

Programação Orientada a Objetos

POO29004 – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

<http://docente.ifsc.edu.br/mello/poo>

12 DE FEVEREIRO DE 2020



**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
São José

Objetivo da disciplina

Introduzir os conceitos da programação orientada a objetos, apresentar a linguagem de programação Java e um Ambiente Integrado para Desenvolvimento de software

■ **Carga horária: 36h teóricas + 36h práticas**

- 13:30 – 15:20 - quarta-feira – Lab. Sistemas Digitais
- 15:40 – 17:30 - sexta-feira – Lab. CAD3

■ **Atendimento paralelo:**

- quarta-feira 09:40 – 11:20

■ **Notas de aulas**

- <https://docente.ifsc.edu.br/mello/poo>



■ 3 Projetos práticos (p)

- **Recuperação:** Dedução de 10% do valor total da nota para cada dia de atraso após o prazo de entrega, sendo 5 dias o prazo máximo para entrega tardia.

■ 7 Listas de exercícios (e)

- Dedução de 50% da nota quando entregue depois do prazo, sendo 5 dias o prazo máximo para entrega tardia.

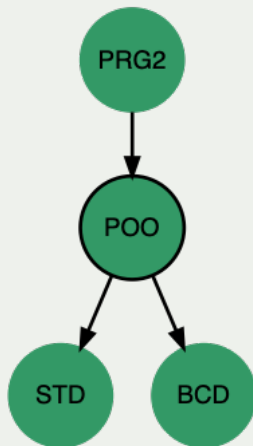
■ Critérios para aprovação na disciplina

- *Conceito final* ≥ 6 ; E
- Mínimo 75% de presença

$$W = \{w_1, w_2, w_3\} = \{3, 4, 2\}$$

$$CF = \left\lfloor \left(\prod_{i=1}^3 p_i^{w_i} \right)^{\frac{1}{\sum_{i=1}^3 w_i}} \times 0,9 + \left(\prod_{i=1}^7 e_i \right)^{\frac{1}{7}} \times 0,1 \right\rfloor, \quad CF \in \mathbb{N}.$$





- 1 Ferramenta de controle de versão
- 2 Conceitos sobre Programação Orientada a Objetos
- 3 Linguagem de programação Java
- 4 Processo de desenvolvimento de software



Linguagem Java

- Em 1991 Sun Microsystems acreditava que a nova onda computacional seria a **união dos dispositivos eletrônicos portáteis com os computadores**
- Foco inicial do projeto em produtos para entretenimento ligados a indústria de **TV digital**





Figura: Produto inicial chamado StarSeven - *7

- Em 1995 Sun lança oficialmente o ambiente Java e sua incorporação no Netscape Navigator trouxe vida as páginas web, antes estáticas

- Aplicações para computadores de mesa – Ex: IRPF
- Aplicações servidoras – Ex: Apache Tomcat
- Aplicações web (Java Server Pages - JSP)
- Dispositivos móveis – Ex: Android
- Sistemas embarcados – Ex: Ginga (SBTVD), SmartTVs, Smartcards



■ Orientada a objetos

- Paradigma que surgiu na década de 60 que tem como foco dados, ou objetos, e suas interfaces
- Recursos de OO do Java são comparáveis aos recursos do C++

■ Robustez

- Ênfase na verificação preliminar de possíveis problemas, verificação dinâmica (em tempo de execução) e eliminação de situações propensas a erros
- Apresenta solução elegante para os principais pontos fracos do C++
 - Alocação dinâmica de memória e ponteiros



■ Neutro em relação à arquitetura

- Compilador gera um código intermediário, chamado de *bytecode*
- *bytecode* é executado pela Máquina virtual Java (JVM)

■ Portável

- Na especificação da linguagem não existem pontos como “dependente de implementação”, como ocorre em C e C++
- Em Java o tipo primitivo `int` consiste de um número inteiro de 32 bits

```
1 #include<stdio.h>
2 #include<limits.h>
3
4 int main(void){
5     // Qual o valor será impresso?
6     printf("%ld\n", LONG_MAX); //Depende da arquitetura: 32 ou 64bits
7     return 0;
8 }
```



