

# Programação Orientada a Objetos

*Professor Filipe Dwan Pereira*

## Aula 1 – Apresentação da Disciplina

“Ensina-nos a contar os nossos dias, de tal maneira que alcancemos corações sábios.” *Salmos 90:12*

# Identificação da Disciplina

**Disciplina:** Programação Orientada a Objetos

**Código:** DCC305

**Turma:** A

**Ano:** 2016    **Semestre:** 2    **Carga horária:**  
60h

**Aulas Teóricas e Práticas:** 2a e 4a, das 10:00  
às 12:00.

**Professor:** Filipe Dwan Pereira

**Email:** [filipe.dwan@ufrr.br](mailto:filipe.dwan@ufrr.br)

**Ambiente Virtual de Aprendizagem:**

<http://ava.ufrr.br/dcc/>

# Apresentação do Plano de Ensino

Programação Orientada a Objetos

# Atendimento ao aluno

- Horário de atendimento do professor:
  - Sexta-feira - 14:00 às 18:00
  - Local: LabGeogeb.

# Objetivo desta Aula

- Responder o que é Java;
- Mostrar as vantagens e desvantagens do Java;
- Entender bem o conceito de máquina virtual;
- Compilar e executar um programa simples.

# Disclaimer

- Este slide foi baseado nas seguintes fontes principais:
  - SOFTBLUE. Professor Carlos Eduardo Gusso Tosin. Fundamentos de Java. <http://www.softblue.com.br/>.
  - Slides professor Horácio Oliveira – UFAM.
  - CAELUM. Java e Orientação a Objetos. Disponível em: <https://www.caelum.com.br/apostila-java-orientacao-objetos/>
  - K19. Java e Orientação a Objetos. Disponível em: <http://www.k19.com.br/cursos/orientacao-a-objetos-em-java>.

# Quais eram os seus maiores problemas quando programava na década de 1990?

- ponteiros?
- gerenciamento de memória?
- organização?
- falta de bibliotecas?
- ter de reescrever parte do código ao mudar de sistema operacional?
- custo financeiro de usar a tecnologia?

# Como tudo começou

- Busca de inovações tecnológicas pela Sun (1992)
- Time liderado por James Gosling
- A grande motivação para criação do Java era um interpretador para trabalhar com diversos aparelhos eletrônicos, como tvs, video-cassetes, aspiradores, liquidificadores e outros.
- A ideia não funcionou.



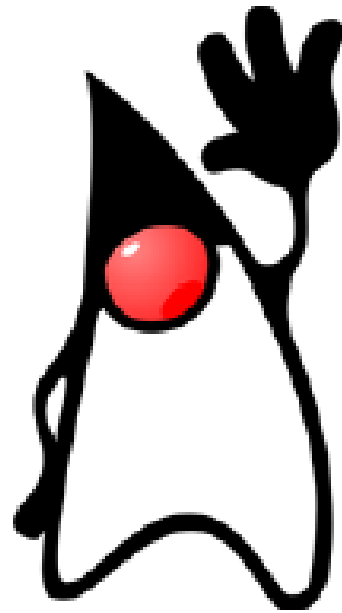
# O Java e a internet

- Com o aparecimento da Internet, a Sun resolveu usar sua ideia para rodar aplicações dentro dos browsers
- Grande quantidade de sistemas operacionais e browsers
- Na época os browsers só renderizavam HTML
- Criação das applets

# No final das contas

- Java nasceu com um objetivo
- Foi lançado com outro objetivo
- Firmou-se como tecnologia de desenvolvimento de aplicações no lado servidor

# Logotipo e Mascote



# O que é o java?

- Uma plataforma integral, com uma biblioteca enorme contendo uma grande quantidade de códigos reutilizáveis e um ambiente de execução que fornece serviços como:
  - Segurança;
  - Portabilidade para diferentes SO's;
  - Coleta de lixo automático.
- Linguagem com uma vasta documentação.

