Java Básico

Aula 1

por Thalles Cezar

Quem sou eu?

- Thalles Cezar, muito prazer!
- Estudante de Ciências da Computação, indo para o 9º semestre.
  - Quase lá!! =)
- Estagiário no projeto CIn/Samsung TVD.
- Professor desse curso pela 4ª vez.

Quem são vocês?

O Curso

- Total de 10 dias de aulas.
  - 40 horas.
- Número máximo de faltas permitidas = 3
- Aulas com teoria e prática.
  - Pratiquem!!!

O Curso

- No último dia de aula, haverá uma **prova!!**
  - 20 questões objetivas!

Introdução

- Programação?

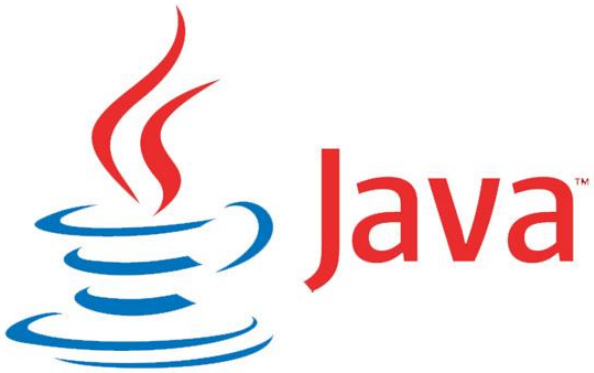
*“**Programação** é o processo de escrita, teste e manutenção de um programa de computador.”*

Introdução

- Computadores NÃO são inteligentes.
  - Até podem ficar, mas isso é outra história.
- Os computadores apenas fazem o que são mandados.
- Para mandar, precisamos falar o idioma dos computadores.

Introdução

- Para isso utilizamos as Linguagens de Programação.



Introdução

- E como vamos passar as ordens?
  - Lembrem-se, computadores não pensam!

- Utilizando ALGORITMOS.

*"Um **algoritmo** é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finita."*

Introdução

- Para entender melhor, utilizaremos o **pseudocódigo.**

- Então, vamos tentar de uma maneira prática:
  - Preciso de um voluntário (y)

Introdução

- As únicas instruções que o voluntário sabem são:
  - Vire para a direita;
  - Vire para a esquerda;
  - Ande;
  - Levante;
- Agora, façam com que ele chegue até a porta e depois retorne a sua cadeira!

Introdução

- Essa foi muito fácil!
- Agora, vamos montar um algoritmo, utilizando pseudocódigo, que vai receber 3 números e calcular sua média.
  - escreva – Escreve na tela do computador;
  - leia – Lê o número passado pelo usuário;
  - soma – Realiza a soma de dois números inteiros;
  - divide – divide dois números inteiros;
  - var - utilizado para criar uma variável.

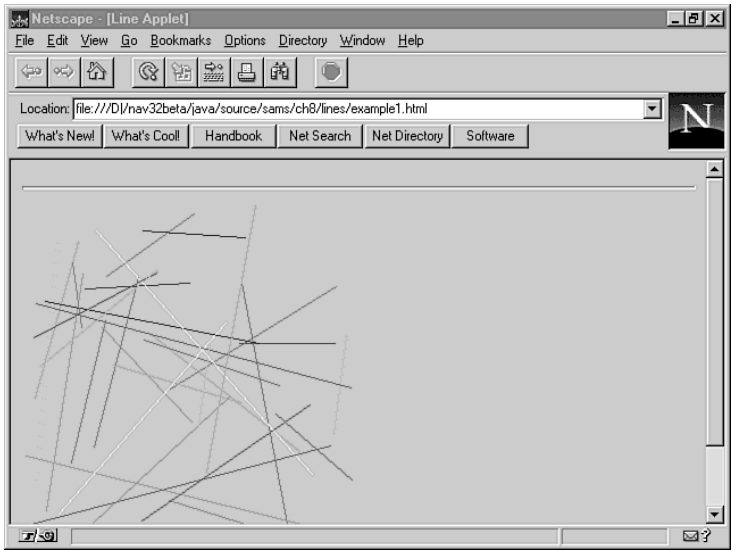

EXI

Histórico

- Green Project (1991)
  - Desenvolver plataforma para eletrodomésticos inteligentes
  - Tentaram usar C++
  - Mas resolveram desenvolver uma linguagem própria
- Oak language (1992)
  - Já rodava no PDA Star7
  - Também rodavam ela em um decodificador de TV a cabo

Histórico

- Java na Web (1994)
  - Advento dos browsers
  - Internet se tornava mais iterativa do que a TV a cabo
- Java no Netscape (1995)



Histórico

- JDK é lançado (1996)
- JavaOne (1997)
  - A maior conferência de desenvolvedores do mundo (8.000 visitantes)
- 75% dos desenvolvedores usam Java como sua primeira linguagem (2003)
- Open Source (2006)

Histórico

- Várias edições
  (abaixo: Java Micro Edition, Android e Java Enterprise Edition)

