

ECM251 - Linguagens I

Teória - Introdução a POO com Java

Prof. Murilo Zanini de Carvalho

Prof. Tiago Sanches da Silva

Antes de começar!

Clone seu repositório do Github

- Lembre-se sempre antes de iniciar uma aula, clonar seu repositório remoto e realizar as atividades nele.
- Para cada atividade desenvolvida, criar um novo diretório.

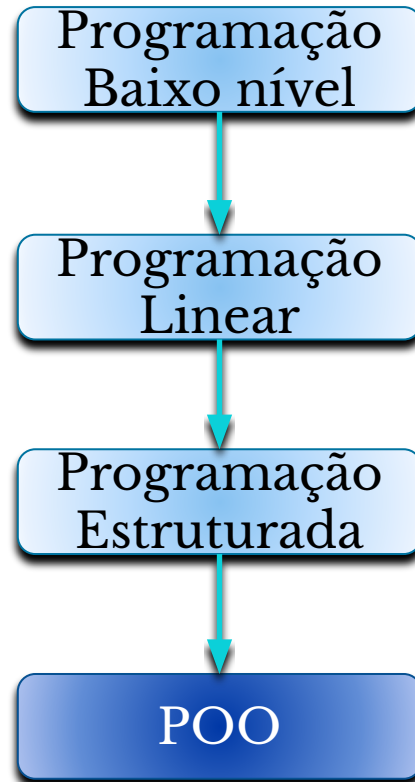


GitHub

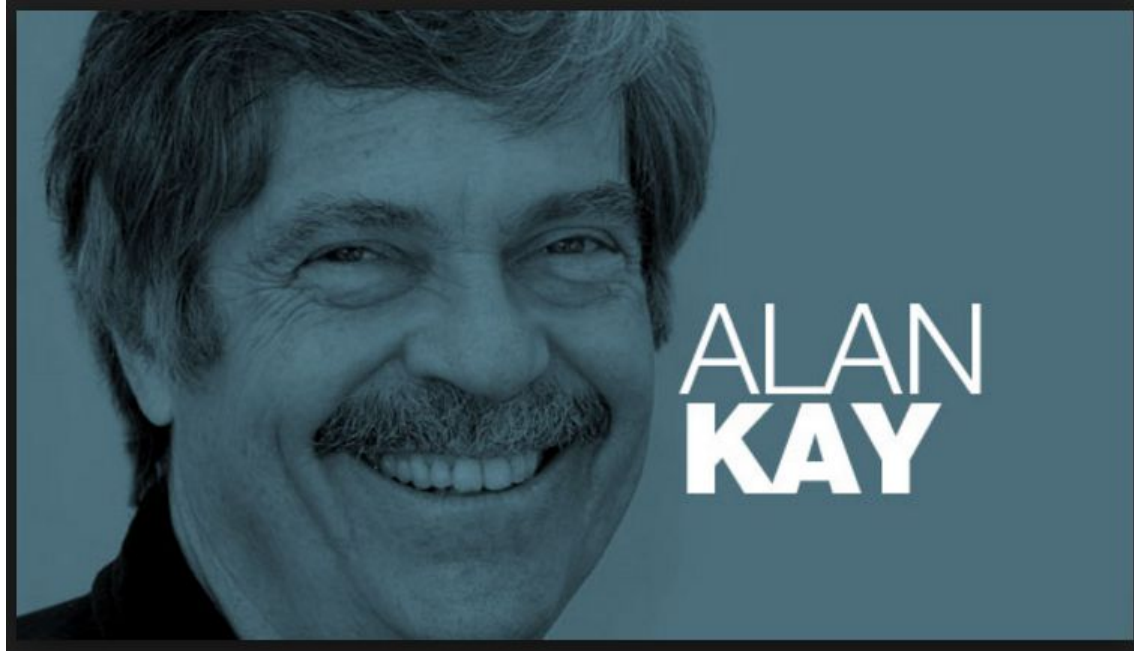
Retirado de
(https://miro.medium.com/max/4000/0*MZMI76wKo2FQLqG0.png), em 07/03/2021

Introdução a Orientação a Objetos

Como era?



Quem criou?



Matemático e Biólogo

<http://techland.time.com/2013/04/02/an-interview-with-computing-pioneer-alan-kay/>

Vantagens do POO

Confiável: Isolamento entre os objetos (partes do SW) gera um sistema seguro.

Oportuno: A divisão em partes possibilita o desenvolvimento paralelo.

Extensível: Software não é estático ele deve crescer ao longo do tempo.

Reutilizável: Aproveite as classes já criadas em outros projetos.