# O que é o java?

vídeo

- Os autores do Java escrevem um artigo influente que explica seus objetivos e destacam características chave, sobre a linguagem:

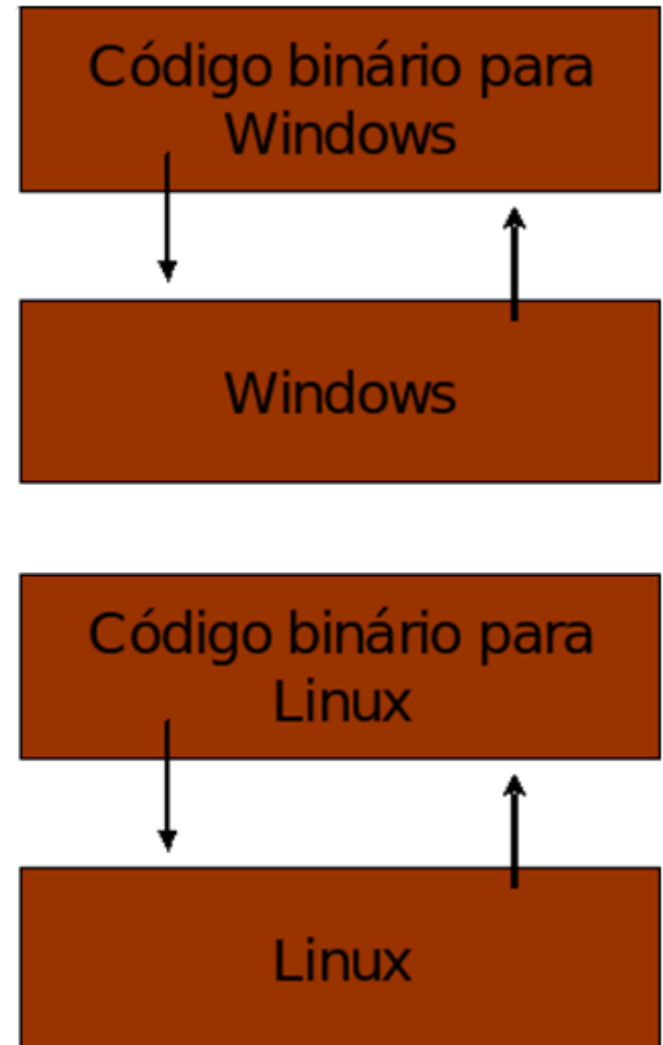
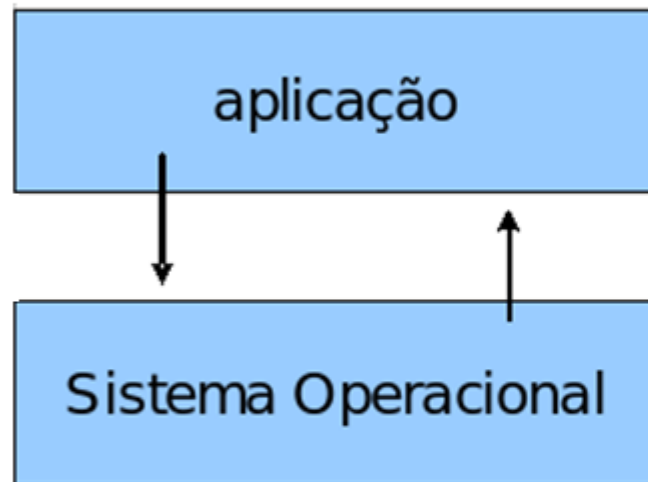
- Simples;
- **Orientada a Objetos;**
- Compatibilidade com redes;
- Robusto;
- Seguro;
- Arquitetura neutra;

- Portável
- Interpretado;
- Alto desempenho
- Múltiplos threads
- Dinâmico;