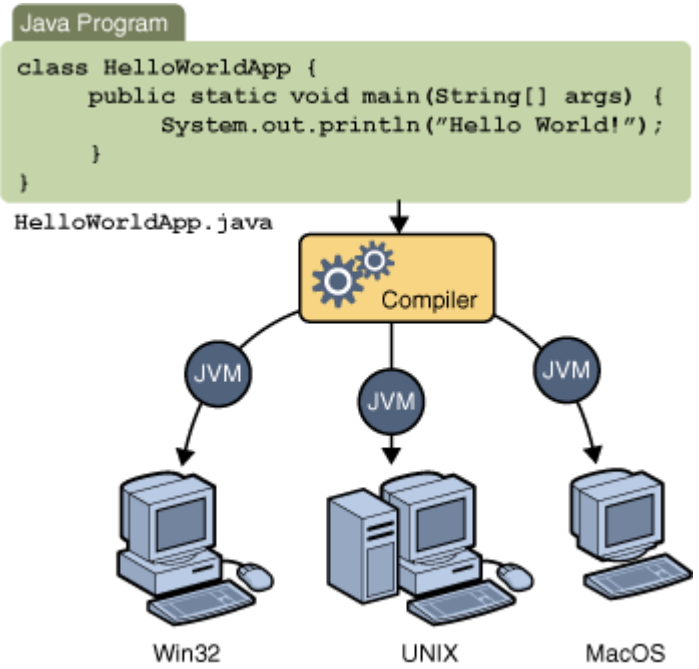
O que exatamente é Java?

- Duas coisas:
  - A linguagem de programação Java

  - A plataforma Java
    - Java Virtual Machine (Máquina Virtual Java)
    - Java Application Programming Interface (API Java)

O que exatamente é Java?

- Através de uma máquina virtual, o mesmo código em Java pode ser rodado em diferentes arquiteturas de software e hardware!



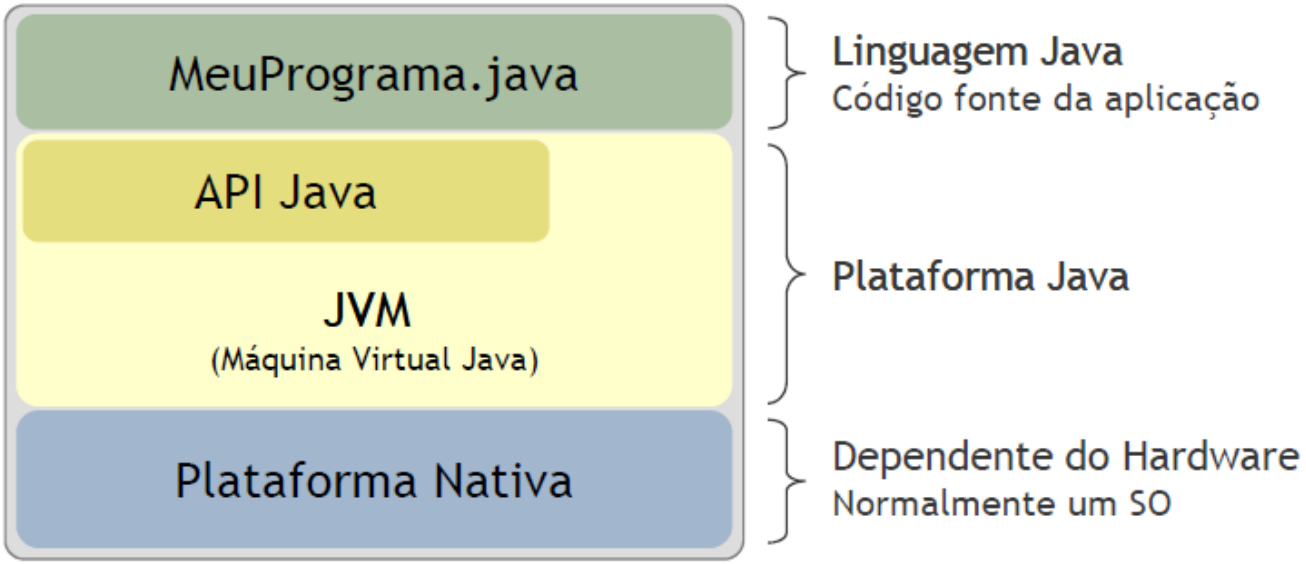
O que exatamente é Java?

- Essa **portabilidade** só é possível porque os compiladores Java geram códigos intermediários (**bytecodes**) que por sua vez são rodados nas máquinas virtuais Java

