

SISTEMAS DE INFORMAÇÃO

DISCIPLINA: Prog. Orientada a Objetos I

Professor: Clávison M. Zapelini

P00

Introdução

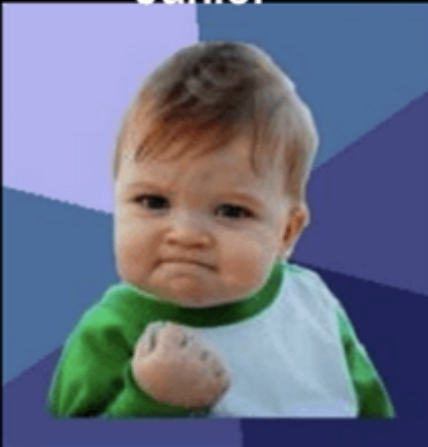


How to differentiate junior dev from senior dev

After 1st compilation ==>

```
-----  
0 Warning(s)  
0 Error(s)
```

Junior



“Yes! Made it!”

Senior



“That seems too good to be true..maybe
there’s some hidden bugs in it..”

O QUE VAMOS UTILIZAR DURANTE AS AULAS?



<https://github.com/>

Vamos criar nossa
conta...

O QUE VAMOS UTILIZAR DURANTE AS AULAS?



O QUE VAMOS UTILIZAR DURANTE AS AULAS?

IDEs de desenvolvimento



Visual Studio Code

O QUE VAMOS UTILIZAR DURANTE AS AULAS?

Testes Unitários

JUnit

O QUE VAMOS UTILIZAR DURANTE AS AULAS?

Maven (<https://maven.apache.org/>)

O Apache Maven: ferramenta de automação e gerenciamento de projetos Java. Gerencia as dependências do projeto (web e próprias).

Isso garante que sejam quantas forem as equipes no projeto, as bibliotecas vão ser as mesmas, os comportamentos vão ser os mesmos ("**funciona na minha máquina!**").



<https://www.devmedia.com.br/introducao-ao-maven/25128>



O QUE VAMOS UTILIZAR DURANTE AS AULAS?

O que eles temem:



Vampiro



Sol



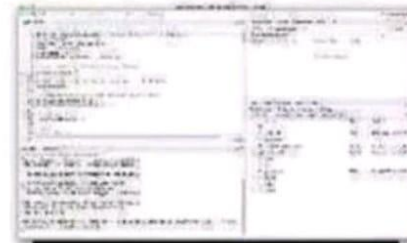
Superman



Kriptonita



Programador



Tema Branco

Faça o melhor que puder!!!

Faça um programa que leia o nome, sexo, peso e altura de N pessoas.