# Java

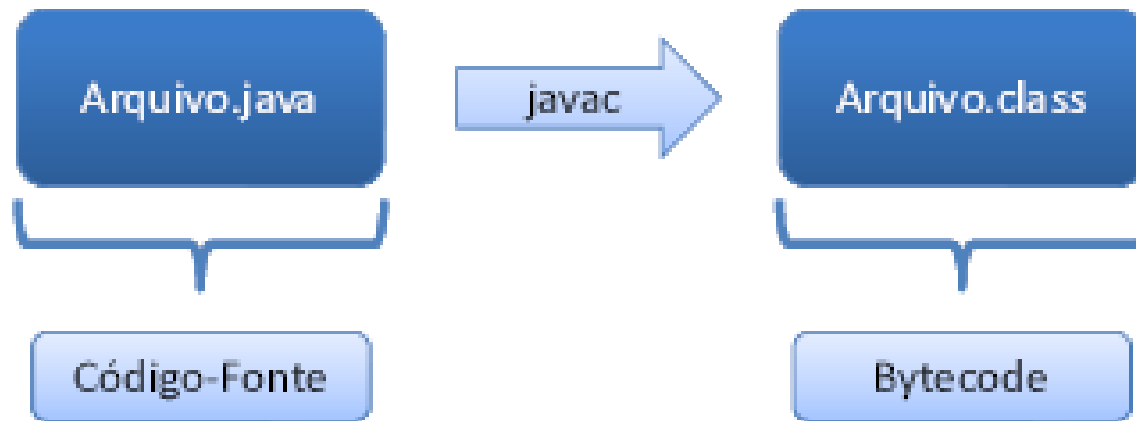
- No Brasil, diversos grupos de usuários se formaram para tentar disseminar o conhecimento da linguagem. Um deles é o GUJ (<http://www.guj.com.br>), uma **comunidade virtual com artigos, tutoriais e fórum** para tirar dúvidas, o maior em língua portuguesa.

# Compilação



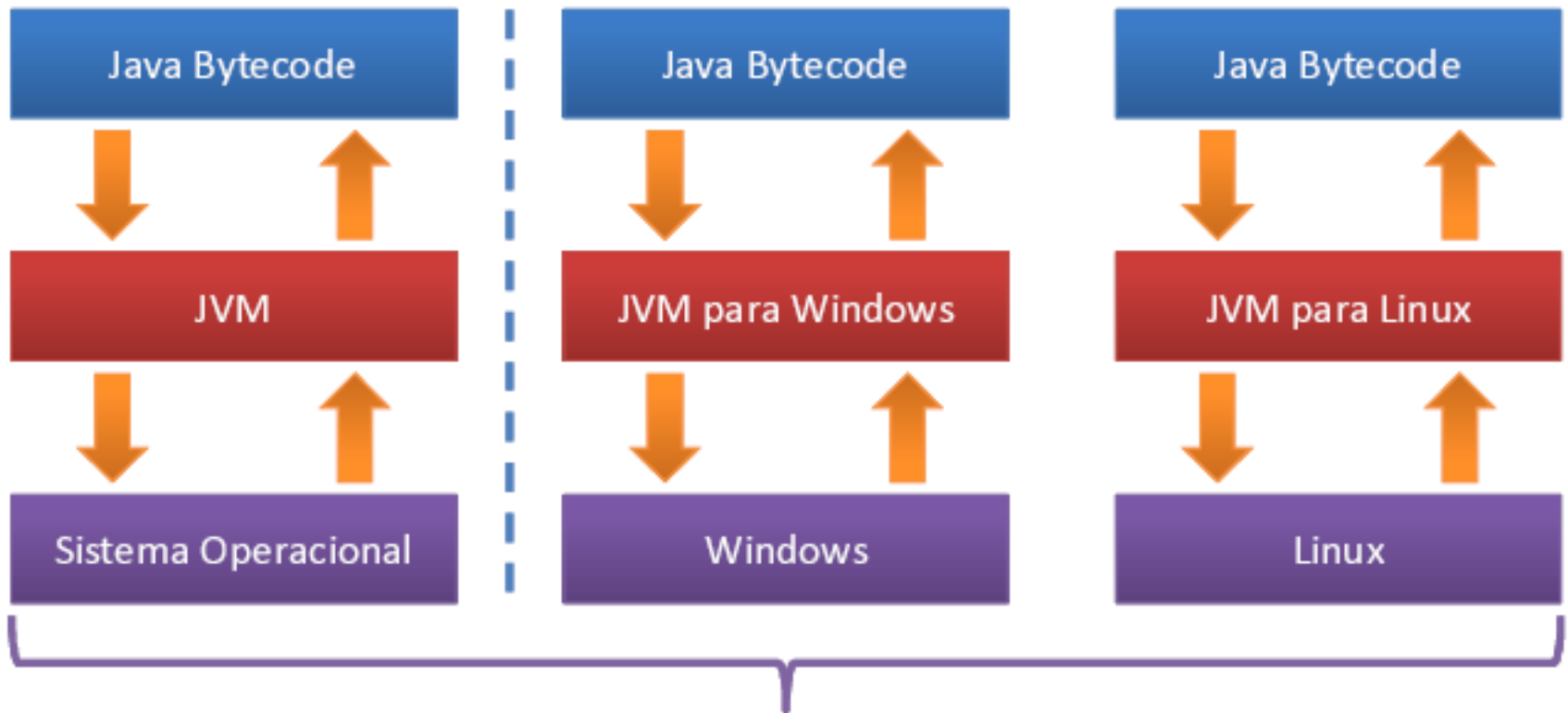
# Bytecode

- O bytecode é uma linguagem entendida pela JVM
- A geração do bytecode é feita através da compilação do código Java