- Os **bytecodes** são armazenados em arquivos .class

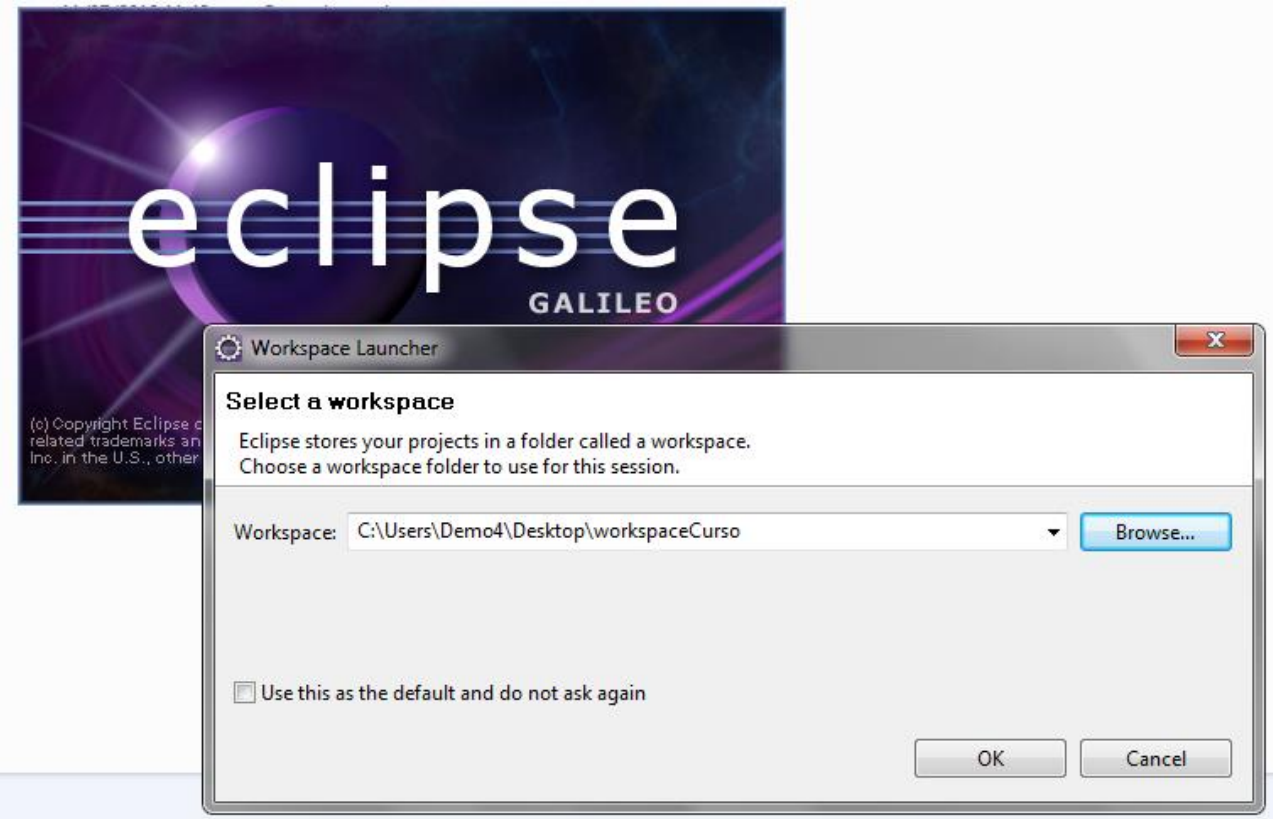
O que exatamente é Java?

- A API Java é uma grande coleção de componentes de softwares que podemos usar para nos ajudar a construir nossos programas!
- Na API Java temos componentes que nos ajudam a programar jogos, interfaces, banco de dados, etc...