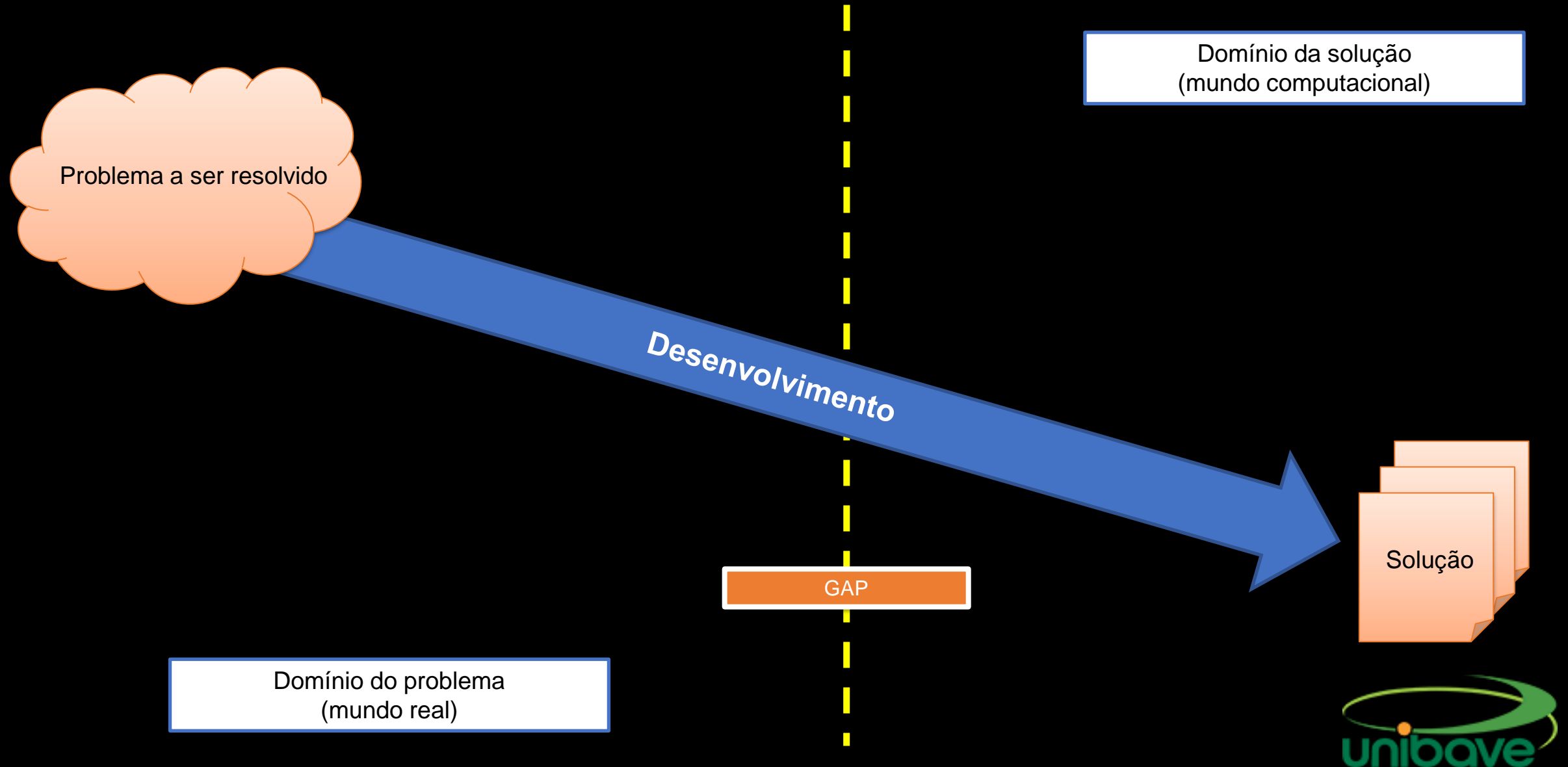
Após o cadastro, exiba o nome, o IMC e a avaliação do IMC de todas as pessoas.

Utilize a tabela de **IMC adultos - com distinção de sexo** disponível em:

<https://indicedemassacorporal.com/tabela-imc.html>



POO (introdução)



POO (introdução)

O que é um Objeto ?

Três características principais:

Algo que tenha uma identidade

Algo que possua propriedades (estados)

Algo que possa realizar coisas (comportamento)

Conta bancária?

Pessoa?

Dor de dente?

POO (introdução)

OBJETOS

Tudo o que existe é objeto

Um programa é um conjunto de objetos “conversando” uns com outros por meio de mensagens

Todo objeto tem um tipo (**classe**)

Todos objetos de um mesmo tipo (classe) podem receber/enviar as mesmas mensagens.