# Máquina Virtual



Ao mudar a plataforma, o bytecode não precisa ser alterado

# Máquina Virtual Java (JVM)

- Independência de Plataforma (Portabilidade)
  - A aplicação roda sem nenhum envolvimento com o SO. Sempre conversando apenas com a JVM
- Garbage collection
- O JVM não entende código Java, mas sim um código de máquina específico. Esse código de máquina é gerado por um compilador Java, como o javac, e é conhecido por “bytecode”.

Princípio **WORA**

Write once, run anywhere

# A JVM é uma Especificação

- Diversas empresas implementam a JVM
  - Oracle, IBM, etc.
- Como a JVM é uma especificação que deve ser seguida por quem a implementa, é possível trocar de JVM sem a necessidade de recompilar os códigos das aplicações

# Java lento? Hotspot e JIT

- Hotspot é a tecnologia que a JVM utiliza para detectar pontos quentes da sua aplicação: código que é executado muito, provavelmente dentro de um ou mais loops.
- Quando a JVM julgar necessário, ela vai compilar estes códigos para instruções realmente nativas da plataforma, tendo em vista que isso vai provavelmente melhorar a performance da sua aplicação. Esse compilador é o JIT: Just inTime Compiler, o compilador que aparece “bem na hora” que você precisa.

# Evolução das versões do Java

- Java 1.0 e 1.1
  - Primeiras versões
- Java 2 (Java 1.2)
  - Aumento significativo no tamanho da API
- Java 2 (Java 1.3 e 1.4)
  - Melhorias na API
  - Java 1.3.1, 1.4.1, 1.4.2, etc.
- Java 5 (Java 1.5)
  - Diversas mudanças significativas

# Evolução das versões do Java

- Java 6 (Java 1.6)
  - Mais recursos na API
  - Muitas melhorias de performance da JVM
- Java 7
  - Novas APIs
  - Novos recursos na linguagem
  - Melhorias internas na JVM
- Java 8
  - Nova API de data e hora
  - Suporte a expressões lambda

Até o momento, o java é a 3ª linguagem mais utilizada do mundo, segundo o rank da Tiobe

<http://www.tiobe.com/tiobe-index/>

Sep 2022	Sep 2021	Change	Programming Language		Ratings	Change
1	2	▲		Python	15.74%	+4.07%
2	1	▼		C	13.96%	+2.13%
3	3			Java	11.72%	+0.60%
4	4			C++	9.76%	+2.63%
5	5			C#	4.88%	-0.89%
6	6			Visual Basic	4.39%	-0.22%