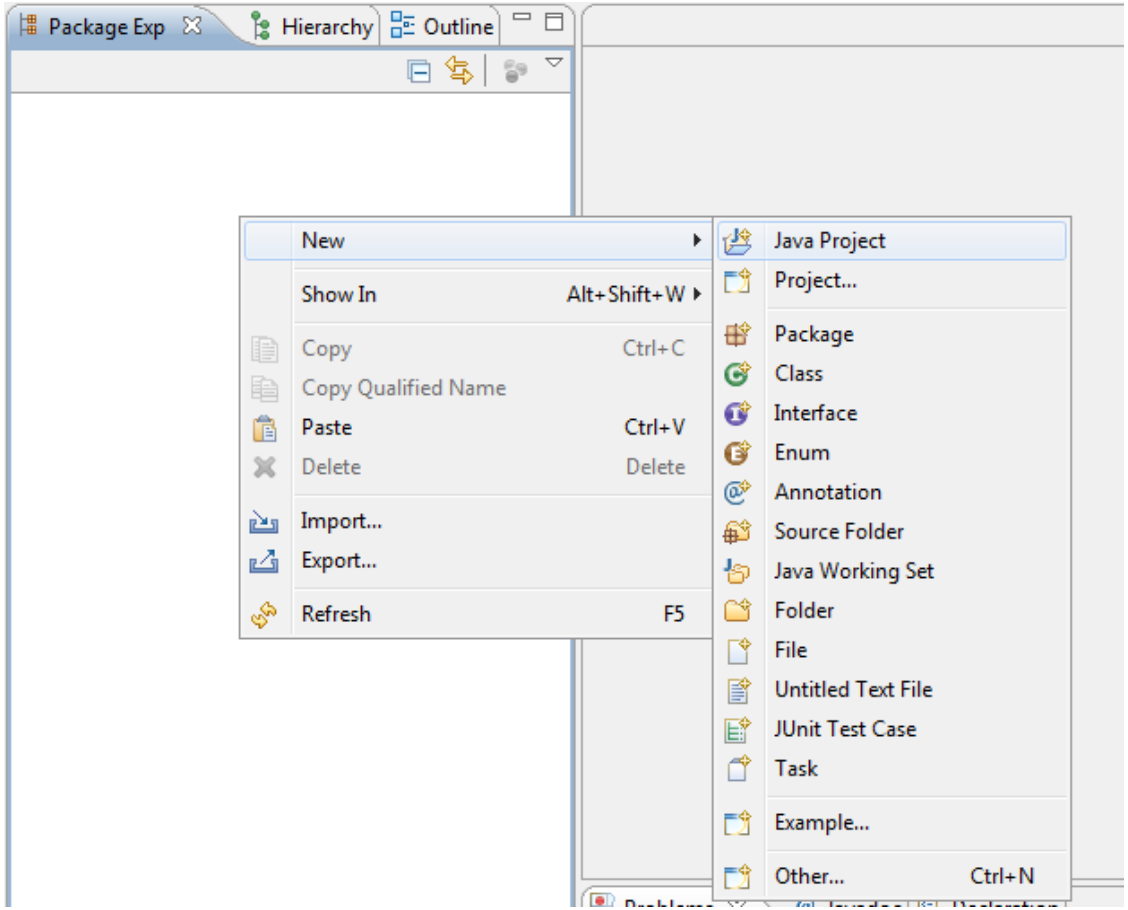
Trabalhando com a IDE Eclipse

- Vamos abrir o **Eclipse**



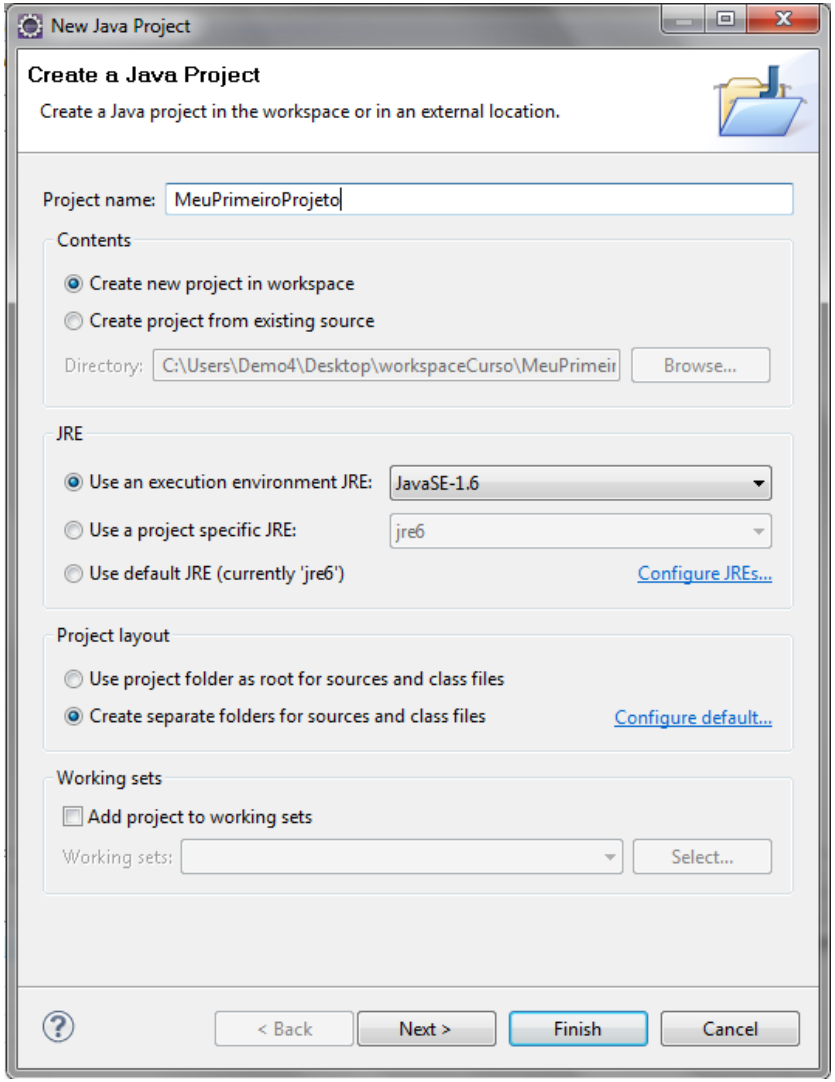
Trabalhando com a IDE Eclipse

- Agora apertamos com o botão direito no **Package Explorer** para criarmos um novo projeto Java...



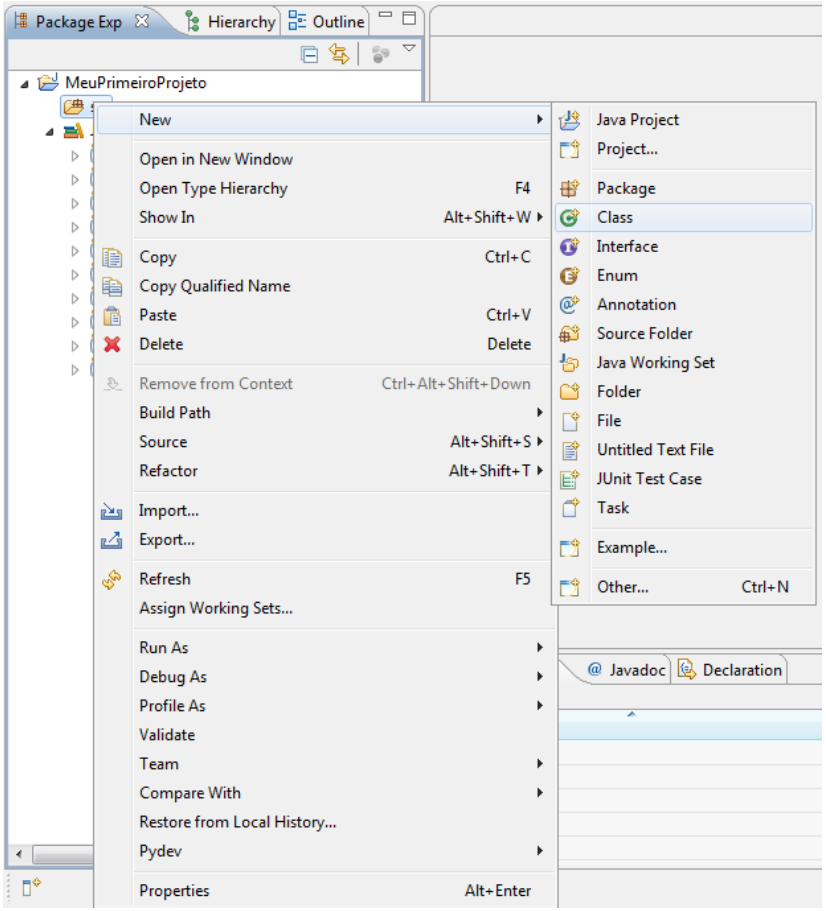
Trabalhando com a IDE Eclipse

- Dê um nome para seu projeto e clique em **Finish**!



