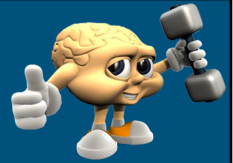


Exercício 1 - Robô



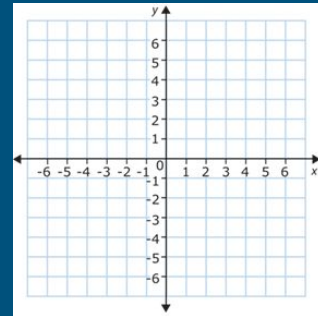
Criar a classe Robô que permite um robô ser navegado pelas direções: 'N', 'S', 'L' ou 'O'.

A classe possui coordenadas x, y que indicam a posição atual do robô no plano cartesiano, e um valor máximo no plano até o qual ele pode ser movido.

A cada chamada do método para mover o robô, ele deve se deslocar uma unidade.

Deve-se verificar se o movimento do robô é válido.

OBS: crie os métodos necessários!



emerson@paduan.pro.br

Exercício - Ebook

Considere que você está desenvolvendo um aplicativo de livro eletrônico. Os atributos e métodos estão no diagrama. As funcionalidades incluem exibir em que página está agora, para que página foi, etc.

Escreva uma classe para implementar estas funcionalidades e faça o app (main) para testar essa navegação entre as páginas.

Ebook
+ titulo: String
+ autor: String
+ totalPaginas: int
+ paginaAtual: int
+ avancarPagina(): void
+ retrocederPagina(): void
+ irParaPagina(int): void
+ exibirPagina(): int
+ mostrarCapa(): void

emerson@paduan.pro.br

Associação entre classes



emerson@paduan.pro.br

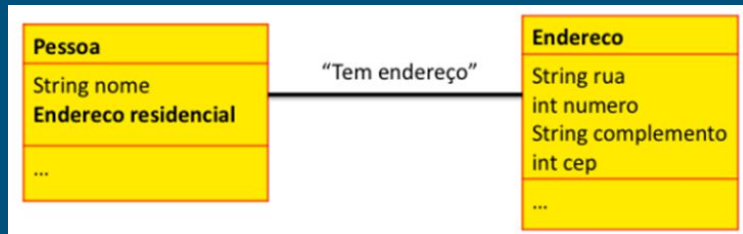
Associação

Um sistema é composto por várias Classes.

- As classes se conectam para poderem se comunicar por troca de mensagens (chamadas de métodos)
- Quando um ou mais atributos de uma Classe é uma referência para outra Classe temos uma associação.

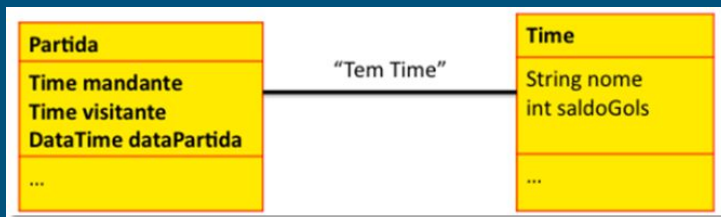
emerson@paduan.pro.br

Exemplo



emerson@paduan.pro.br

Exemplo 1: N



emerson@paduan.pro.br

Em Java

```
public class Pessoa{
    //atributos
    private String nome;
    private int idade;
    private char sexo;
    private Endereco end;
    ....

    public String imprimir(){
        return "Nome: " + nome +
            "\nIdade: " + idade +
            "\nSexo: " + sexo +
            "Endereço: " + end.imprimir();
    }
}
```

```
public class Endereco{
    //atributos
    private String logradouro;
    private String complemento;
    private int numero;
    private String cep;
    ....

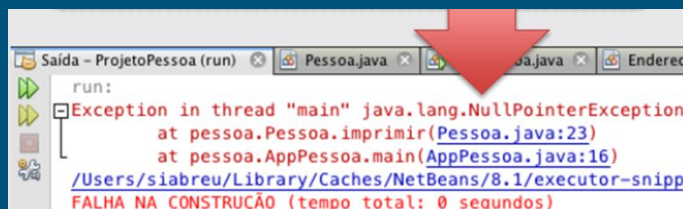
    public String imprimir(){
        return "Logradouro: " + logradouro +
            "\nComplemento: " + complemento +
            "\nNúmero: " + numero +
            "CEP: " + cep;
    }
}
```

emerson@paduan.pro.br

Atenção!!!