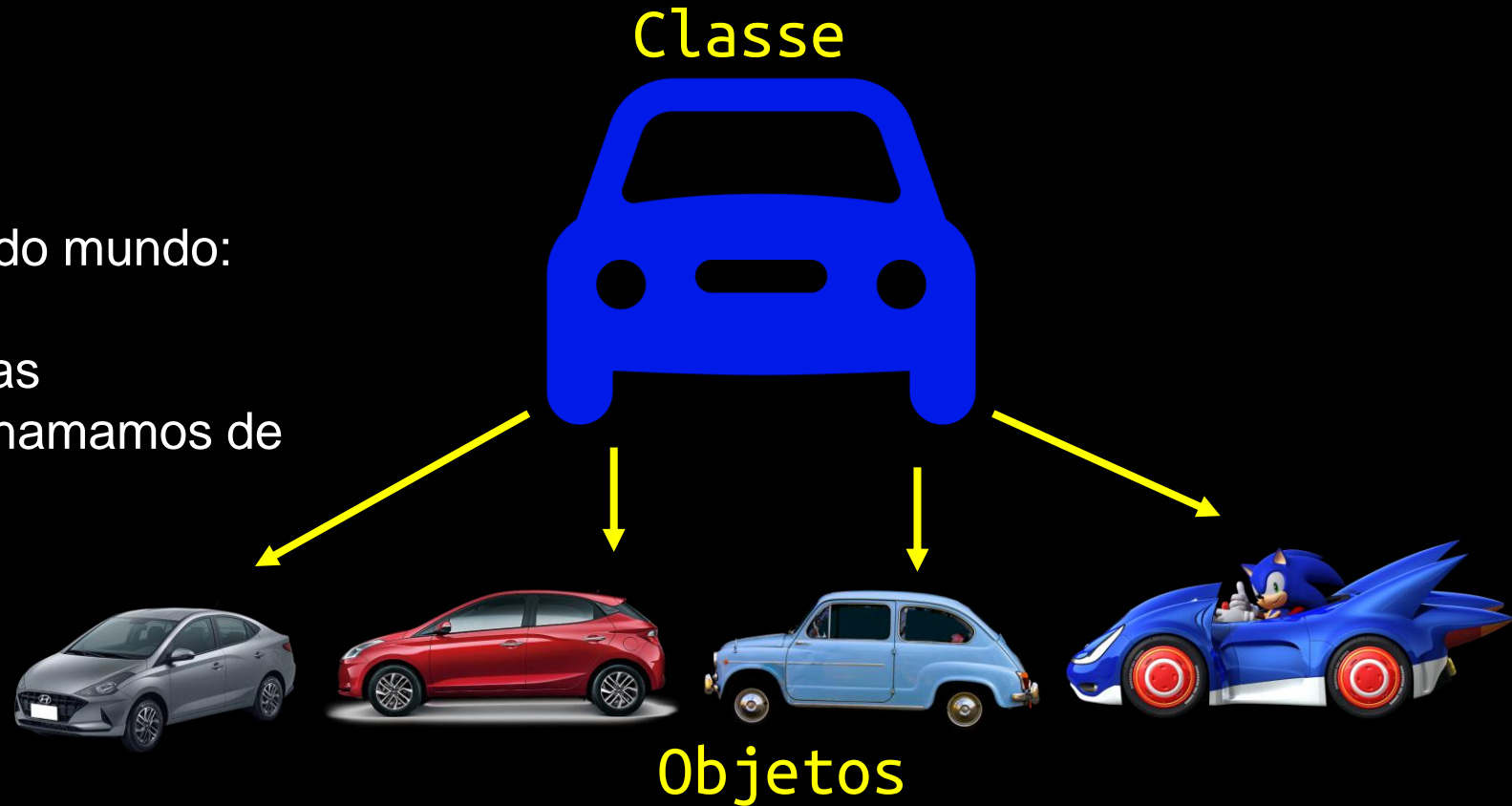
POO (introdução)

CLASSES

O ser humano classifica os objetos do mundo:

Juntamos os objetos com as mesmas características em categorias que chamamos de classes.

Pessoas em uma festa
Produtos
Empresas
Animais



Todas as contas de um banco tem saldo, número, correntista, mas cada conta é única!!

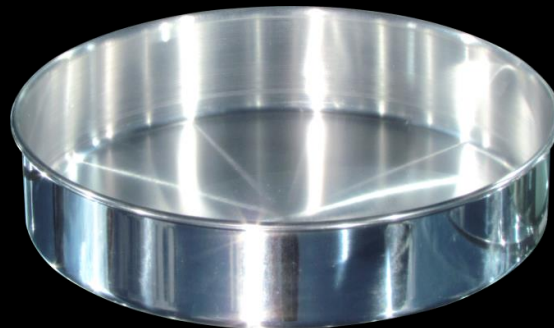
POO (introdução)

Classes são utilizadas por linguagens OO para modelar tipos compostos, ou ainda tipos abstratos de dados TADs que definem os objetos de uma classe.

Um objeto é uma instância de uma classe

Para criar um novo objeto é necessário especificar de qual classe.

