

Ambiente de Programação JAVA

POO

Prof. Marcio Delamaro

Java e POO

- Antes de entrarmos propriamente em POO vamos ver algumas ferramentas
- Elas estão mais ligadas à linguagem Java
- Compilador, editor, IDE, etc

Java oracle

- Existem diversos compiladores/ferramentas Java
- O mais utilizado (?) é o da Oracle
- Java Platform, Standard Edition (JDK)
- <http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>
- Conjunto de ferramentas e código que permite o desenvolvimento de aplicações Java
- JVM

Primeiro programa

```
public class MeuPrograma
{
    public static void main(String args[])
    {
        System.out.println("Meu primeiro programa Java");
    }
}
```

Compilando

- > `javac MeuPrograma.java`
 - Vai gerar o arquivo `MeuPrograma.class`
- Esse arquivo é o arquivo “executável” do seu programa Java
- Na verdade, ele contém uma representação intermediária do seu programa
- Essa representação precisa da JVM

Compilando

- > `javac MeuPrograma.java`
 - Vai gerar o arquivo `MeuPrograma.class`
- Esse arquivo é o arquivo “executável” do seu programa Java
- Na verdade, ele contém uma representação intermediária do seu programa
- Essa representação precisa da JVM
- O que tem de especial o nome do arquivo????

Nome do arquivo .java

- O nome que usamos para o arquivo deve ser o mesmo que usamos para a classe
- Classe: **MeuPrograma**
Arquivo: **MeuPrograma.java**

Nome do arquivo .java

- O nome que usamos para o arquivo deve ser o mesmo que usamos para a classe
- Classe: `MeuPrograma`
Arquivo: `MeuPrograma.java`
- `> javac hello.java`

Nome do arquivo .java

- O nome que usamos para o arquivo deve ser o mesmo que usamos para a classe
- Classe: `MeuPrograma`
Arquivo: `MeuPrograma.java`
- `> javac hello.java`

hello.java:2: error: class MeuPrograma is public, should be declared in a file named MeuPrograma.java

```
public class MeuPrograma
```

^

1 error

Executando

- > `java MeuPrograma`
 - Vai procurar o arquivo `MeuPrograma.class` e vai executá-lo
- Essa é a JVM. Em tese, ela interpreta o código Java.
- Write once, run anywhere (WORA).

Código intermediário

```
public class MeuPrograma {
```

```
    public MeuPrograma();
```

```
    Code:
```

```
        0: aload_0
```

```
        1: invokespecial #1
```

```
        4: return
```

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

```
    Code:
```

```
        0: getstatic    #2
```

```
        3: ldc          #3
```

```
        5: invokevirtual #4
```

```
        8: return
```

```
}
```

Classes x arquivos

- Cada classe é armazenada em um arquivo
- Todos os arquivos estão armazenados em um diretório
- Esse é o **classpath** onde a JVM procura as classes

Classes x arquivos

- Cada classe é armazenada em um arquivo
- Todos os arquivos estão armazenados em um diretório
- Esse é o **classpath** onde a JVM procura as classes
- **> javac MyCl1.java MyCl2.java**
> java MyClc1
- Por padrão o classpath é o diretório corrente

Parâmetros

- Tanto o javac quanto a JVM aceitam vários parâmetros
- `> javac -help`

Parâmetros

- Tanto o javac quanto a JVM aceitam vários parâmetros
- `> javac -help`
- `> javac -sourcepath src -d bin src/MeuPrograma.java`
- Vai pegar os arquivos fontes de src e gerar os executáveis em bin

Parâmetros

- Tanto o javac quanto a JVM aceitam vários parâmetros
- > javac -help
- > javac -sourcepath src -d bin src/MeuPrograma.java
- Vai pegar os arquivos fontes de src e gerar os executáveis em bin
- > java -cp bin MeuPrograma
- Vai executar o programa usando bin como o classpath

