

# TI-220 Java Orientado a Objetos

---

ANTONIO CARVALHO - TREINAMENTOS

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

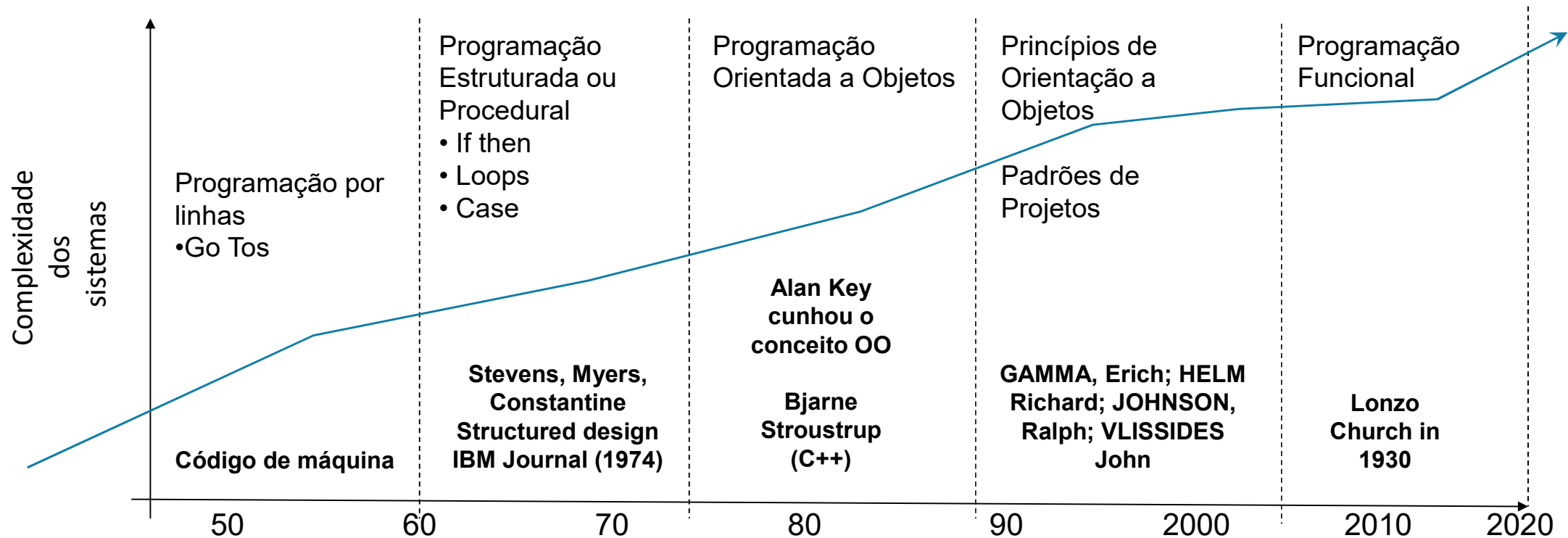
# Java Orientação a Objetos

---

# Introdução

---

# Evolução dos paradigmas de programação



# O que é Orientação a Objetos

---

A orientação a Objetos visa abstrair características de entidades concretas e abstratas do mundo real, criando padrões ou classificações que posteriormente serão transformados em objetos virtuais.