Em java o operador responsável por criar um objeto de uma classe é: new



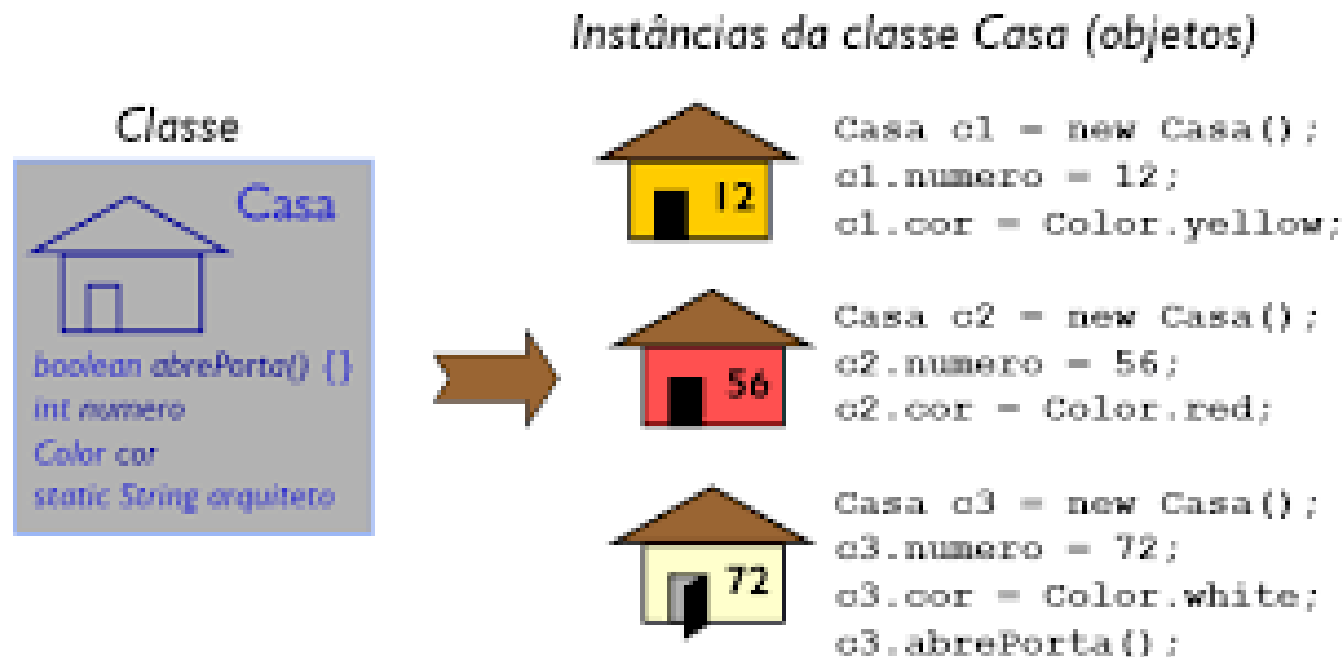
POO (introdução)

Membros de uma classe

Uma classe pode ter dois tipos de membros:

Atributos (variáveis)

Métodos (funções e procedimentos)



Atributos são como partes de um tipo composto:

Celular (bateria, tecla, visor....)

Métodos é o que é executado no contexto de uma classe:

Celular(ligar/desligar, despertar ...)

POO (introdução)

Membros de uma classe

Atributos

São definidos como variáveis no escopo de uma classe

```
public class Pessoa {  
    String nome;  
    double peso, altura;  
}
```

Acesso via operador “.”

```
Pessoa p1 = new Pessoa();  
p1.nome = "Fulano de Tal";  
p1.altura = 1.74;  
p1.peso = 65;
```


POO (introdução)

Membros de uma classe

Atributos podem ser referências a objetos

```
public class Departamento{  
    String nome;  
    Pessoa gerente;  
}
```

```
Departamento d = new Departamento();  
d.nome = "RH";  
d.gerente = new Pessoa();  
d.gerente.nome = "Maria";  
d.gerente.altura = 1.60;  
d.gerente.peso = 55;
```

POO (introdução)

Valores default para atributos

Um atributo pode ser inicializado

```
public class Pessoa {  
    double peso = 82;  
    double altura = 1.82;  
}
```

Quando não inicializamos explicitamente, um valor default é atribuído:

Tipo de dado	Valor default
boolean	false
char	'\u0000'
byte	0
short	0

Tipo de dado	Valor default
int	0
long	0
float	0.0
double	0.0

Atributos primitivos

POO (introdução)

O valor null

O valor default para atributos referência (Objetos) é **null**.

