

Aula 01

Fundamentos da linguagem JAVA e Introdução à Orientação a Objetos

Pauta:

- História do Java
- Características da Linguagem
- Máquina Virtual Java – JVM
- Ambiente de Trabalho
- Primeiro Programa

Pauta (continuação):

- Compilação e Execução de classes
- String
- Aritmética
- Declaração de variáveis

História do Java

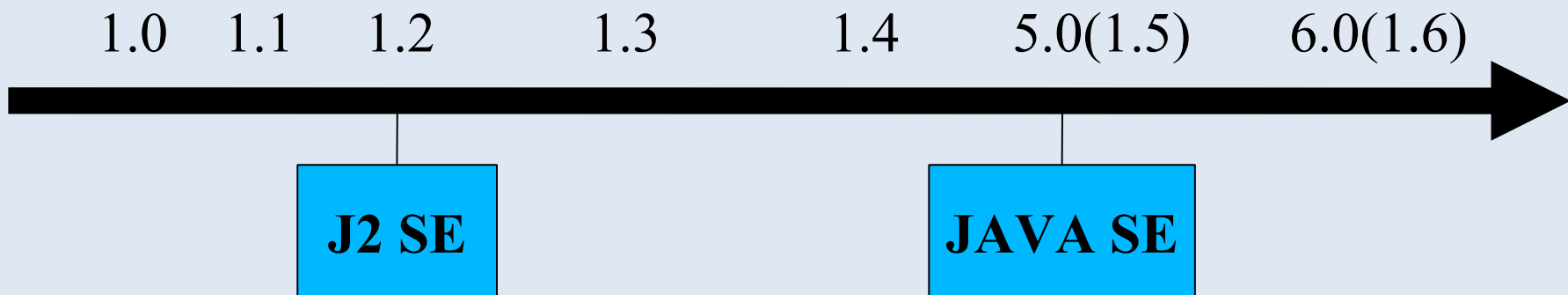
- Projeto Green (1991)
- Web ficou popular em 1993
- Sun anunciou o Java 1995
- Inicialmente o Java despertou interesse para desenvolvimento WEB
- Java Community Process

Características da linguagem

- Orientado a objetos;
- Multiplataforma;
- Roda em uma máquina virtual;
- É mantido pelo JCP;
- Internet:
 - <http://java.sun.com>
 - <http://www.jcp.org>

Características da linguagem

- Java SE
- Java EE
- Java ME



Máquina Virtual Java - JVM

- Carregar e executar o bytecode;
- Coletar o lixo;
- Segurança;
- Uma interface para o código nativo;
- JIT (Just in Time)

Máquina Virtual Java - JVM

Código JAVA:

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

Compilador

Byte code:

```
111001001000101  
001001010100011  
110100011101011
```

JVM

Ambiente de Trabalho

- Jbuilder 2007 (Eclipse)
 - www.codegear.com
 - www.eclipse.org
- JDK 5.0 (Pode-se utilizar a versão 6.0)
 - www.java.sun.com
- Criar área de trabalho na máquina do aluno

Primeiro programa:

- Pacote java.lang
- `public class NomeDaClasse`
- Método main
- Case sensitive (maiúsculas diferentes de minúsculas)
- Nome da classe é o mesmo nome do arquivo
- Após cada instrução deve existir um ponto vírgula (;)

Primeiro programa:

- Comentários:

