S.

vnt/school
powered by venturus

Stream

Stream é uma solução para processar sequências de dados de
forma: Declarativa, Sem efeitos colaterais e Sob demanda.

Acesso sequencial (não há índices).

Single-use: só pode ser "usada" uma vez.

Pipeline: operações em streams retornam novas streams.

vnt/school
powered by venturus

Stream

O pipeline é composto por zero ou mais operações intermediárias e uma terminal.

Operação intermediária:

Produz uma nova stream (encadeamento).

Só executa quando uma operação terminal é invocada (lazy evaluation).

Operação terminal:

Produz um objeto não-stream (coleção ou outro).

Determina o fim do processamento da stream.

vnt/school

powered by venturus

Stream

Operações intermediárias

Filter, map, flatmap, peek, distinct, sorted, skip, limit.

Operações terminais

ForEach, forEachOrdered, ToArray, reduce

Collect, min, max , count, AnyMatch, allMatch

NoneMatch, findFirst, findAny.

vnt/school

powered by venturus

Stream

Criar uma stream:
Basta chamar o método **stream()** ou **parallelStream()** a partir de qualquer objeto Collection.

Outras formas de se criar uma stream:
Stream.of,
Stream.ofNullable,
Stream.iterate

vnt/school
powered by venturus

Stream

Stream: pode ser construída de diversas formas, veja alguns exemplos:

```
public static void main(String[] args) {
    List<Integer> list = Arrays.asList(1, 3, 5, 7, 9, 11);
    Stream<Integer> st1 = list.stream()
            .map(x -> x*10);
    System.out.println(Arrays.toString(st1.toArray()));

    Stream<String> st2 = Stream.of("Erinaldo", "Isabela", "Maria", "Joao");
    System.out.println(Arrays.toString(st2.toArray()));

    Stream<Integer> st3 = Stream.iterate(0, x -> x + 3);
    System.out.println(Arrays.toString(st3.limit(10).toArray()));
}
```

Vamos Praticar!!