Natural: Mais fácil de entender. Atenção nas funcionalidades e não na implementação.

Que tal vermos essas vantagens mais tarde novamente?

Defina o que é um objeto



Coisa **material** ou **abstrata** que pode ser percebida pelos sentidos ou descrita por meio das suas **características**, **comportamento** e **estado atual**.

Mude seu jeito de pensar

Resista a vontade de pensar em um programa de forma estruturada.

Evite o pensamento: “Não estou entendendo, prefiro a estruturada!”

Tente de verdade enxergar um mundo cheio de objetos!
Mantenha sua mente aberta!!!!



Descreva alguns objetos!

Características e o que fazem.



Características

Modelo
Cor
Ponta
Carga

O que faz?

Escrever
Rabiscar
Pintar
Tampar
Destampar



Descreva alguns objetos!



reunião



Pessoa

Mão na massa
pessoal!

Ex.

Características



O que faz?



Caneta

- Modelo
- Cor
- Ponta
- Carga
- Escrever
- Rabiscar
- Pintar
- Tampar
- Destampar

Classes e objetos

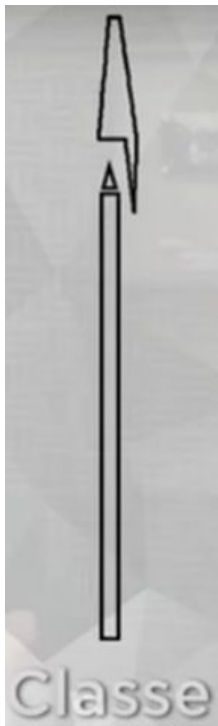
Caneta	
-	modelo
-	cor
-	ponta
-	carga
-	Escrever
-	Rabiscar
-	Pintar
-	Tampar
-	Destampar



INSTÂNCIA

Caneta azul
modelo esferográfica
cor azul,
ponta 0.5
carga 95%

Classes e objetos



Caneta	
-	modelo
-	cor
-	ponta
-	carga
-	Escrever
-	Rabiscar
-	Pintar
-	Tampar
-	Destampar

← Nome da classe

← **Atributos** dessa classe

← **Métodos** dessa classe

```
Classe Caneta
modelo: Caractere
cor: Caractere
ponta: Real
carga: Inteiro
tampada: Logico
Metodo rabiscar()
    Se (tampada) entao
        Escreva("ERRO")
    senao
        Escreva("Rabisco")
    FimSe
FimMetodo
Metodo tampar()
    tampada = verdadeiro
FimMetodo
FimClasse
```

Paradigma Programação Orientado a Objeto

Allan Kay, um dos criadores do SmallTalk:

- *Tudo* são objetos;
- Um *programa* é um grupo de objetos enviando mensagens uns aos outros.

Paradigma Programação Orientado a Objeto

No paradigma estruturado, temos procedimentos (ou funções) que são aplicados globalmente em nossa aplicação.

No caso da orientação a objetos, temos métodos que são aplicados aos dados de cada objeto.



Os quatro pilares da POO

Veremos em detalhes cada uma delas no momento apropriado.

Abstração

Herança

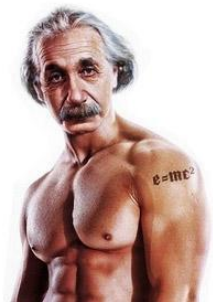
Encapsulamento

Polimorfismo

Abstração

Esse pilar, como o próprio nome diz, visa abstrair algo do mundo real e transforma-lo em um objeto na programação.

Identifique quais características e ações do objeto, são relevantes para o cenário (SW/Sistema).



Aluno

?

Modelo

?

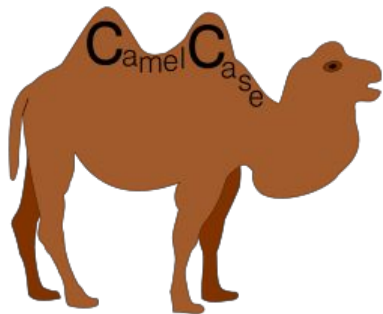
Cliente

?

Abstração

Padronização!

Camel Case?



CamelCase é a denominação em inglês para a prática de escrever palavras compostas ou frases, onde cada palavra é iniciada com Maiúsculas e unidas sem espaços. É um padrão largamente utilizado em diversas linguagens de programação, como Java, C#, Ruby, PHP e Python, principalmente nas definições de Classes e Objetos.

Leitura em casa:

- <http://java-hunters.blogspot.com.br/2014/12/o-padrao-camelcase.html>
- <https://codeflavor.wordpress.com/2010/03/23/camelcase-o-que-e-porque-usar/>

Java é *case-sensitive*!