■ Independente de plataforma

- Escreva uma única vez e execute em qualquer lugar que tenha uma JVM
- Outras linguagens de programação executadas pela JVM. Ex: Groovy, Scala, Kotlin, Jython, JRuby

■ Desempenho

- Os *bytecode* são interpretados pela JVM resultando em um desempenho inferior quando comparado com códigos compilados para um CPU específico
- Os compiladores de *bytecode* “*just-in-time*” surgem como uma solução para este problema



■ O Java é interpretado, portanto é muito mais lento

- Compiladores *just-in-time* permitem que códigos Java sejam executados com tanta rapidez como códigos C++
- A inicialização da JVM e as interfaces gráficas em Java (GUI) são sim lentas

■ Javascript é uma versão simplificada do Java

- Javascript foi criada pela Netscape para criação de scripts que podem ser usada em páginas Web



■ Java Development Kit – JDK

- Compilador, máquina virtual Java, código fonte, documentação das APIs
- Distribuído gratuitamente pela Oracle para diversos sistemas operacionais e arquiteturas de máquina

■ Última versão LTS: JDK 11



■ Standard Edition – Java SE

- Para o desenvolvimento de aplicações *desktop*

■ Enterprise Edition – Java EE

- Para o desenvolvimento de aplicações empresariais

■ Mobile Edition – Java ME

- Para o desenvolvimento de aplicações para dispositivos móveis



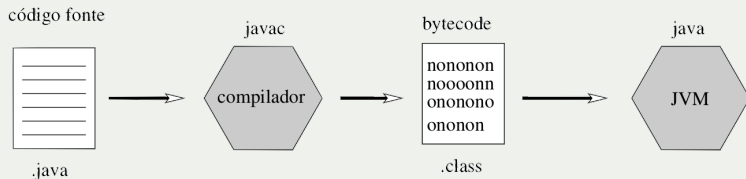
- Os ambientes integrados de desenvolvimento (IDE) tornam mais agradável e produtivo o desenvolvimento de aplicações
 - **Netbeans** – <http://www.netbeans.org>
 - **Eclipse** – <http://www.eclipse.org>
 - **Visual Studio Code** – <https://code.visualstudio.com/>
 - **IntelliJ IDEA** – <https://www.jetbrains.com>
 - Versão *open source* e versão comercial completa (licença gratuita para estudantes e professores)
 - Android Studio é derivado deste



- Um programa em Java consiste em uma coleção de classes
- Geralmente cada classe possui seu respectivo arquivo `.java`
- O nome do arquivo deve ser idêntico ao nome da classe
- O conteúdo do método `main` é a primeira parte de uma classe a ser executada, quando pretende-se que esta seja executada como um aplicativo



Criando e executando um aplicativo Java



■ Compilando

```
9 javac Arquivo.java
```

■ Executando

```
10 java Arquivo
```



Primeiro código em Java – OlaMundo.java

```
11 public class OlaMundo{  
12  
13     public static void main(String[] args){  
14         // imprimindo a mensagem na tela  
15         System.out.println("Ola mundo!");  
16     }  
17 }
```

■ Compilando e executando

```
18 $ javac OlaMundo.java  
19  
20 $ java OlaMundo
```



Referências sobre a linguagem

Tipos primitivos

```
21 byte b = 65;
22 char c = 'A'; // ou c = 65;
23
24 int i = 65;
25 long l = 65L;
26 short s = 65;
27
28 double d = 65.1;
29 float f = 65.1f; // ou f = (float) 65.1;
30
31 boolean b = true; // ou false
32
33 if ((i == 10) && (c == 'a')){
34     while((b == true) || (i >= 0 )){
35         i--;
36     }
37 }
```

[Clique aqui para ver a documentação oficial](#)



Referências sobre a linguagem

```
38 int[] vet = new int[10]; //vetor de inteiros com 10 posições
39 vet[0] = 5;
40 vet[9] = 4;
41 int[][] mat = new int[2][2]; //vetor com 2 dimensões (matriz)
42
43 // algumas funções matemáticas
44 double d = Math.sqrt(25); // obtém a raiz quadrada
45 d = Math.pow(4,2); // 4 elevado a 2
46 d = Math.sin(45); // Math.cos(45), Math.tan(45)...
47
48 // obtendo números pseudo-aleatórios de 0 a 9
49 Random r = new Random();
50 int i = r.nextInt(10);
51
52 // formatando a saída
53 // largura de campo de 8 caracteres e precisão de 2 caracteres
54 System.out.printf("%8.2f", d);
```