// comentário de linha

/* comentário de bloco */

/** comentário de Java DOC */

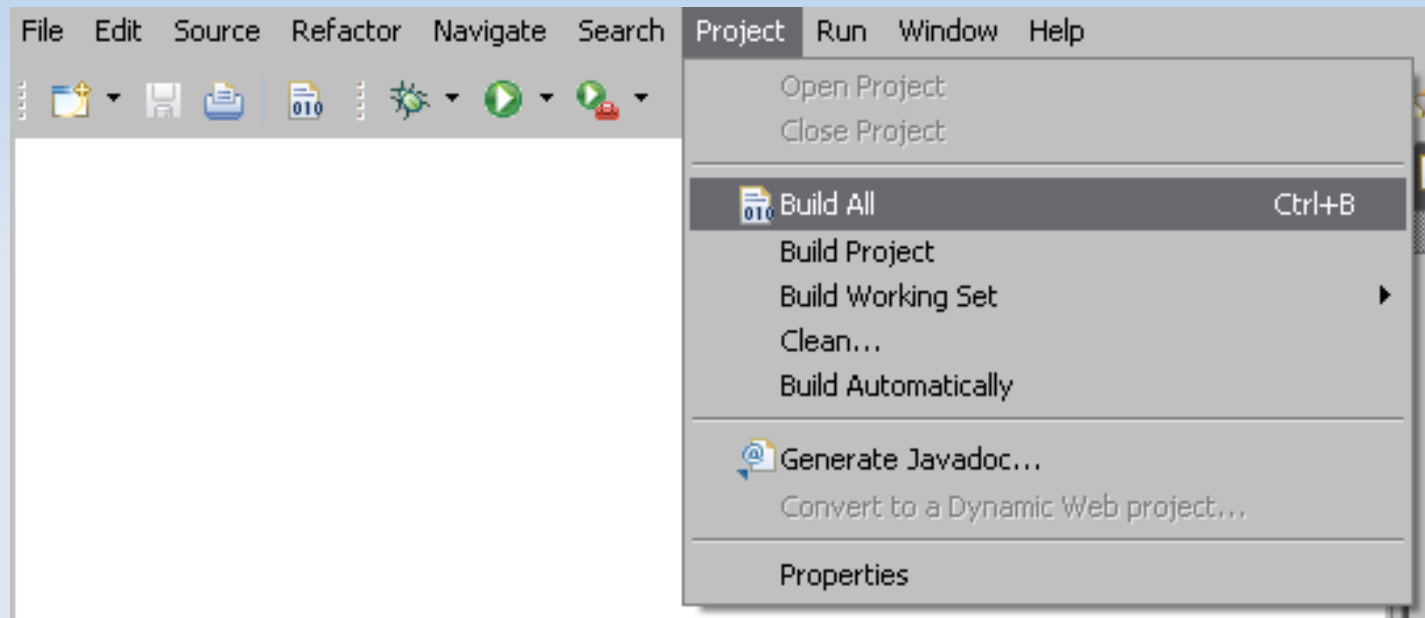
Primeiro programa:

```
public class Welcome1 {  
  
    public static void main( String args[] ) {  
  
        System.out.println( "Welcome to Java Programming!" );  
  
    }  
  
}
```