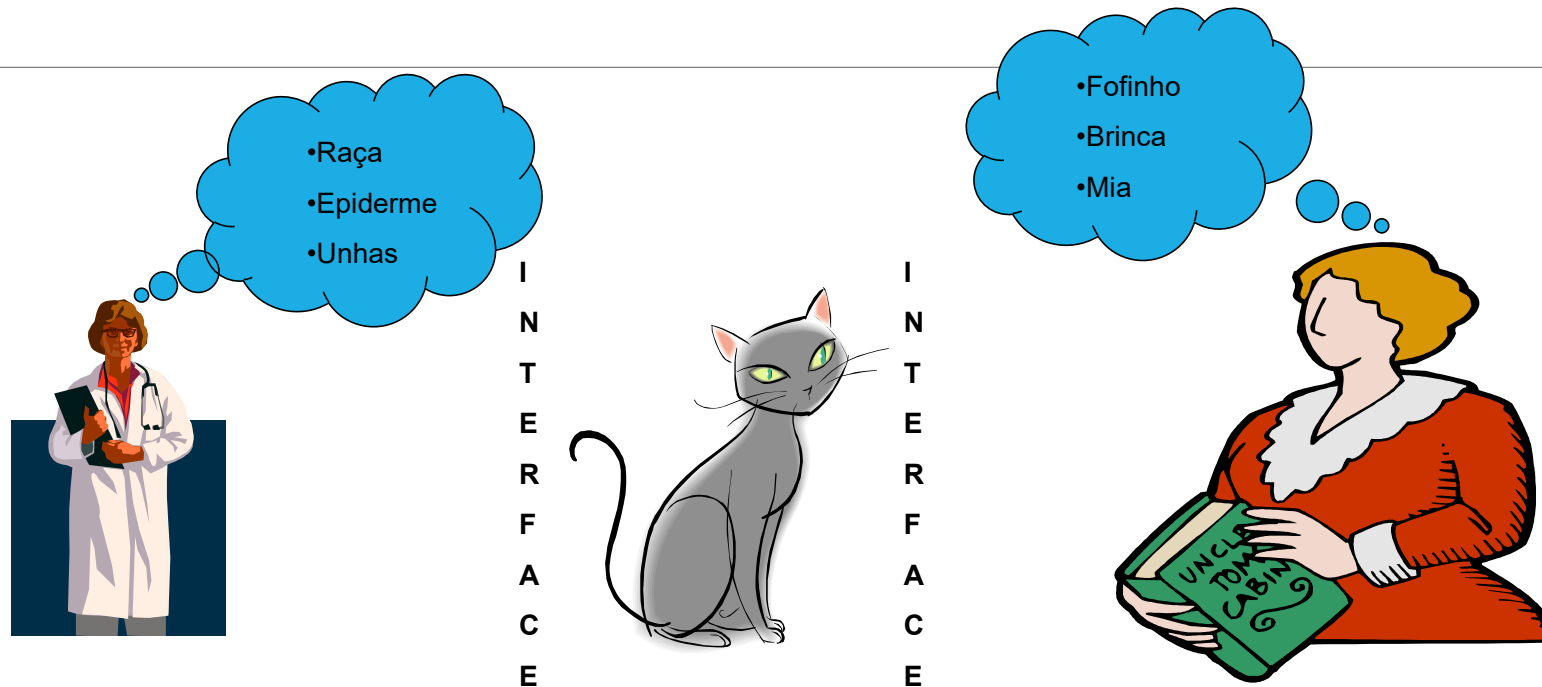
# O que é Orientação a Objetos

---

Todo objeto é composto por :

- Características, também conhecidas como atributos ou estado. As características de um objeto modificam seu estado.
- Comportamentos, também conhecidos como métodos, funções, ações. Os comportamentos são ações que o objeto pode executar.

# O que é Abstração



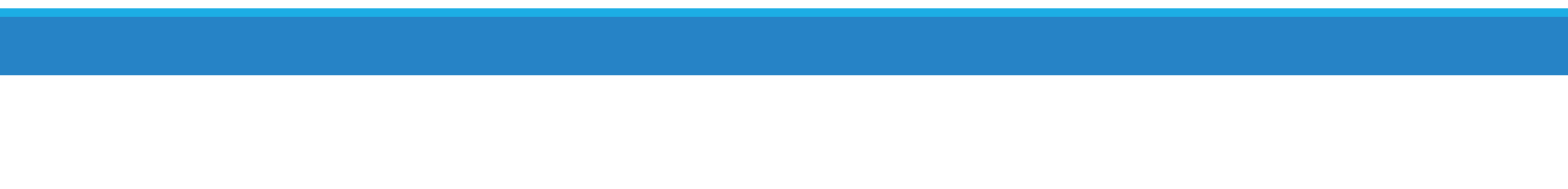
**Nota :** Duas ou mais pessoas podem ver características distintas em um mesmo objeto. O objeto comporta ambas as características porém cada uma das pessoas visualiza aquilo que mais atendem a sua realidade, mediante aos seus próprios filtros internos. Isso chama-se **abstração**.