Classe String

```
55 String s = "Engenharia";
56 String sub = null; // valor nulo
57
58 if (s.empty()){ // verifica se está vazia
59     System.out.println("Vazia");
60 }else if (s.equals("Tele")){ // para comparar Strings
61     System.out.println("Iguais");
62 }
63 // a partir da posição 0 pegue 4 caracteres
64 sub = s.substring(0,4);
65
66 int tamanho = sub.length(); //obtendo o tamanho
67 char c = sub.charAt(1); // obtendo caractere na posição 1
68
69 String alunos = "Joao:Pedro:Ana";
70
71 // criará vetor de Strings com 3 elementos
72 String[] vetAlunos = alunos.split(":");
73
74 System.out.println(vetAlunos[0]); // Joao
```



Lendo informações pelo teclado

```
75 import java.util.Scanner;
76
77 public class Segundo{
78
79     public static void main(String[] args){
80
81         Scanner teclado = new Scanner(System.in);
82
83         System.out.print("Entre com seu nome: ");
84         String s = teclado.nextLine(); // lendo cadeia de caracteres
85         System.out.println("Nome: " + s);
86         System.out.print("Informe um número inteiro: ");
87         int i = teclado.nextInt(); // lendo inteiro
88         System.out.print("Informe um número real: ");
89         double r = teclado.nextDouble(); // lendo real
90         System.out.println("inteiro: " + i + ", real: " + r);
91     }
92 }
```



- Se o método `nextLine()` for chamado depois dos métodos `nextInt()`, `nextDouble()`, `nextFloat()`, `nextByte()`, `nextShort()`, `nextLong()` ou `next()`, então ele não irá ler os valores
- Os métodos `nextXXXX()` ignoram o caractere de nova linha (NL) e assim esse caractere é consumido pelo `nextLine()` subsequente
- **Solução:** Adicionar um chamada extra do método `nextLine()`

```
93 int i = teclado.nextInt();  
94 teclado.nextLine(); // chamada extra para consumir NL  
95 String s = teclado.nextLine(); // lendo cadeia de caracteres
```



Argumentos de linha de comando

```
96 public class Argumentos{
97
98     public static void main(String[] args){
99
100         System.out.println("Forneceu " + args.length() + "argumentos");
101         System.out.println("Argumentos fornecidos: ");
102
103         for(String argumento: args){
104             System.out.println(argumento);
105         }
106         // o laço abaixo é equivalente ao laço acima feito com foreach
107         for(int i = 0; i < args.length(); i++){
108             System.out.println(args[i]);
109         }
110     }
111 }
```

```
112 javac Argumentos.java
113 java Argumentos Ola mundo
```



Usando interface gráfica para interagir com o usuário

```
114 import javax.swing.JOptionPane;
115
116 public class Terceiro{
117
118     public static void main(String[] args){
119         String s = JOptionPane.showInputDialog("Entre com um numero");
120
121         //convertendo String para int
122         int numero = Integer.parseInt(s);
123
124         JOptionPane.showMessageDialog(null, numero);
125     }
126 }
```



Exercício

- Faça um programa que leia do teclado dois números inteiros e um operador (+, -, \ ou *) e apresente o resultado da operação aritmética.

```
127 Entre com o 1o. operando: 10
128 Entre com o 2o. operando: 20
129 Entre com o operador: +
130
131 Resultado: 10 + 20 = 30
```





CAELUM

APOSTILA CAELUM FJ-11 JAVA E ORIENTAÇÃO A OBJETOS

<http://docente.ifsc.edu.br/mello/livros/java/apostila-caelum-java-orientacao-objetos-FJ11.pdf>

- Ler capítulos 2 e 3



SUN MICROSYSTEMS

JAVA CODE CONVENTIONS

<http://docente.ifsc.edu.br/mello/livros/java/java-codeconventions.pdf>