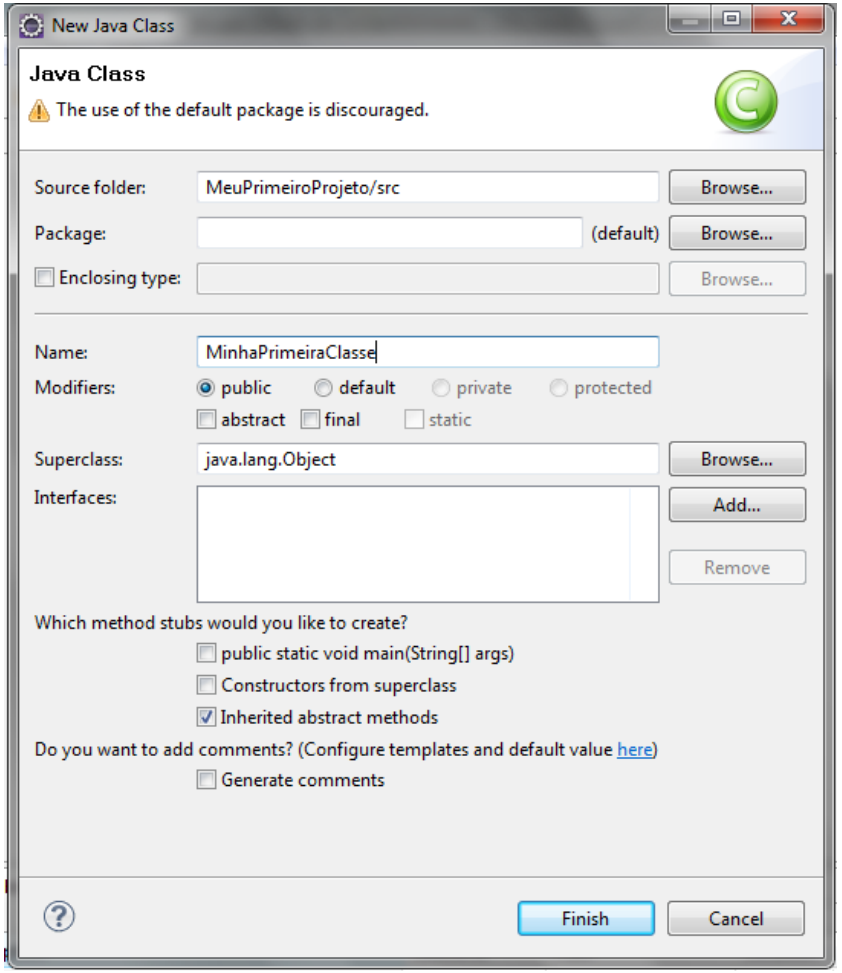
Trabalhando com a IDE Eclipse

- Aperte com o botão direito em **src**, abra **New** e escolha a opção **Class**



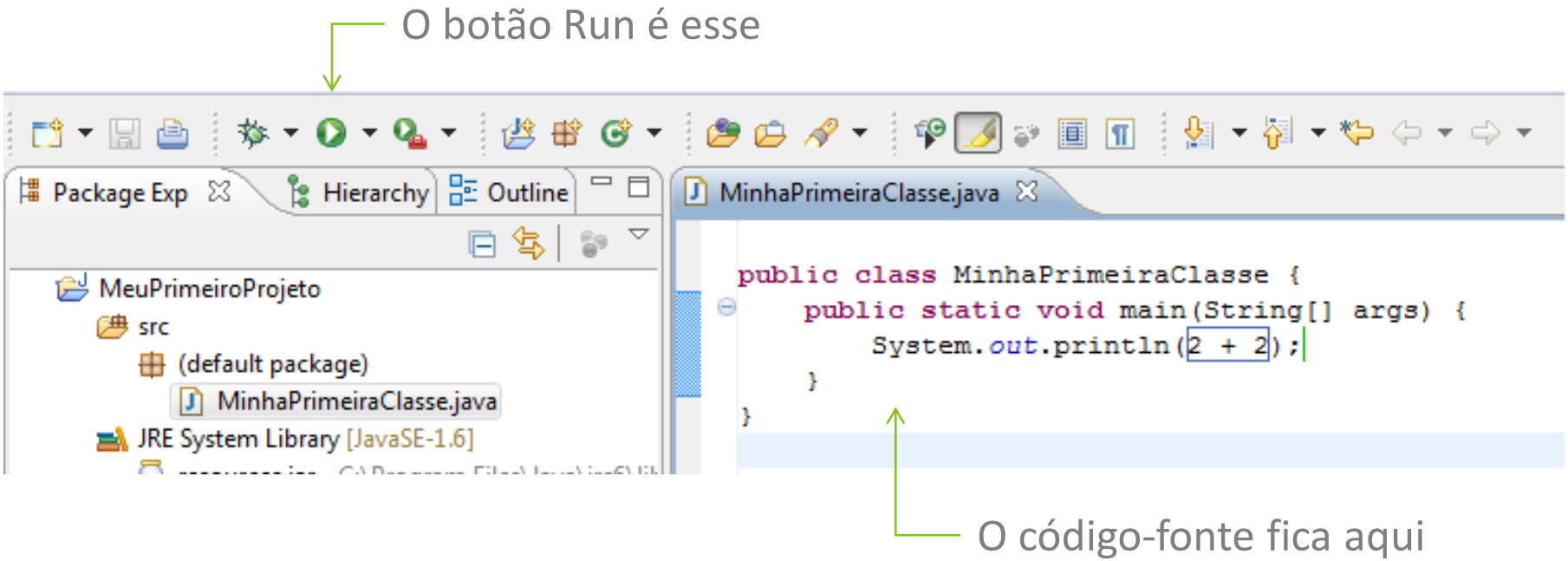
Trabalhando com a IDE Eclipse

- Dê um nome para sua primeira classe seguindo ao padrão de nomenclatura de classes de Java
- O padrão é o seguinte:
  - Todas palavras juntas
  - A cada nova palavra colocar primeira letra como maiúscula
  - Ex: MinhaPrimeiraClasse



Trabalhando com a IDE Eclipse

- Agora é programar e apertar no Run quando terminar!