# O que é Abstração

---

“Uma abstração é qualquer modelo que inclui os aspectos mais importantes, essenciais de alguma coisa, ao mesmo tempo em que ignora os detalhes menos importantes” (EDUARDO BEZERRA)

Através das abstrações podemos concentrar nosso foco, naquilo que realmente importa, ignorando as demais características e comportamentos.

A solid blue horizontal bar spanning the width of the slide at the bottom.



# Objetos


---

São elementos do universo, sejam físicos ou abstratos, reais ou imaginários. São únicos e guardam seu estado, mesmo que possam possuir estados idênticos ainda sim são únicos.

Os objetos podem ser representados por classes, que visam abstrair suas características e comportamentos do mundo real.

Embora únicos os objetos da mesma classe possuem os mesmos tipos de estados e comportamentos.

**Nota:** o resultado do comportamento pode ser diferente em cada objeto, devido ao seu estado.



# Objetos

---

**Objeto** é um elemento concreto de um tipo de classe.

Fusca (1)      Fusca (2)      Fusca (3)

↓      ↓      ↓



# Classe

---

Classe é uma especificação dos tipos de estado e comportamentos dos objetos suportados. (SIERRA, 2008 – Tradução adaptada pelo Autor)

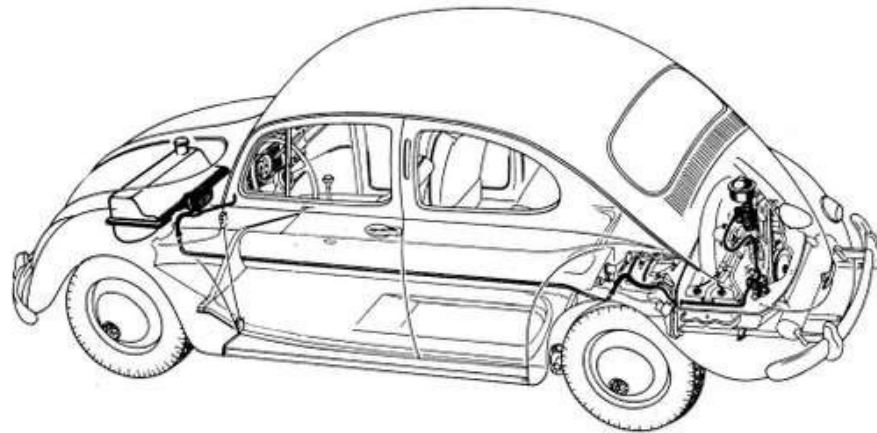
A classe é uma abstração do objeto (seja físico ou abstrato), através dela é possível descrever o objeto sob a ótica das características e comportamentos.

# Classe

---

Através da classe é possível descrever os atributos e comportamentos dos objetos.

**SISTEMA DE ALIMENTAÇÃO**



# Classe - Sintaxe

---

A sintaxe para criar uma classe é :

```
<modificador> class <nome da classe> {  
    <modificadores> <tipo> <nome do atributo1 >;  
    <modificadores> <tipo> <nome do atributo2 >;  
    ..  
    <modificadores> <tipo> <nome do atributoN>;  
  
    <modificadores> <tipo de retorno> <nome do método 1> ( <parâmetros> ) {  
        <Código a ser executado linha 1>;  
        <Código a ser executado linha 2>;  
        ...  
    }  
    <modificadores> <tipo de retorno> <nome do método N> ( <parâmetros> ) {  
        <Código a ser executado linha 1>;  
        <Código a ser executado linha 2>;  
        ...  
    }  
}
```

# Classe - Exemplo

---

| Gato                                |
|-------------------------------------|
| +tamanho : float<br>+peso : float   |
| +brincar() : void<br>+miar() : void |

```
public class Gato {  
    public float tamanho;  
    public float peso;  
    public void brincar() {  
        System.out.println("Estou brincando");  
    }  
  
    public void miar() {  
        System.out.println("miau ...");  
    }  
}
```

# Classe - Exercício

---

A partir de objetos **comuns**, crie 3 classes, abstraindo ao menos 5 características e 3 comportamentos para cada uma.

Crie o código em Java para estas classes



# Classe - Instanciando

---

- Instanciar uma classe é o ato de criar um objeto a partir dela.
- A classe contém os **tipos** das características e os comportamentos que os objetos vão possuir.
- Mas cada objeto possuirá **conteúdos** diferentes em suas características, e o comportamento de cada um irá variar conforme estes conteúdos.



