

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



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

Javac

Compilar usando **javac**: instalar `jdka` (*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**:

```
1 public class Inicial {  
2  
3     /* Este     meu primeiro programa em Java.  
4      * O programa vai mostrar 'Hello World' como sa da  
5      */  
6  
7     public static void main(String []args) {  
8         System.out.println("Hello World"); // printa Hello World  
9     }  
10 }
```

Rodar “Hello world!” no javac

Criar arquivo com o nome **Inicial.java**:

```
1 public class Inicial {  
2  
3     /* Este     meu primeiro programa em Java.  
4      * O programa vai mostrar 'Hello World' como sa da  
5      */  
6  
7     public static void main(String []args) {  
8         System.out.println("Hello World"); // printa Hello World  
9     }  
10 }
```

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

Testando Objetos e Classes

Como criar classes

```
1 <modificador de acesso> class NomeDaClasse {  
2     // Local onde atributos, construtores e metodos sao criados  
3 }
```

Testando Objetos e Classes

Como criar classes

```
1 <modificador de acesso> class NomeDaClasse {  
2     // Local onde atributos, construtores e metodos sao criados  
3 }
```

Exemplo

```
1 public class Pessoa {  
2     private String nome;  
3     private int idade;  
4  
5     public Pessoa() {  
6         //Codigo do construtor  
7     }  
8  
9     public void apresentarPessoa() {  
10        //Codigo do metodo  
11    }  
12 }
```

Testando Objetos e Classes

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

```
1 public class Inicial {  
2     public static void main(String []args) {  
3         Pessoa p = new Pessoa();  
4         p.apresentarPessoa();  
5     }  
6 }
```

Testando Objetos e Classes

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

```
1 public class Inicial {  
2     public static void main(String []args) {  
3         Pessoa p = new Pessoa();  
4         p.apresentarPessoa();  
5     }  
6 }
```

Não apareceu nada, pq?

Testando Objetos e Classes

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

```
1 public class Inicial {  
2     public static void main(String []args) {  
3         Pessoa p = new Pessoa();  
4         p.apresentarPessoa();  
5     }  
6 }
```

Não apareceu nada, pq?

Precisamos adicionar códigos no método **apresentarPessoa()**.

Testando Objetos e Classes

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

```
1 public class Inicial {  
2     public static void main(String []args) {  
3         Pessoa p = new Pessoa();  
4         p.apresentarPessoa();  
5     }  
6 }
```

Não apareceu nada, pq?

Precisamos adicionar códigos no método **apresentarPessoa()**.

```
1 public void apresentarPessoa() {  
2     System.out.println("Nome: " + this.nome + ", Idade: " + this.  
3         idade + " anos.");  
4 }
```

Testando Objetos e Classes

Ainda não apareceu nada....

Os atributos nome e idade estão vazios....



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.

```
1 public Pessoa(String nome, int idade) {  
2     this.nome = nome;  
3     this.idade = idade;  
4 }
```


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.

```
1 public Pessoa(String nome, int idade) {  
2     this.nome = nome;  
3     this.idade = idade;  
4 }
```

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

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

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.

```
1 public Pessoa(String nome, int idade) {  
2     this.nome = nome;  
3     this.idade = idade;  
4 }
```

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

```
1 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

^a[<https://www.eclipse.org/downloads/>](https://www.eclipse.org/downloads/)

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

Classes, Objetos, Ambiente, etc



FIM!

jonathan@unir.br