Programando

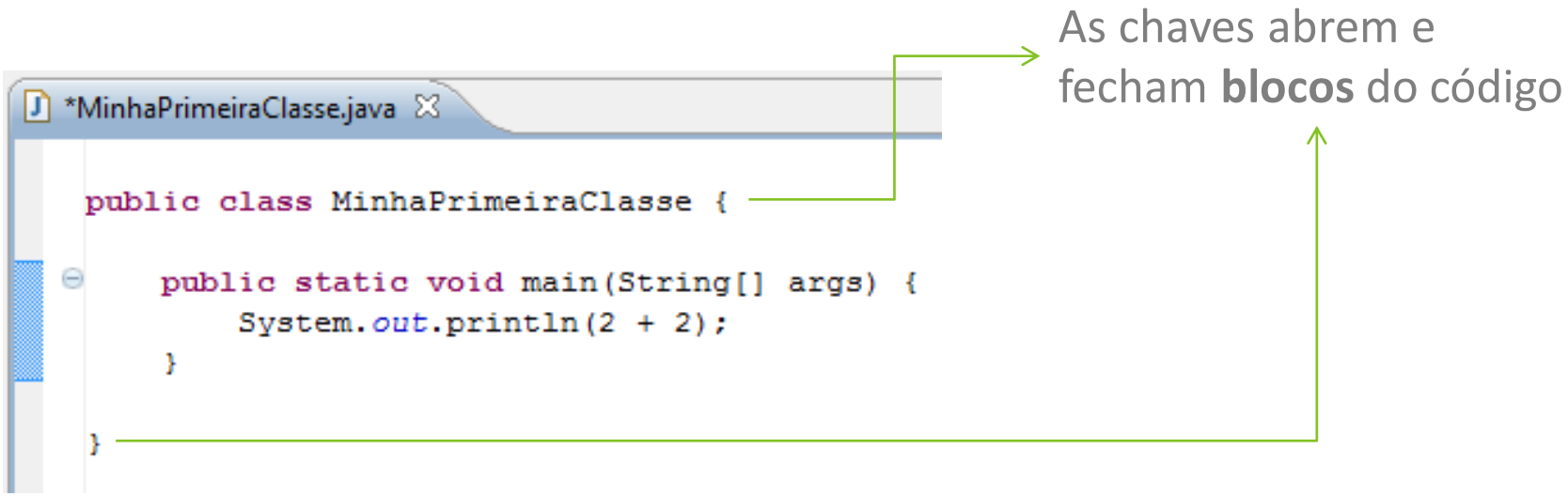
- Vamos ver um exemplo de código e tentar entender o que ele faz...

*MinhaPrimeiraClasse.java

```
public class MinhaPrimeiraClasse {

    public static void main(String[] args) {
        System.out.println(2 + 2);
    }

}
```

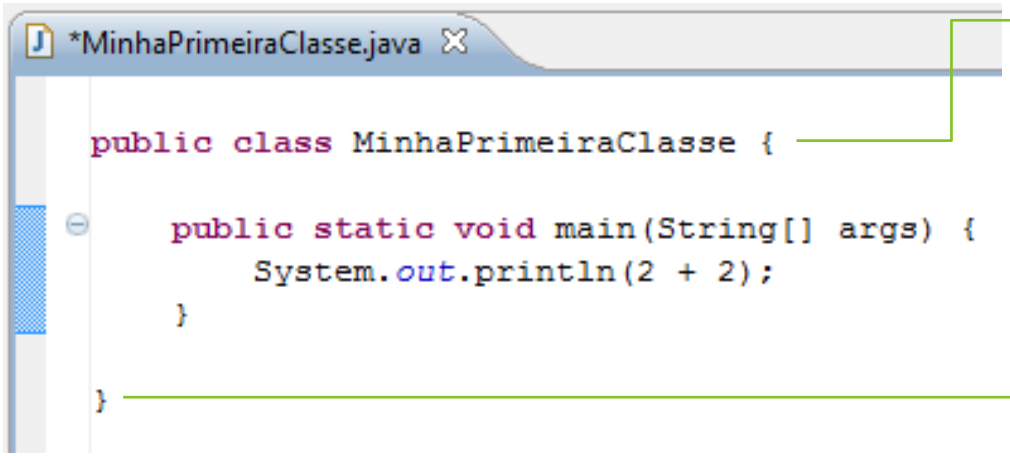
Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...