# Classe - Instanciando

---

- A sintaxe para instanciar uma classe é:

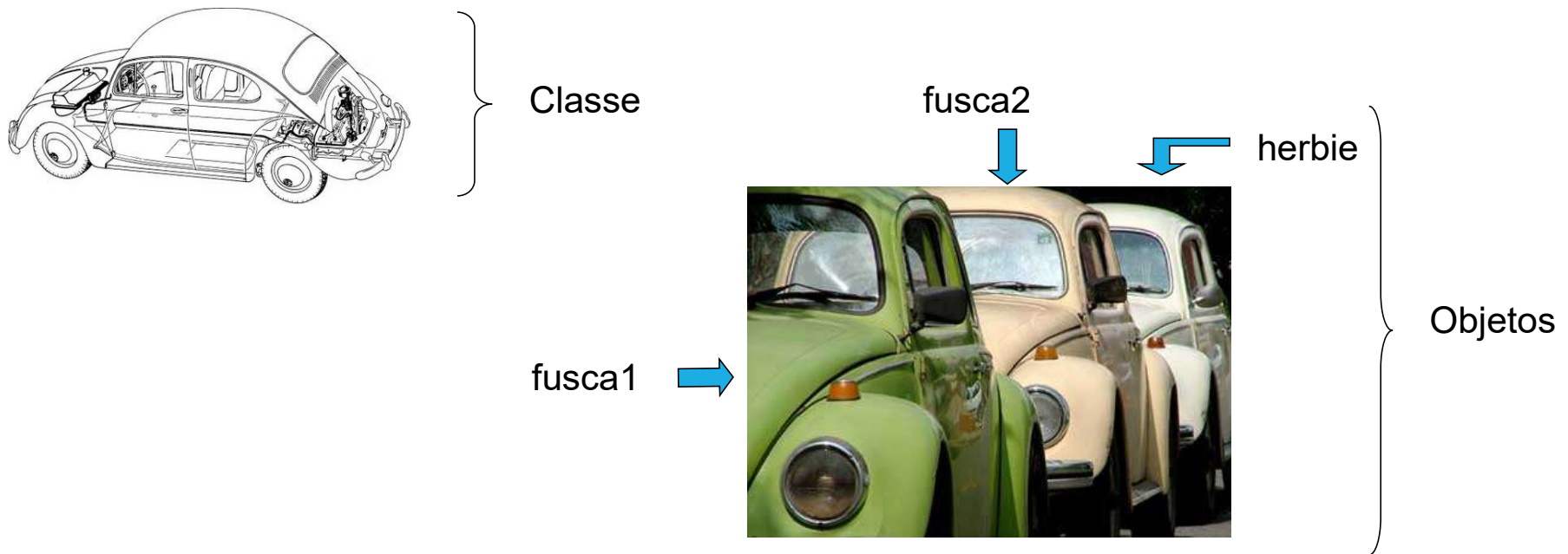
```
<tipo da variável> <nome da variável> = new <nome da classe>(<parâmetros>);
```

- Exemplo:

// Para instanciar um objeto do tipo Gato usa-se :

```
Gato felix = new Gato();
```

# Classe - Instanciando



A partir de uma mesma classe é possível criar diversos objetos.

# Classe - Instanciando

---

A classe carro mostrada na página anterior é apenas a planta (o desenho), mostrando como os objetos instanciados serão e como se comportarão:

```
public class Carro {
    float autonomia;
    int maxKmHora;
    String marca;
    String modelo;
    int ano;
    int velocidade;

    public void acelerar() {
        System.out.println(" Acelerando ... ");
        velocidade = velocidade + 10;
    }

    public void frear() {
        System.out.println(" Freando ... ");
        velocidade = velocidade - 10;
    }
}
```

Para se criar os objetos a partir da classe é preciso instanciá-los, conforme no exemplo abaixo:

```
// Para instanciar um objeto do tipo Carro usaremos :
Carro fusca1 = new Carro ();
Carro fusca2 = new Carro ();
Carro herbie = new Carro ();
```

# Objetos Instanciados

---

A partir do momento em que a classe for instanciada, o Java criará mais um objeto na memória, que poderá conter **informações** diferentes em suas **características** (atributos).