Em língua portuguesa, significa algo como "sensível à caixa das letras" ou "sensível a maiúsculas e minúsculas".

Arroz != arroz

Int != int

vendaMensal != VendaMensal != vendamensal

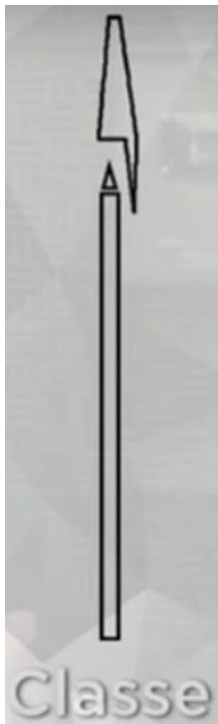
Tipos primitivos no Java

Tipo	Descrição
<code>boolean</code>	Pode assumir o valor <code>true</code> ou o valor <code>false</code>
<code>char</code>	Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535.
<code>byte</code>	Inteiro de 8 bits em notação de complemento de dois. Pode assumir valores entre $-2^7 = -128$ e $2^7 - 1 = 127$.
<code>short</code>	Inteiro de 16 bits em notação de complemento de dois. Os valores possíveis cobrem a faixa de $-2^{15} = -32.768$ a $2^{15} - 1 = 32.767$
<code>int</code>	Inteiro de 32 bits em notação de complemento de dois. Pode assumir valores entre $-2^{31} = -2.147.483.648$ e $2^{31} - 1 = 2.147.483.647$.
<code>long</code>	Inteiro de 64 bits em notação de complemento de dois. Pode assumir valores entre -2^{63} e $2^{63} - 1$.
<code>float</code>	Representa números em notação de ponto flutuante normalizada em precisão simples de 32 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável por esse tipo é $1.40239846e-46$ e o maior é $3.40282347e+38$
<code>double</code>	Representa números em notação de ponto flutuante normalizada em precisão dupla de 64 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável é $4.94065645841246544e-324$ e o maior é $1.7976931348623157e+308$

Equivalentes em classes

- `byte` has `Byte`
- `short` has `Short`
- `int` has `Integer`
- `long` has `Long`
- `boolean` has `Boolean`
- `char` has `Character`
- `float` has `Float`
- `double` has `Double`

Classes e objetos




Caneta	
-	modelo
-	cor
-	ponta
-	carga
-	Escrever
-	Rabiscar
-	Pintar
-	Tampar
-	Destampar

← Nome da classe

← **Atributos** dessa classe

← **Métodos** dessa classe



```
Classe Caneta
modelo: Caractere
cor: Caractere
ponta: Real
carga: Inteiro
tampada: Logico
Metodo rabiscar()
    Se (tampada) entao
        Escreva("ERRO")
    senao
        Escreva("Rabisco")
    FimSe
FimMetodo
Metodo tampar()
    tampada = verdadeiro
FimMetodo
FimClasse
```

Classes

- Toda classe possui um nome;
- Possuem visibilidade, exemplo: **public**, **private** e **protected**;
- Possuem membros como: Características e Ações;

Criar uma classe no Java

```
public class Teste{  
  
    //ATRIBUTOS OU PROPRIEDADES  
  
    //MÉTODOS  
  
}
```

Atributos

Atributos são as características de um objeto, essas características também são conhecidas como variáveis, utilizando o exemplo dos cães, temos alguns atributos, tais como: cor, peso, altura e nome.

```
public class Cachorro{  
    String nome;  
    float peso;  
    float altura;  
    String cor;  
}
```

Métodos

Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos.

Utilizando o exemplo dos cães, temos alguns exemplos: latir, correr, pular.

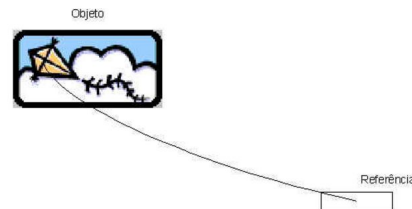
```
public class Cachorro{  
    public String nome;  
    public float peso;  
    public float altura;  
    public String cor;  
  
    void pular {  
    }  
}
```


Objetos em Java

- É importante saber que tipos primitivos em java alocam espaço na memória quando são declarados;
- Objetos NÃO alocam o seu espaço na memória quando declarados, apenas quando criados;
- Variáveis declarados como o tipo do objeto não contem dado, apenas um ponteiro para os dados.

Como assim? Como se cria um objeto em Java?

Como programamos no Java?



Perguntas?



Retirado de
(<https://cdn-icons-png.flaticon.com/512/1268/1268705.png>), em 02/03/2022