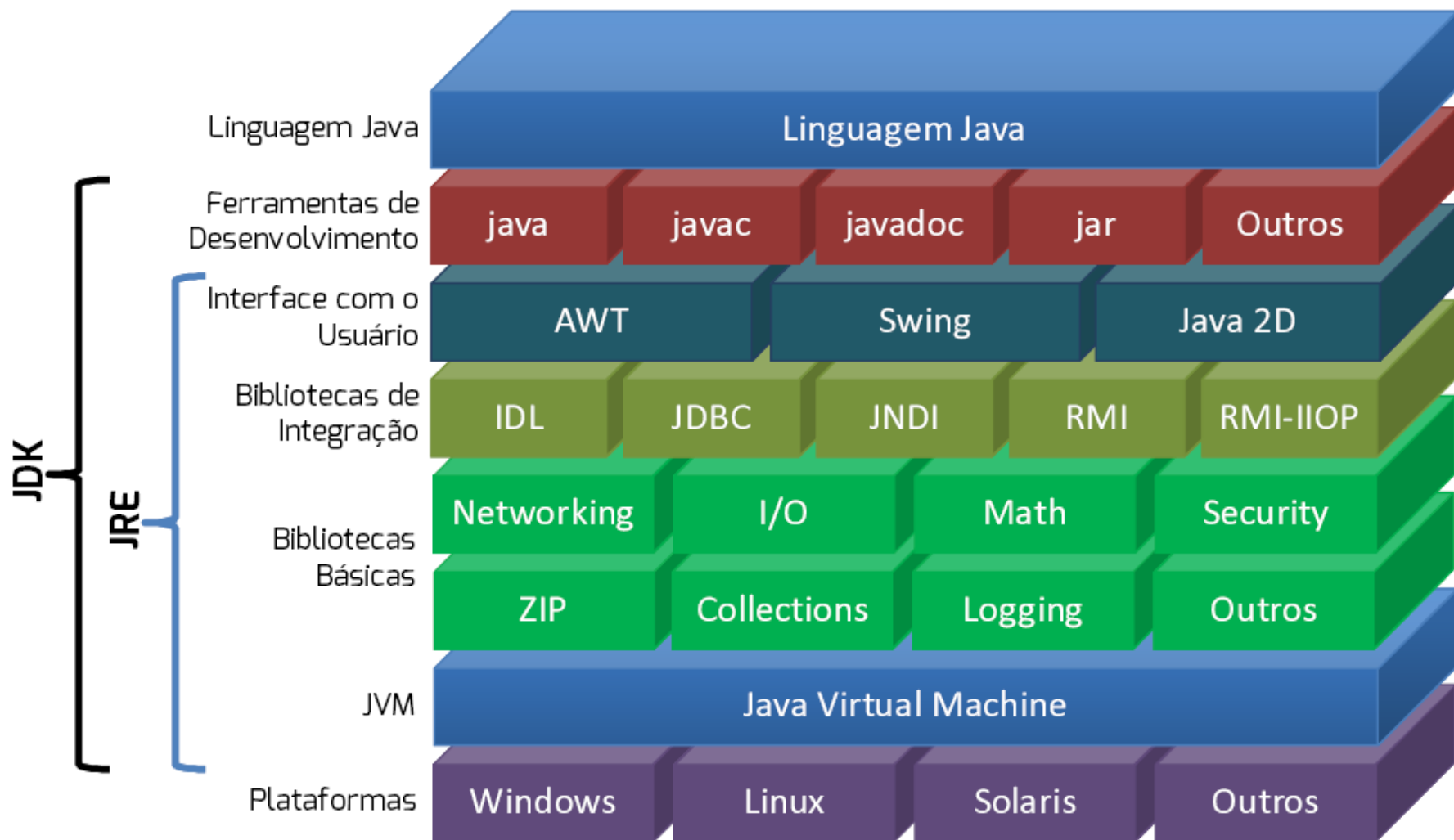
# JVM? JRE? JDK? O que devo baixar?

- JVM = apenas a virtual machine, esse download não existe, ela sempre vem acompanhada.
- JRE = Java Runtime Environment, ambiente de execução Java, formado pela JVM e bibliotecas, tudo que você precisa para executar uma aplicação Java. Mas nós precisamos de mais.
- JDK = Java Development Kit: Nós, desenvolvedores, faremos o download do JDK do Java SE (Standard Edition). Ele é formado pela JRE somado a ferramentas, como o compilador.

**Tanto o JRE e o JDK podem ser baixados do site <http://java.sun.com>, hoje gerenciado pela Oracle.**



# Elementos da JRE e JDK



# Onde usar o Java

- O foco da plataforma é projetos de médio a grande porte.
- Aplicações que virão a se expandir no futuro
- Times de desenvolvimento
- Para pequenas aplicações com poucos desenvolvedores, outras linguagens podem ser mais produtivas

# Compilando o Primeiro Código

- Vamos para o nosso primeiro código! O programa que imprime uma linha simples.
- Para mostrar uma linha, podemos fazer:  
`System.out.println("Minha primeira aplicação Java!");`

# O mínimo que precisaríamos escrever

```
1 class MeuPrograma {  
2     public static void main(String[] args) {  
3  
4         // miolo do programa começa aqui!  
5         System.out.println("Minha primeira aplicação Java!!");  
6         // fim do miolo do programa  
7  
8     }  
9 }
```