Programando

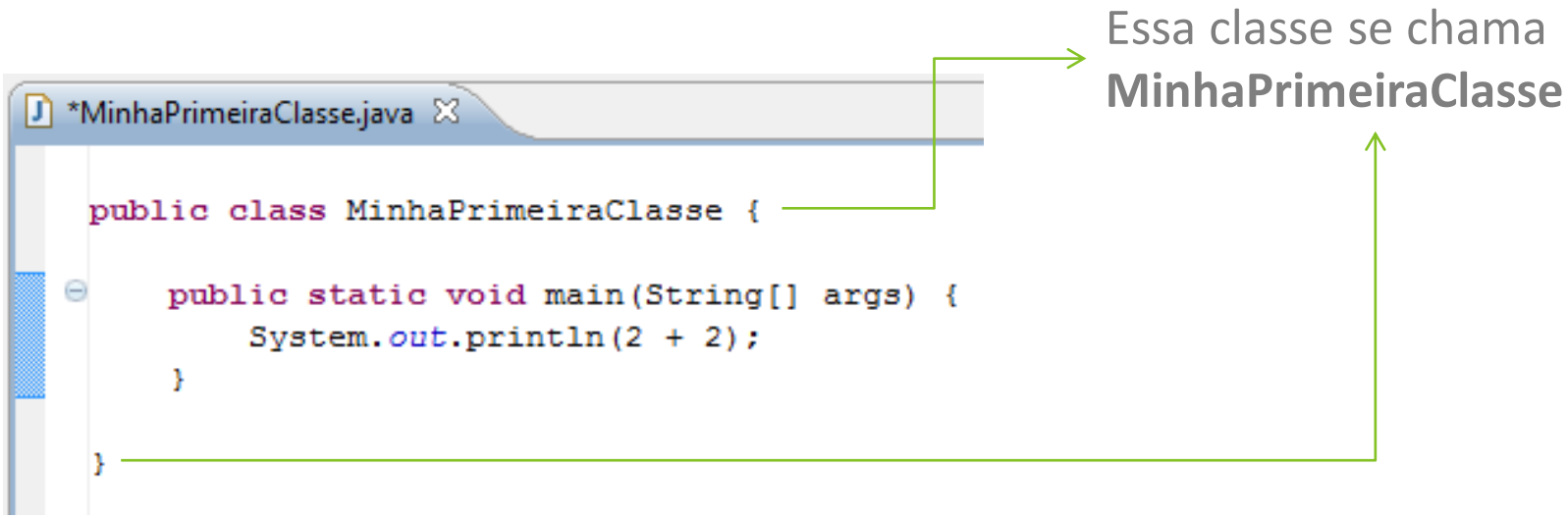
- Vamos ver um exemplo de código e tentar entender o que ele faz...



Nesse caso, as chaves abrem e fecham uma **classe**

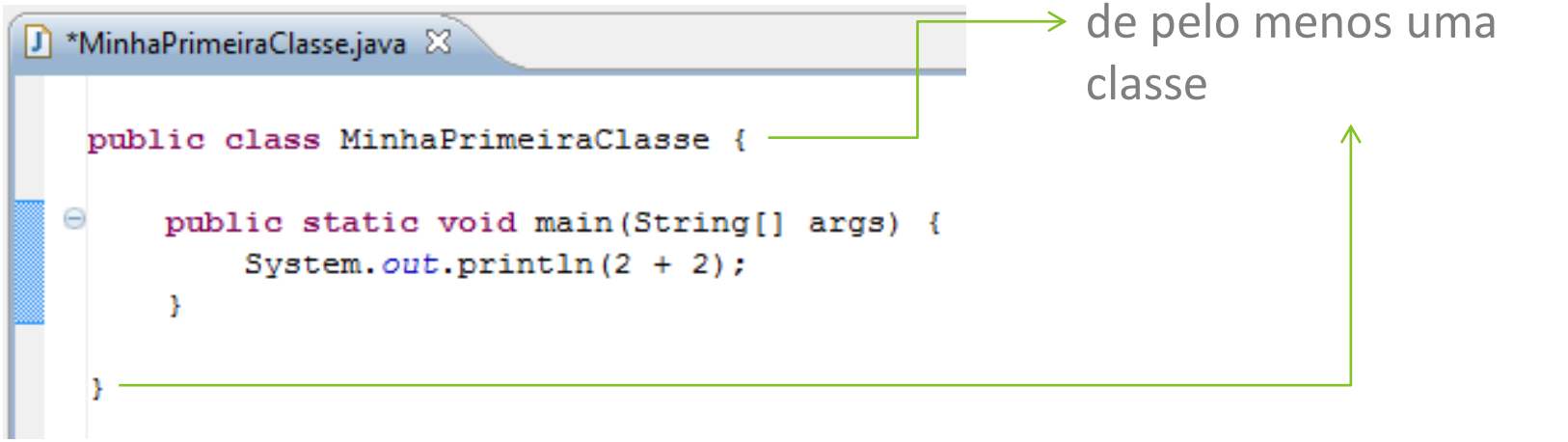
Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...



Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...



Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...

*MinhaPrimeiraClasse.java

```
public class MinhaPrimeiraClasse {

    public static void main(String[] args) {
        System.out.println(2 + 2);
    }

}
```

Agora as chaves abrem e fecham um **método**

Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...

*MinhaPrimeiraClasse.java

```
public class MinhaPrimeiraClasse {

    public static void main(String[] args) {
        System.out.println(2 + 2);
    }

}
```

Esse método se chama **main**, é ele que vai ser executado quando o programa for executado

Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...

*MinhaPrimeiraClasse.java

```
public class MinhaPrimeiraClasse {

    public static void main(String[] args) {
        System.out.println(2 + 2);
    }

}
```

A única coisa que o main de MinhaPrimeiraClasse faz é **chamar** o método System.out.println

Programamando

- Vamos ver um exemplo de código e tentar entender o que ele faz...

