



Fabiano Moreira  
[fabiano.pinho@rj.senac.br](mailto:fabiano.pinho@rj.senac.br)

# Vantagens



- Sintaxe amigável
- Recursos orientados a objetos
- Gerenciamento de memória
- Portabilidade

# O Ambiente de Desenvolvimento

## Os sabores do Java



# O Ambiente de Desenvolvimento

## Pré requisitos



### **SDK (Software Development Kit)** **Java Standard Edition**

**O SDK inclui tudo que você precisa  
para compilar e executar o Java**

**O SDK não inclui a documentação da API**

**Um editor de texto, como o bloco de notas**

# O Ambiente de Desenvolvimento

## Variáveis de ambiente

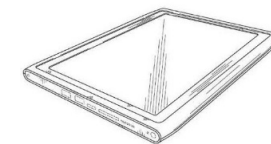
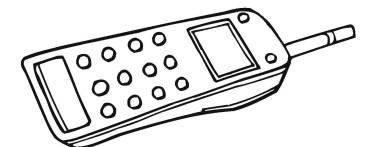
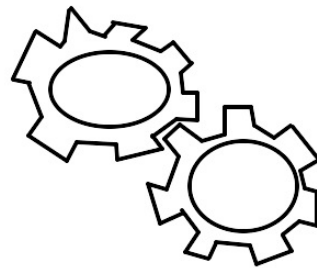


- Configurando o path: criar a variável JAVA\_HOME e atribuir a ela o caminho onde o Java foi instalado.
- Configurar a variável PATH. Incluir a seguinte diretiva: ;%JAVA\_HOME%\bin
- Adicionar a variável CLASSPATH: ;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\htmlconverter.jar;%JAVA\_HOME%\jre\lib;%JAVA\_HOME%\jre\lib\rt.jar;

# Como o Java funciona



```
End Sub  
Private Sub tbToolBar_ButtonClick  
On Error Resume Next  
timTimer.Enabled = True  
Select Case Button.Key  
Case "Back"  
    brwWebBrowser.GoBack  
Case "Forward"  
    brwWebBrowser.GoForward  
Case "Refresh"  
    brwWebBrowser.Refresh  
Case "Home"  
    brwWebBrowser.Home  
End Select  
End Sub
```



Código-fonte

Compilador

Saída (código)

Maquinas virtuais

1

2

3

4

# A Máquina Virtual Java e o Compilador



A máquina virtual é quem efetivamente faz um programa ser executado.

Há uma boa razão para o Java ter sido projetado para usar um compilador de bytecode. Se ele fosse uma linguagem puramente interpretada, onde -no tempo de execução- a máquina virtual tivesse de converter código-fonte diretamente de um editor de texto, o programa Java seria executado muito lentamente.

Java é uma linguagem fortemente tipificada, o que significa que o compilador não pode permitir que as variáveis armazenem dados com o tipo errado. Esse é um recurso de segurança crucial. O compilador pode bloquear a maioria das violações antes que elas cheguem a máquina virtual.

# Sintaxe



- Cada instrução deve terminar com um ponto e virgula. Ex.: **x = x + 1;**
- Um comentário de linha única começa com duas barras. Ex.: **// esta linha é um comentário**
- A maioria dos espaços em branco não é importante.
- As variáveis são declaradas com um nome e um tipo. Ex.: **Int altura;**
- As classes e métodos devem ser definidos dentro de um par de chaves. Ex.:

```
public void ler() {  
    // o código entra aqui  
}
```



# Operador de pré e pós-incremento/decremento



O atalho para se adicionar ou subtrair 1 unidade de uma variável.

**x++;**

é o mesmo que:

**x = x + 1;**

**x--;**

é o mesmo que:

**x = x - 1;**

# Anatomia de uma Classe



- Em Java, tudo é inserido em uma classe. O código fonte deve ser criado com extensão .java
- A classe representa uma parte do programa, embora um aplicativo muito pequeno possa precisar de apenas uma classe
- Uma classe tem um ou mais métodos. Os métodos devem ser declarados dentro de uma classe.
- Após a compilação, o arquivo será convertido em um novo arquivo de classe, com extensão .class

# Anatomia de uma Classe



Quando a JVM começar a ser executada, procurará a classe que você forneceu na linha de comando. Em seguida, começará a procurar um método especialmente escrito que seja exatamente como este:

```
public static void main (String[] args){  
    // seu código aqui  
}
```

Depois a JVM executará tudo que estiver entre as chaves {} do método principal. Todo aplicativo Java precisa ter pelo menos uma classe e um método main (não um método main por classe, apenas um por aplicativo).

# Anatomia de uma Classe



```
public class MinhaClasse {  
    public static void main (String args[]) {  
        System.out.print("Hello World");  
    }  
}
```

# Loops



- O Java tem três estruturas de loop padrão: while, do-while e for. Por enquanto, veremos apenas o while.
- A idéia do loop é de que enquanto alguma condição seja verdadeira, será executado algo dentro do bloco de loop. O bloco de loop é delimitado por um par de chaves; portanto, o que deve ser repetido terá que estar dentro desse bloco.
- A principal parte de um loop é o teste condicional. Em Java, o teste condicional é uma expressão que resulta em um valor booleano. Em outras palavras é algo que é verdadeiro ou falso.

# Loops

## Testes booleanos simples



Operadores de comparação:    < (menor que)  
   > (maior que)  
   == (igualdade)

Observe a diferença entre o operador de atribuição (apenas um sinal de igualdade) e o operador igual a (dois sinais de igualdade).

`int x = 4;`

**(atribui 4 a x)**

`z == 17`

**(compara se o conteúdo da variável z é igual a 17)**

# Loop - while



```
while (<<condição>>){  
    // código a repetir  
}
```

# Exercícios



Cada um dos arquivos Java abaixo representa um arquivo-fonte completo. Sua tarefa é personificar o compilador e determinar se cada um deles pode ser compilado. Se não puderem ser compilados, como você os corrigiria?

**A**

```
class Exercicio1{
    public static void main(String [] args){
        int x = 1;
        while (x < 10){
            System.out.println("número:" + x);
        }
    }
}
```

**B**

```
public static void main(String [] args){
    int x = 5;
    while (x > 1){
        x = x - 1;
        System.out.println("número:" + x);
    }
}
```

**C**

```
class Exercicio1{
    int x = 5;
    while (x > 1){
        x = x - 1;
        System.out.println("número: "+ x);
    }
}
```



# Exercícios



Faça um programa que imprima de 0 a 10.

Faça um programa que imprima de 1 a 500.

Faça um programa que imprima de 1 a 500 em ordem decrescente.

Faça um programa que imprima de 1 a 500, mas apenas os números pares.