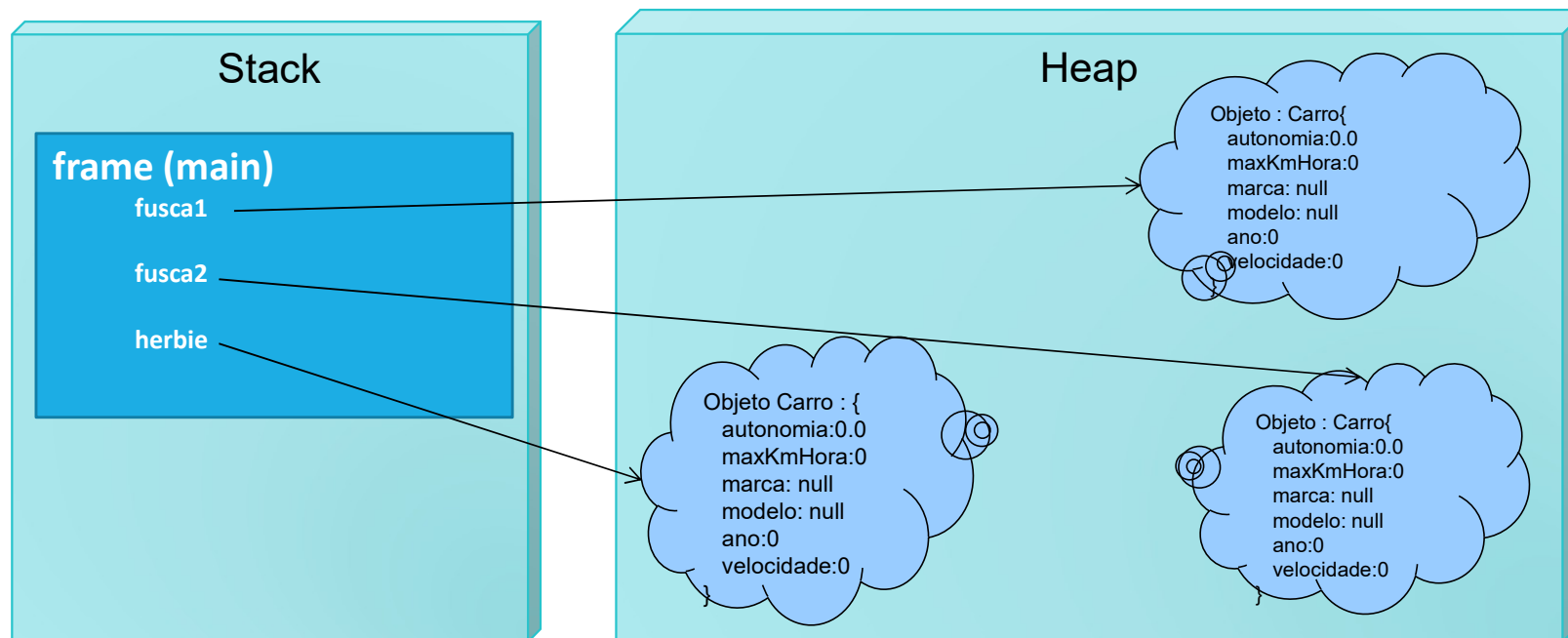
Para alterar o conteúdo de uma **característica** do objeto é preciso referenciar seu atributo e atribuir uma nova informação.

**Lembrem-se** atributos e características são a mesma coisa.



# Objetos Instanciados

Objetos criados na memória a partir do momento em que são instanciados



# Objetos Instanciados

---

A sintaxe para atribuir uma informação para uma característica do objeto é :

**<nome do objeto>.<nome do atributo> = <novo valor>;**

Exemplo:


```
herbie.autonomia = 398.5f;
```

Para utilizar a característica de um objeto basta mencioná-la conforme abaixo :

**<nome do objeto>.<nome do atributo>**

Exemplo:

```
System.out.println( herbie.autonomia );
```



# Objetos Instanciados

---

Quando o código abaixo for executado, os objetos instanciados na memória sofrerão alteração no estado.