Um objeto nulo é uma referência que não aponta para nenhum objeto (não existe na memória).

Usar propriedades de uma referência nula, gera uma exceção: **NullPointerException**

```
Departamento d = new Departamento();  
d.nome = "RH"; //OK  
d.gerente.nome = "Maria"; // NullPointerException  
d.gerente.altura = 1.60;  
d.gerente.peso = 55;
```

POO (introdução)

Métodos

Um método é uma função/procedimento que opera no contexto de uma classe (mensagem que o objeto recebe ou responde).

```
public class Principal {  
    public static void main(String[] args) {  
        Pessoa p1 = new Pessoa();  
        p1.nome = "Fulano de Tal";  
        p1.altura = 1.74;  
        p1.peso = 65;  
        p1.calculaImc();  
    }  
}
```

calculaImc() é a mensagem que o objeto p1 pode receber.

Qualquer outra instância da classe Pessoa é capaz de receber esta mensagem.

O método de P1 poderá chamar mensagens de outros objetos.

POO (introdução)

Parâmetros e valor de retorno de métodos

Um método pode aceitar parâmetros

```
void calculaSalario(int horas, double vlHora) {  
    double salario = horas * vlHora;  
    JOptionPane.showMessageDialog(null, salario);  
}
```

Um método pode retornar um valor

```
double calculaSalario(int horas, double vlHora) {  
    double salario = horas * vlHora;  
    return salario;  
}
```

Os parâmetros e retorno dos métodos, podem ser tipos primitivos ou referências (objetos).

POO (introdução)

Definição de um Método

Para definir um método deve ser especificado no mínimo:

Seu tipo de retorno

Seu nome

Seus parâmetros

Sua implementação

```
tipoDeRetorno nome (/* lista de parâmetros */) {  
    // Corpo (implementação) do método  
}
```

POO (introdução)

Assinatura de um método

Define o método de uma forma única

Em java, o que constitui a assinatura do método é o nome e a lista de parâmetros

Não pode haver dois métodos com a mesma assinatura na mesma classe (mesmo que o tipo de retorno seja diferente)

Portanto, pode haver métodos com o mesmo nome, isso se chama **SOBRECARGA**.

POO (introdução)

Passagem de Parâmetros

Funciona como uma atribuição, desta forma: **Os tipos devem ser compatíveis**

Tipos primitivos recebem cópias de valores, não altera valor.

Tipos referência recebem a referência a memória, altera o valor

POO (introdução)

Passagem de Parâmetros

Tipo Primitivo:

```
int i = 10;
recebePrimitivo(i);
JOptionPane.showMessageDialog(null, i);

void recebePrimitivo(int vl) {
    vl = 20;
    JOptionPane.showMessageDialog(null, vl);
}
//20
//10
```

POO (introdução)

Passagem de Parâmetros

Tipo Referência:

```
Pessoa p = new Pessoa();  
p.nome = "Maria";  
recebeReferencia(p);  
JOptionPane.showMessageDialog(null, p.nome);  
  
void recebeReferencia(Pessoa pes) {  
    pes.nome = "João";  
    JOptionPane.showMessageDialog(null, pes.nome);  
}  
  
//João  
//João
```

POO (introdução)

Retorno dos métodos

Um método retorna um valor (tipo primitivo ou referência) utilizando o comando **return**.

Em métodos que não retornam valores (void) o **return** é opcional. Se utilizado indica que o método deve ser interrompido.

Da mesma forma que a passagem de parâmetros, os tipos de retorno devem ser compatíveis.

POO (introdução)



Faça um programa que possua as seguintes funcionalidades:

Cadastrar Carro (marca, ano de fabricação e cor. Para a cor utilizar ENUM).

Listar os carros por período de fabricação (ano inicial e final)

Listar carros **por marca**

Listar carros **por cor** (ENUM)

- Para todas as listagens, deve ser informado o percentual dos carros listados em relação ao total cadastrado.
- Somente poderão ser inseridos no cadastro se todos os atributos forem preenchidos.

Desenvolver a classe de teste para verificar se as funcionalidades estão corretas.