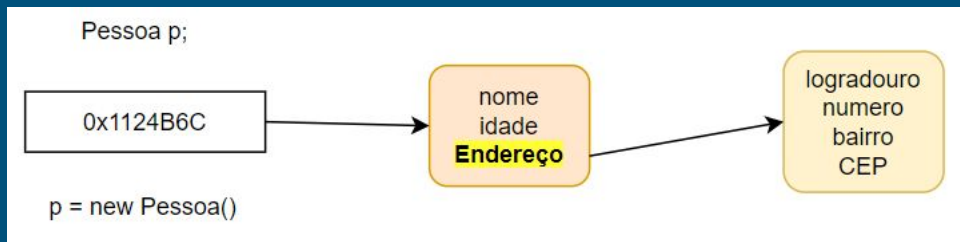
```
public class AppPessoa {
    public static void main(String[] args) {
        Pessoa objPessoa = new Pessoa();

        System.out.println(objPessoa.imprimir());
    }
}
```



emerson@paduan.pro.br

Criar objeto no construtor



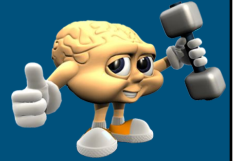
emerson@paduan.pro.br

Resolvendo

```
public class Pessoa{  
    //atributos  
    private String nome;  
    private int idade;  
    private char sexo;  
    private Endereco end;  
  
    //construtor default  
    public Pessoa() {  
        this.end = new Endereco();  
    }  
  
    public String imprimir()  
    return "Nome: " + nome +  
        "\nIdade: " + idade +  
        "\nSexo: " + sexo +  
        "Endereço: " + end.imprimir();  
}
```

emerson@paduan.pro.br

Exercício



1. Criar um relacionamento de Associação entre as classes *Animal* e *Proprietário*:
Animal "Tem UM" *Proprietário*
2. Criar a classe *Animal* com os seguintes atributos: nome, raça, ano de nascimento e proprietário
3. Criar a Classe *Proprietário* com os seguintes atributos: nome e telefone
4. Criar uma classe main

emerson@paduan.pro.br

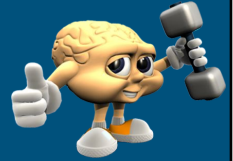
Sobrecarga (Overload)



O mesmo método, múltiplas
"funções"

emerson@paduan.pro.br

Exercício



Crie uma classe Placar, que representa o placar de um jogo de futebol. Discuta quais atributos devem ser adicionados à classe (e os tipos de dados), e faça 3 construtores:

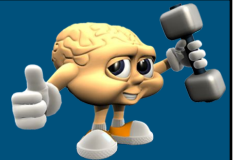
- 1 construtor padrão (default, sem parâmetros);
- 1 construtor que defina apenas quais são os times do jogo (mantendo o placar em 0 a 0);
- 1 construtor que defina os times e o placar do jogo;

Crie no main três objetos do tipo placar, um testar para cada tipo de construtor, e exiba os dados do placar no formato: *Time1 Gols X Gols Time2*.

Exemplo: São Paulo 1 X 0 Corinthians 🤪

emerson@paduan.pro.br

Exercício Funcionário

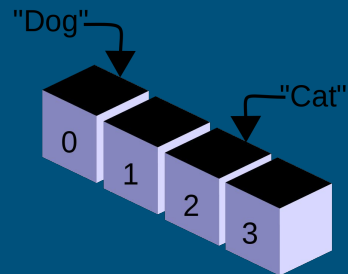


Construir uma classe para representar um funcionário com os seguintes atributos: nome, horas trabalhadas e valor pago por hora trabalhada. Implementar um método para calcular e retornar o salário final de um funcionário e um método para exibir os dados.

