

Programação Orientada a Objetos

Aula 01 – Classes, Objetos, ambiente, etc INF31098/INF31030

Prof. Dr. Jonathan Ramos jonathan@unir.br

Departamento Acadêmico de Ciências de Computação – DACC

Núcleo de Tecnologia – NT

19/09/2022

Sumário



1 POO vs. Programação Procedural

2 Configuração do Ambiente: JDK

3 Configuração do Ambiente: Eclipse

POO vs. Programação Procedural/Estruturada



Procedural

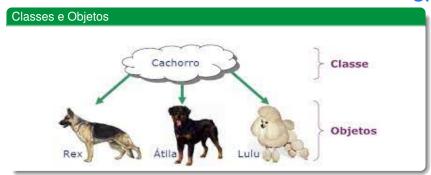
- Trabalha com funções, possui menos reusabilidade de código, menos flexibilidade, menos abstração.
- <https://cplusplus.com/doc/tutorial/variables/>

POO

- Mais abstração, mais reusabilidade de código, mais flexibilidade, representação mais próxima do "Mundo real" (Classes e Objetos)
- <https://www.tutorialspoint.com/java/java_object_classes.htm>



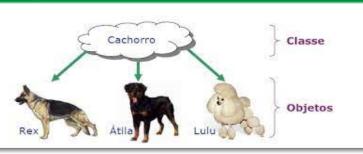




Classes e Objetos



Classes e Objetos



Outros exemplos:

■ Classe: Pessoa:

■ Objetos: Maria, João, Pedro...

■ Classe: Carro:

■ Objetos: Nissan, Fiat, BMW...

Configuração do Ambiente de Programação: JDK



5/11

Javac

Compilar usando **javac**: instalar jdk^a (*Java Development Kit*):

- No terminal (cmd);
- Comando javac para compilar;
- Comando java para rodar o código compilado.

^a<https://www.oracle.com/java/technologies/downloads/>

Rodar "Hello world!' no javac



Criar arquivo com o nome Inicial.java:

Rodar "Hello world! ' no javac



Criar arquivo com o nome Inicial.java:

```
public class Inicial {
    /* Este         meu primeiro programa em Java.
    * 0 programa vai mostrar 'Hello World' como sa da
    */
    public static void main(String []args) {
        System.out.println("Hello World"); // printa Hello World
    }
}
```

- Jave é case-sensitive.
- Todo programa java inicia em public static void main(String args[]): é mandatório.

UNII

Como criar classes

Como criar classes

Exemplo

```
public class Pessoa {
   private String nome;
   private int idade;

public Pessoa() {
    // Codigo do construtor
}

public void apresentarPessoa() {
   // Codigo do metodo
}
}
```



8/11

Depois instanciar o objeto Pessoa e chamar o método apresentar Pessoa():

```
public class Inicial {
   public static void main(String []args) {
     Pessoa p = new Pessoa();
   p.apresentarPessoa();
}
}
```



8/11

Depois instanciar o objeto Pessoa e chamar o método apresentar Pessoa():

```
public class Inicial {
   public static void main(String []args) {
      Pessoa p = new Pessoa();
      p.apresentarPessoa();
   }
}
```

Não apareceu nada, pq?



Depois instanciar o objeto Pessoa e chamar o método apresentarPessoa():

```
public class Inicial {
   public static void main(String []args) {
     Pessoa p = new Pessoa();
   p.apresentarPessoa();
}
}
```

Não apareceu nada, pq?

Precisamos adicionar códigos no método apresentarPessoa().



Depois instanciar o objeto Pessoa e chamar o método apresentarPessoa():

```
public class Inicial {
   public static void main(String []args) {
     Pessoa p = new Pessoa();
   p.apresentarPessoa();
}
```

Não apareceu nada, pq?

Precisamos adicionar códigos no método apresentarPessoa().

```
public void apresentarPessoa() {
    System.out.println("Nome: " + this.nome + ", Idade: " + this.
    idade + " anos.");
}
```

Ainda não apareceu nada....

Os atributos nome e idade estão vazios....



9/11

Testando Objetos e Classes

Ainda não apareceu nada....

Os atributos nome e idade estão vazios....Uma alternativa é criar um construtor que recebe os valores.

```
public Pessoa(String nome, int idade) {
    this.nome = nome;
    this.idade = idade;
}
```

9/11

Testando Objetos e Classes

Ainda não apareceu nada....

Os atributos nome e idade estão vazios....Uma alternativa é criar um construtor que recebe os valores.

```
public Pessoa(String nome, int idade) {
    this.nome = nome;
    this.idade = idade;
}
```

Ao instanciar a Classe Pessoa já podemos passar valores agora...

```
Pessoa p = new Pessoa("Jonathan Ramos", 18);
```

9/11

Testando Objetos e Classes

Ainda não apareceu nada....

Os atributos nome e idade estão vazios....Uma alternativa é criar um construtor que recebe os valores.

```
public Pessoa(String nome, int idade) {
    this.nome = nome;
    this.idade = idade;
}
```

Ao instanciar a Classe Pessoa já podemos passar valores agora...

```
Pessoa p = new Pessoa("Jonathan Ramos", 18);
```

Outras alternativas...

Para alterar o valor dos atributos:

- Podemos definir GETTERS e SETTERS para os atributos privados (Encapsulamento);
- Definir atributos como **public** (não recomendável, há exceções);
- Demonstrações nos códigos:

Configuração do Ambiente de Programação: Eclipse



Eclipse

Compilar usando IDE (Integrated Development Environment) Eclipse^a:

- Exemplos de outros IDEs:
 - NetBeans, BlueJ

^a<https://www.eclipse.org/downloads/>

Configuração do Ambiente de Programação: Eclipse



Eclipse

Compilar usando IDE (*Integrated Development Environment*) **Eclipse**^a:

- Exemplos de outros IDEs:
 - NetBeans, BlueJ

Rodar os mesmos códigos usados no javac no Eclipse. Intro ao javax

^a<https://www.eclipse.org/downloads/>

Classes, Objetos, Ambiente, etc



FIM!

jonathan@unir.br