Exercício

Programação Funcional

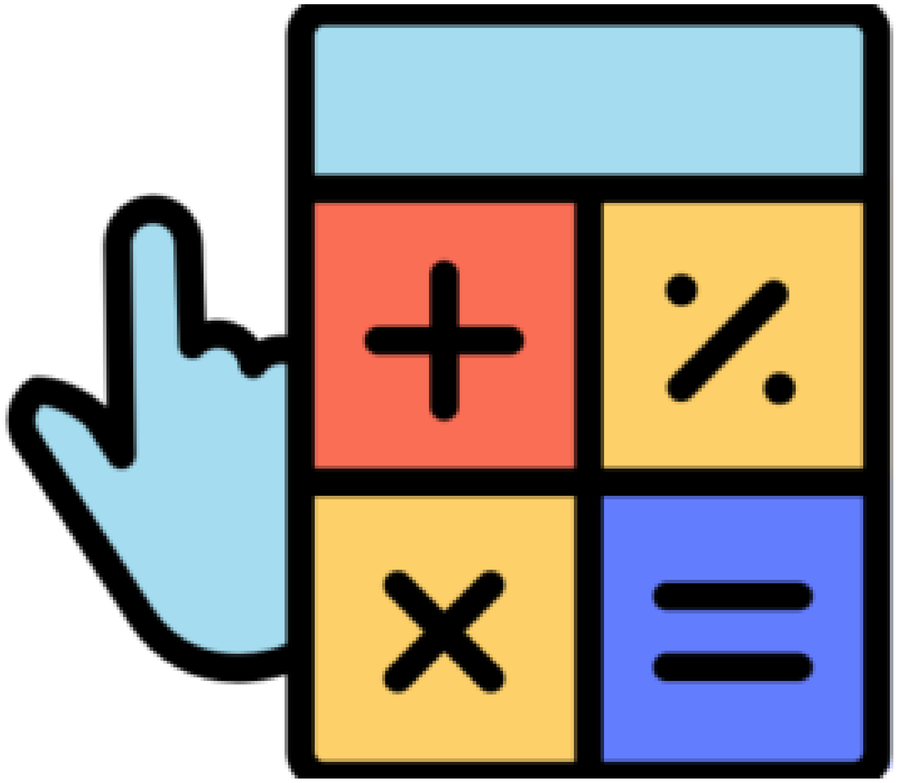
vnt/school

powered by venturus

Atividade 1

Crie uma lista de inteiros do tipo: (3, 4, 5, 10, 7, 14, 16). Em seguida crie uma Stream a partir dessa lista. Multiplique por 10 e imprima o resultado.

Em sequência crie uma nova lista a partir dessa primeira lista e a transforme numa stream. Use as funções apropriadas para filtrar os valores que são pares e multiplique os mesmos por 20. Imprima o resultado na tela.



Atividade 2

Fazer um programa para ler um conjunto de produtos a partir de um arquivo em formato .csv (suponha que exista pelo menos um produto). Em seguida mostrar o preço médio dos produtos. Depois, mostrar os nomes, em ordem decrescente, dos produtos que possuem preço inferior ao preço médio.

vnt/school
powered by venturus

Atividade 2

Crie uma arquivo .txt e cole tais informações:

Tv,900.00
Mouse,50.00
Tablet,350.50
HD Case,80.90
Computer,850.00
Monitor,290.00

Execução:

Entre com o caminho: C:\ÁreadeTrabalho\mediaPreco.txt
Media preco: 420.23
Tablet Mouse
Monitor
HD Case

vnt/school
powered by venturus

Atividade 3

Fazer um programa para ler os dados (nome, email e salário) de funcionários a partir de um arquivo em formato .csv. Em seguida mostrar, em ordem alfabética, o email dos funcionários cujo salário seja superior a um dado valor fornecido pelo usuário. Mostrar também a soma dos salários dos funcionários cujo nome começa com a letra 'M'.