## Para saber mais: Como é o Bytecode?

```
javap -c MeuPrograma
```

E a saída:

```
MeuPrograma();
```

Code :

```
0:  aload_0
```

```
1: invokespecial #1; //Method java/lang/Object."<init>":()V
```

```
4: return
```

```
public static void main(java.lang.String[]);
```

Code:

```
0:    getstatic    #2; //Field java/lang/System.out:Ljava/io/PrintStream;
```

```
3: ldc    #3; //String Minha primeira aplicação Java!!
```

```
5:    invokevirtual    #4; //Method java/io/PrintStream.println:
                                (Ljava/lang/String;)V
```

```
8: return
```

# Possibilidade de erros

- Erros de Compilação : Erros de digitação e de uso da sintaxe da linguagem.
- Erros de Link-Edição : Erro no uso de bibliotecas de subprogramas necessárias ao programa principal.
- Erros de Execução : Erro na lógica do programa (algoritmo).

# O que pode dar Errado?

Código:

```
1 class X {  
2     public static void main (String[] args) {  
3         System.out.println("Falta ponto e vírgula")  
4     }  
5 }
```

Erro:

```
X.java:4: ';' expected  
        }  
        ^
```

1 error

# O que pode dar Errado?

Exception in thread "main" java.lang.NoSuchMethodError: main

```
class X {  
    public void main (String[] args) {  
        System.out.println("Faltou o static, tente executar!");  
    }  
}
```

Main method not public.

```
class X {  
    static void main (String[] args) {  
        System.out.println("Faltou o public");  
    }  
}
```



# Comando para ler uma variável:

```
1  package testarhabilidades;
2
3  import java.util.Scanner;
4
5  public class Main {
6
7      public static void main(String[] args) {
8          Scanner input = new Scanner(System.in);
9          int n1,n2, soma;
10         System.out.println("Entre com o primeiro valor: ");
11         n1 = input.nextInt();
12         System.out.println("Entre com o segundo valor: ");
13         n2 = input.nextInt();
14         soma = n1 + n2;
15         System.out.printf("A soma dos números é: "+soma);
16     }
17
18 }
```

# Exercício em Sala

1. Crie um aplicativo em java para calcular a média aritmética de duas notas.
2. Crie um aplicativo em java para calcular a área de uma circunferência.
3. Crie um aplicativo em java para calcular o Índice de Massa Corpórea (IMC) de um indivíduo.  **$IMC = \text{Peso} / (\text{Altura})^2$**

# Links interessantes:

- Tutorial de introdução à linguagem:
  - <http://docs.oracle.com/javase/tutorial/getStarted/index.html>
- Link do Javadoc oficial do Java 8. Ele detalha todas as classes, interfaces, métodos, etc. da API do Java:
  - <http://docs.oracle.com/javase/8/docs/api/>
- Preparação do ambiente para desenvolvimento em Java.
  - <http://www.devmedia.com.br/preparacao-do-ambiente-para-desenvolvimento-em-java/25188>
- Download Eclipse IDE
  - <https://eclipse.org/downloads/>
- Download NetBeans IDE
  - <https://netbeans.org/downloads/>

# Referências Bibliográficas

- DEITEL, Harvey M. e DEITEL, Paul J. Java - Como Programar, 8ª edição. Pearson. 2010.
- BLOCH, Joshua. Effective Java, 2ª edição. Addison-Wesley, 2008.
- CAELUM. Java e Orientação a Objetos. Disponível em: <https://www.caelum.com.br/apostila-java-orientacao-objetos/>
- SOFTBLUE. Professor Carlos Eduardo Gusso Tosin. Fundamentos de Java. <http://www.softblue.com.br/>.
- K19. Java e Orientação a Objetos. Disponível em: <http://www.k19.com.br/cursos/orientacao-a-objetos-em-java>.
- HORSTMANN, CORNELL. Core Java Volume I – Fundamentos, 8ª Edição. São Paulo, Pearson Education, 2010.
- BRAUDE, E. J. Projeto de software - da programação à arquitetura: uma abordagem baseada em Java. Porto Alegre: Bookman, 2005.
- SANTOS, R. Introdução à Programação Orientada a Objetos usando Java. São Paulo: Campus, 2003.
- Slides do Professor Doutor Horácio Fernandes da UFAM.

"Péssima ideia, a de que não se pode mudar". Montaigne



filipedwan@gmail.com

 filipedwan

 @filipedwan