Criar uma classe AppFuncionario para instanciar objetos da classe Funcionario e realizar chamada a seus métodos.

emerson@paduan.pro.br

Vetores



Armazenando como conjunto.

emerson@paduan.pro.br

O que é ?

Estrutura de Dados capaz de armazenar um conjunto de dados de um mesmo tipo (Homogêneo).

É uma estrutura de tamanho fixo (estático) na criação.

emerson@paduan.pro.br

Aplicação

Como armazenar as notas de dez alunos?

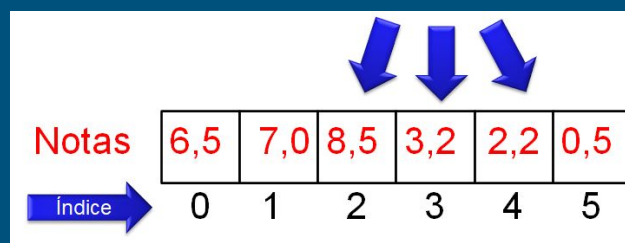
Opções:

Declarar 10 variáveis

Criar um vetor com 10 posições

emerson@paduan.pro.br

Visual



emerson@paduan.pro.br

em Java

Um vetor é declarado definindo-se seu nome (*plural*), tipo e tamanho.

```
int [ ] idades; // declara um vetor chamado idades para armazenar inteiros
```

```
idades = new int [10]; // aloca (cria) o vetor com 10 posições
```

Pode ser feito:

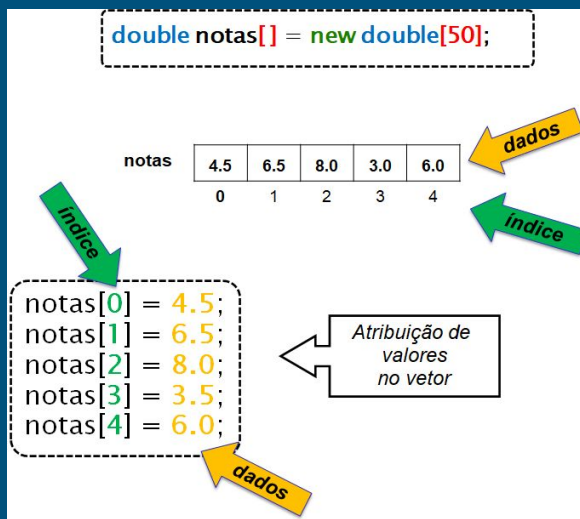
```
int [ ] idades = new int[10];
```

Você também encontrará:

```
int idades[ ] = new int[10];
```

emerson@paduan.pro.br

usando o índice



Para saber o tamanho do vetor é possível usar:

`vetor.length`

emerson@paduan.pro.br

Exemplo 01

Lendo um conjunto de notas e exibindo os dados do vetor:

```
for( i = 0; i < notas.length; i++){  
    System.out.println("Digite uma nota:");  
    notas[i] = entrada.nextDouble();  
}
```

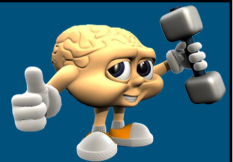
```
for( i = 0; i < 5; i++){  
    System.out.println("Nota: " + notas[i]);  
}
```

4.5	6.5	8.0	3.5	6.0
0	1	2	3	4

Nota: 4.5
Nota: 6.5
Nota: 8.0
Nota: 3.5
Nota: 6.0
CONSTRUÍDO COM SUCESSO

emerson@paduan.pro.br

Exercício



Escreva um programa que leia os nomes de 7 pessoas e armazene em um vetor.

A seguir, mostre os nomes em ordem inversa ao que foram digitados.

Exemplo:

Digitado: Marcos, João, . . ., Samanta, Felipe

Exibido: Felipe, Samanta, . . ., João, Marcos

emerson@paduan.pro.br