vnt/school
powered by venturus

Atividade 3

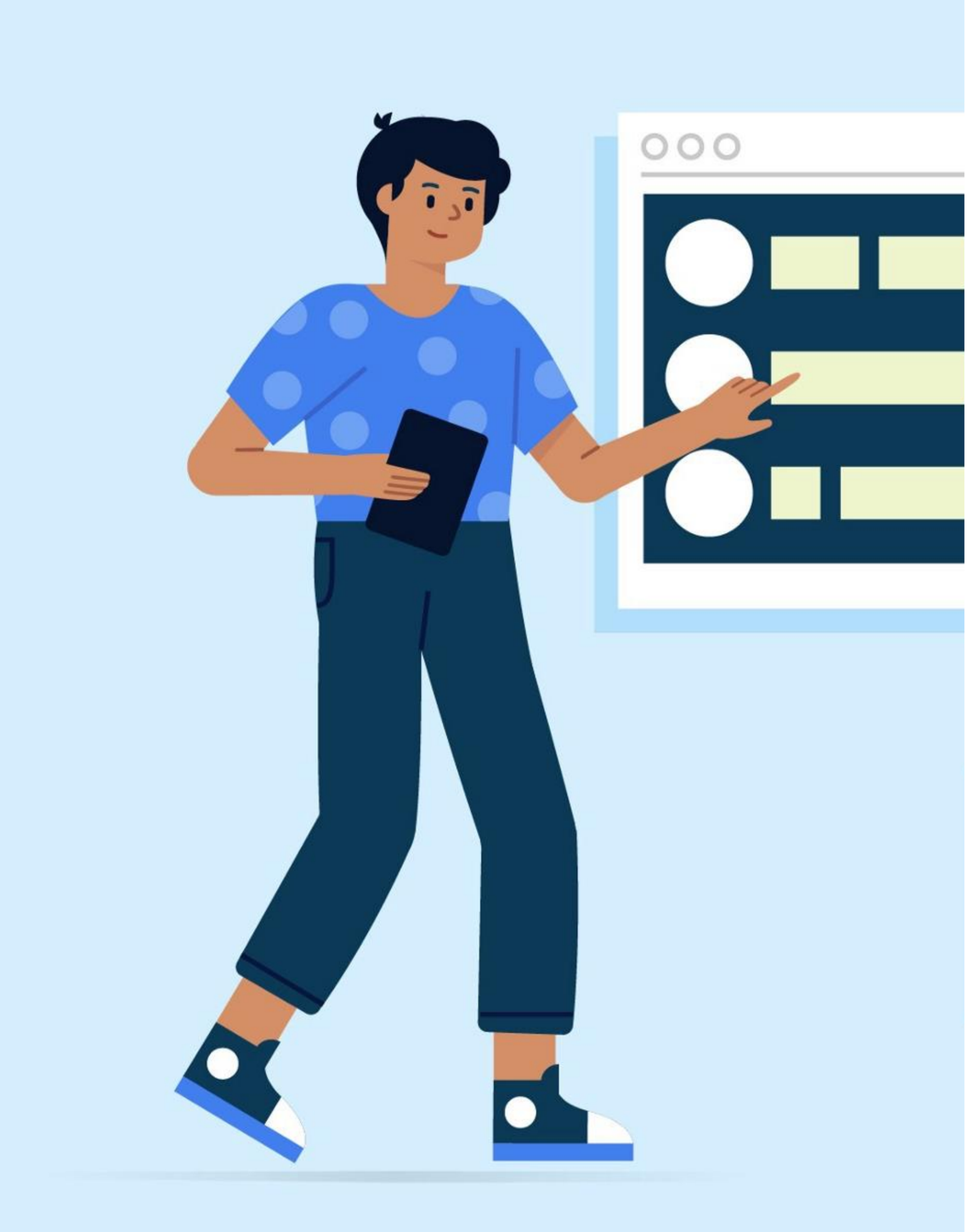
Crie uma arquivo .txt e cole tais informações:

Maria,maria@gmail.com,3200.00
Alex,alex@gmail.com,1900.00
Marco,marco@gmail.com,1700.00
Bob,bob@gmail.com,3500.00
Anna,anna@gmail.com,2800.00

Execução:

Entre com o caminho: C:\ÁreadeTrabalho\salario.txt
Entre salario: 2000.00
Email cujo salario é maior que 2000.00:
anna@gmail.com
bob@gmail.com
maria@gmail.com
Soma dos salarios das pessoas com nomes iniciam com 'M':
4900.00

vnt/school
powered by venturus



Comunidade VNT

Dica de hoje

Os dois links abaixo apresentam informações na página oficial da Oracle sobre Stream e Collections. Aproveite para conferir um pouco mais sobre o assunto.

Collection (Java SE 10 & JDK 10) (oracle.com)

Java 8: Iniciando o desenvolvimento com a Streams API | Oracle Brasil

Boa leitura!!

Referências

[1] A. Goldman, F. Kon, Paulo J. S. Silva; Introdução à Ciência da Computação com Java e Orientação a Objetos (USP). 2006. Ed. USP.
[2] Algoritmo e lógica de programação. Acessado julho/2022: https://visualg3.com.br/
[3] G. Silveira; Algoritmos em Java; Ed. Casa do Código.
[4] M. T. Goodrich, R. Tamassia; Estrutura de dados e algoritmos em Java. Ed Bookman. 2007.
[5] Algoritmo e lógica de programação. Acessado julho/2022: https://www.cursoemvideo.com/
[6] P. Silveira, R. Turini; Java 8 Pratico: lambdas, streams e os novos recursos da linguagem. Ed. Casa do Código.
[7] Linguagem Java: Curso acessado em agosto/2022: https://www.udemy.com/
[8] Linguagem Java: Curso acessado em setembro/2022: https://www.cursoemvideo.com/