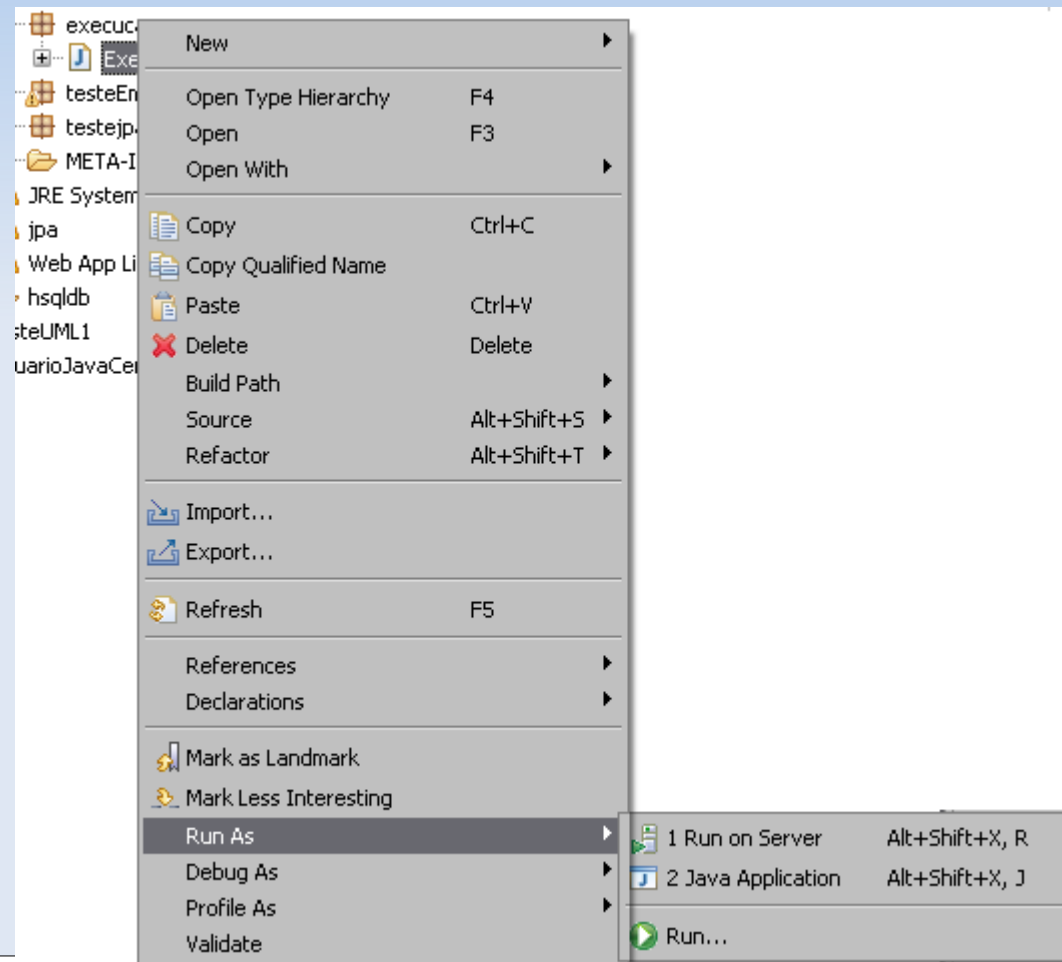
Compilação e Execução de classes

- Compilação:
 - `javac Welcome1.java`
- Execução:
 - `java Welcome1`

Compilação e Execução de classes



Compilação e Execução de classes



String

- Sequência de caracteres
- Suporta as sequências de escape:
 - \n Nova Linha
 - \t Tabulação
 - \r Retorno de carro
 - \" Aspas duplas
 - \\ Barra Invertida

String

```
• public class Welcome3 {  
•     public static void main( String args[] ) {  
•         System.out.println( "Welcome\nto\nJava\nProgramming!" );  
•     }  
• }
```

String

- Saída formatada incluída no Java 5
- Exemplo:

```
System.out.printf("%s\n%s", "Welcome to", "Java Programming!")  
    ;
```

String

```
• public class Welcome4 {  
•     public static void main( String args[] ) {  
•         System.out.printf( "%s\n%s\n",  
•             "Welcome to", "Java Programming!" );  
•     }  
• }
```

String

- Podemos utilizar o operador + para concatenar duas strings. Ex.:

“Primeira String” + “Segunda String”

resulta em:

“Primeira StringSegunda String”

Aritmética

- Tipo primitivo int
- Maior prioridade:

Multiplicação *

Divisão /

Resto %

- Menor prioridade

Adição: +

Subtração -

Declaração de variáveis

- Exemplos:

`int numero;`

`int numero = 0;`

`int numero, resposta;`

`int numero = 0, resposta = 4;`

Introdução a classes e objetos

Pauta

- A diferença entre classe e objeto
- Declaração de classes
- Declaração de métodos
- Métodos que recebem parâmetros
- Métodos que retornam valores
- Executando métodos de objetos
- Variáveis de instância
- Métodos Construtores

A diferença entre classe e objeto

- Qual a diferença entre classe e objeto?

A diferença entre classe e objeto

- Qual a diferença entre classe e objeto?
- Qual a diferença entre o projeto de um carro e o carro construído?

A diferença entre classe e objeto

- Classe é o código escrito – projeto do carro
- Objeto é a instância da classe – o carro construído

Declaração de classes

```
public class NomeDaClasse{  
}
```

- Em java os nomes das classes iniciam com letra maiúsculas.

Declaração de Métodos

```
public void nomeDoMetodo()  
{  
}
```

- Em java os nomes dos métodos iniciam com letra minúsculas com o restante das palavras iniciando com letras maiúsculas

Método com parâmetros

```
public void nomeDoMetodo(tipo parametro){  
  
}
```

Método com múltiplos parâmetros

```
public void nomeDoMetodo(tipo p1, tipo p2){  
  
}
```

Método com retorno

```
public int nomeDoMetodo(){  
    return int;  
}
```


Instanciando um objeto

- Em Java utilizamos a palavra chave new para criar uma nova instância de uma classe:

Classe objeto = new Classe();

Classe objeto;

objeto = new Classe();

Invocando um método do objeto

- Para invocarmos um método de algum objeto utilizamos o operador . (ponto)
- objeto.executaMetodo();

Invocando um método do objeto

```
public class GradeBook{  
    public void displayMessage(){  
        System.out.println( "Welcome to the Grade Book!" );  
    }  
}
```

Um exemplo

```
public class GradeBookTest{  
    public static void main( String args[] ){  
        GradeBook myGradeBook = new GradeBook();  
        myGradeBook.displayMessage();  
    }  
}
```

Variáveis de Instância

- Variáveis declaradas no escopo da Classe (fora do corpo dos métodos)
- Podem ser acessadas a partir das instâncias da classe
 - Obs.: este conceito será melhor definido mais adiante

Métodos construtores

- Não retornam valores;
- Possuem o mesmo nome da classe;
- Podem receber valores;
- O compilador fornece um construtor padrão quando este não é declarado;

Duvidas??

Exercícios:

- Qual o propósito de se utilizar: //
- Qual o propósito de se utilizar: \\
- Qual o valor de x após a execução do código abaixo:

$X = 5 + 3 * 2 + 4 / 2;$

Exercícios

- Qual o propósito da palavra chave new?
- O que é um construtor padrão?
- Qual a diferença entre classe e objeto?

Exercícios

- Crie uma classe chamada Empregado com os seguintes campos:
 - nome e salário
- E o seguinte método:
 - imprimirDados()
- Crie uma classe chamada EmpregadoTeste que instancia a classe Empregado, informa o nome e o salario e executa o método imprimirDados();

Fim