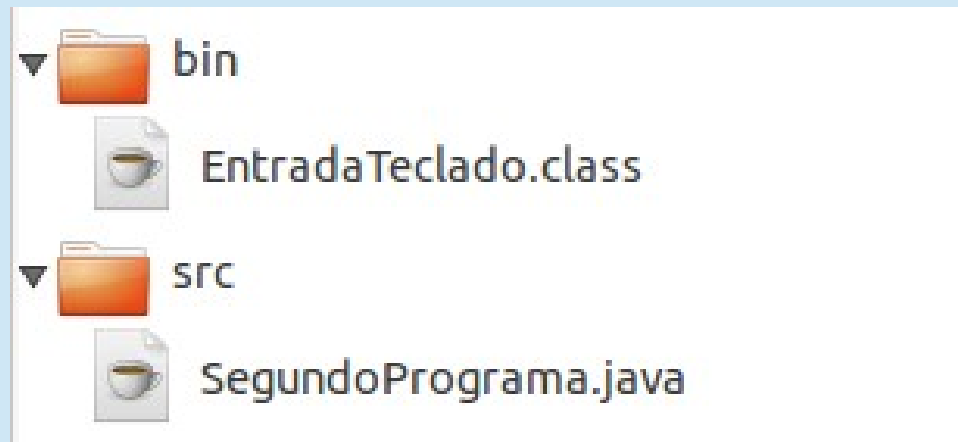
Segundo Programa

- Nesse caso vamos usar uma classe que ajuda a ler coisas do teclado
- A classe é chamada *EntradaTeclado*
- Vamos usar o método *leInt()* que lê um número inteiro e retorna o seu valor

Segundo Programa

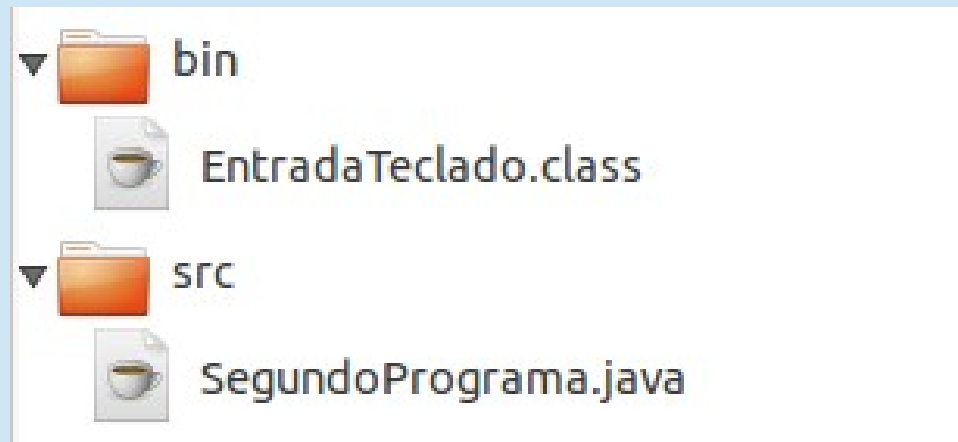
```
public class SegundoPrograma {  
  
    public static void main(String[] args) throws Exception  
    {  
        System.out.printf("Digite um inteiro: ");  
        int k = EntradaTeclado.leInt();  
        System.out.printf("Numero lido: %d\n", k);  
    }  
}
```

Arquivos do Segundo Programa



- A classe auxiliar (somente o executável) está no diretório bin
- O arquivo fonte está no diretório src

Arquivos do Segundo Programa



- A classe auxiliar (somente o executável) está no diretório ***bin***
- O arquivo fonte está no diretório ***src***
- Como compilar e executar?

Compilar e Executar

- Compilar:
 - `> javac -sourcepath src -d bin src/SegundoPrograma.java`

- Executar:

Compilar e Executar

- Compilar:
 - > javac -sourcepath src -d bin src/SegundoPrograma.java
 - > javac -sourcepath src -d bin -cp bin src/SegundoPrograma.java

- Executar:

Compilar e Executar

- Compilar:
 - > javac -sourcepath src -d bin src/SegundoPrograma.java
 - > javac -sourcepath src -d bin -cp bin src/SegundoPrograma.java
- Executar:
 - java -cp bin SegundoPrograma

IDE

- Integrated Development Environment
- Permite que se faça tudo em um único ambiente
- Permite a instalação de plugins
- Eclipse e Netbeans, por exemplo

Eclipse

- Escolher um diretório para ser seu workspace
- Ao entrar é possível sempre escolher um workspace
- Dentro do workspace são criados os seus projetos
- O nome do projeto corresponde a um subdiretório dentro do workspace

Projeto MeuPrograma

- Vamos criar um projeto para o nosso primeiro programa
- Entrar no Eclipse e criar um workspace SSC103
- Criar um um projeto **MyProg**
- Criar Uma classe **MeuPrograma**
- Executar

Exercício

- Crie um projeto para executar o nosso segundo programa. Coloque os dois programas fontes no diretório src.
- Crie um projeto para executar o nosso segundo programa. Use apenas o arquivo .class para a class EntradaTeclado. Para fazer isso, crie dentro do projeto um diretório “lib” e use esse diretório como parte do seu “Build Path”.
-