As variáveis de instância que não forem inicializadas, receberão valores padrões conforme o tipo:

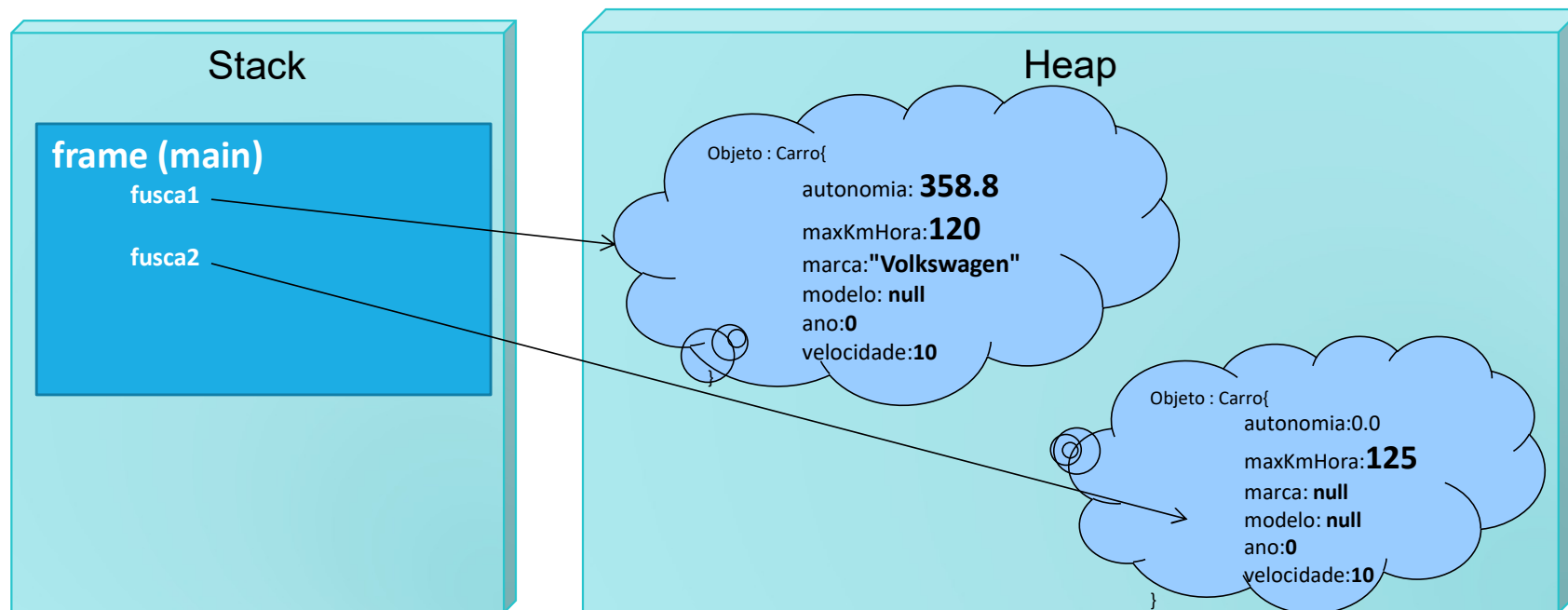
- byte, short, int, long, float e double receberão o valor **0**
- boolean receberá **false**
- char receberá o caractere **'\u0000'**
- demais objetos receberão **null**

Código:

```
Carro fusca1 = new Carro();
Carro fusca2 = new Carro();
fusca1.autonomia = 358.8f;
fusca1.maxKmHora = 120;
fusca2.maxKmHora = 125;
fusca1.marca = "Volkswagen";
fusca1.acelerar();
fusca2.acelerar();
```

# Objetos Instanciados

Objetos são criados na memória a partir do momento em que são instanciados





# Objetos - Exercício

---

Com base nas classes criadas no exercício anterior, instancie **2** objetos para cada classe, e escreva códigos para modificar suas características e executar seus comportamentos.

# Dúvidas

---



# Quiz Time

---

## **Kahoot**

POO – Orientação a Objetos 1

### **Acessar o link**

<https://play.kahoot.it/v2/?quizId=b7cfe37d-8d7d-4b57-b897-75ed64bac870>