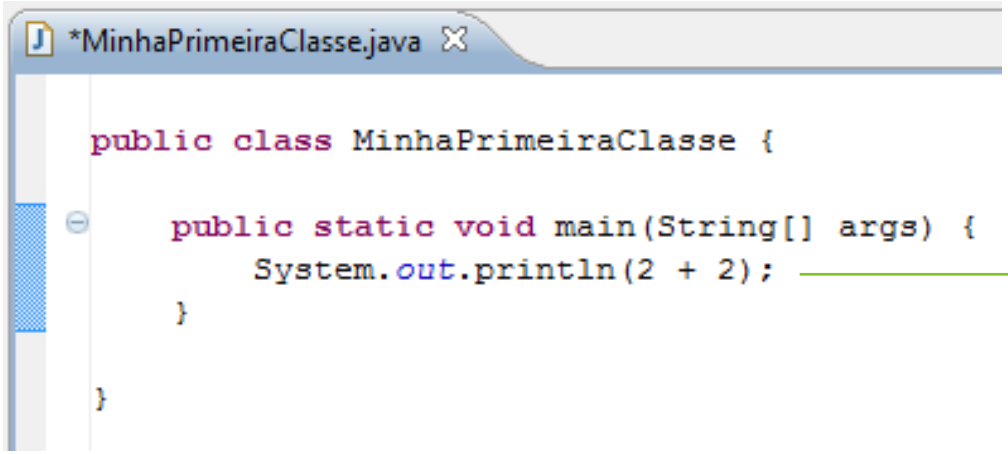
*MinhaPrimeiraClasse.java

```
public class MinhaPrimeiraClasse {

    public static void main(String[] args) {
        System.out.println(2 + 2);
    }

}
```

O método System.out.println é um método da API Java! Ele serve para colocar resultados na tela, ou seja, **imprimir** resultados

Programando

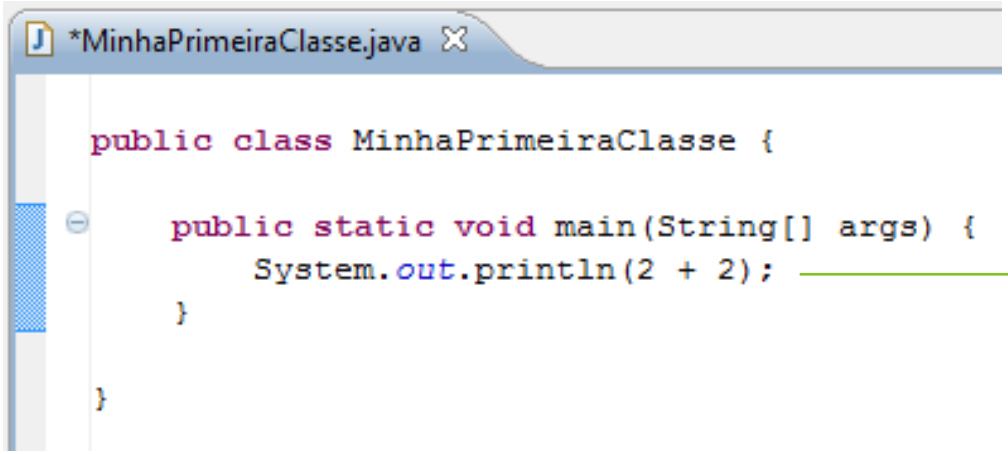
- Vamos ver um exemplo de código e tentar entender o que ele faz...



Ao chamar esse método, dizemos ao computador que ele deve imprimir a **expressão** que colocarmos entre parênteses

Programando

- Vamos ver um exemplo de código e tentar entender o que ele faz...

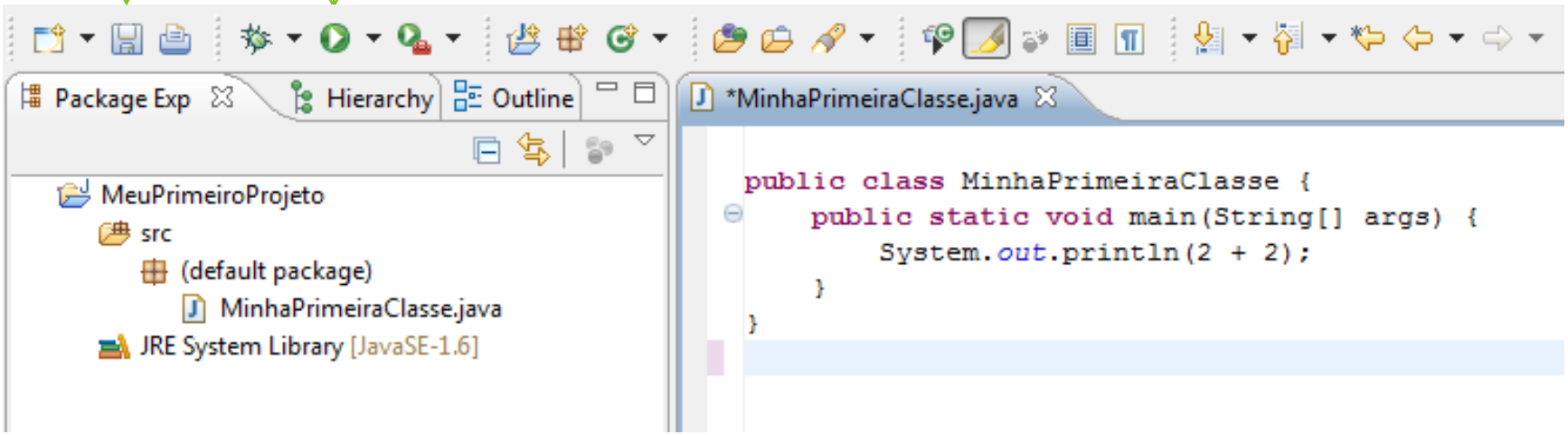


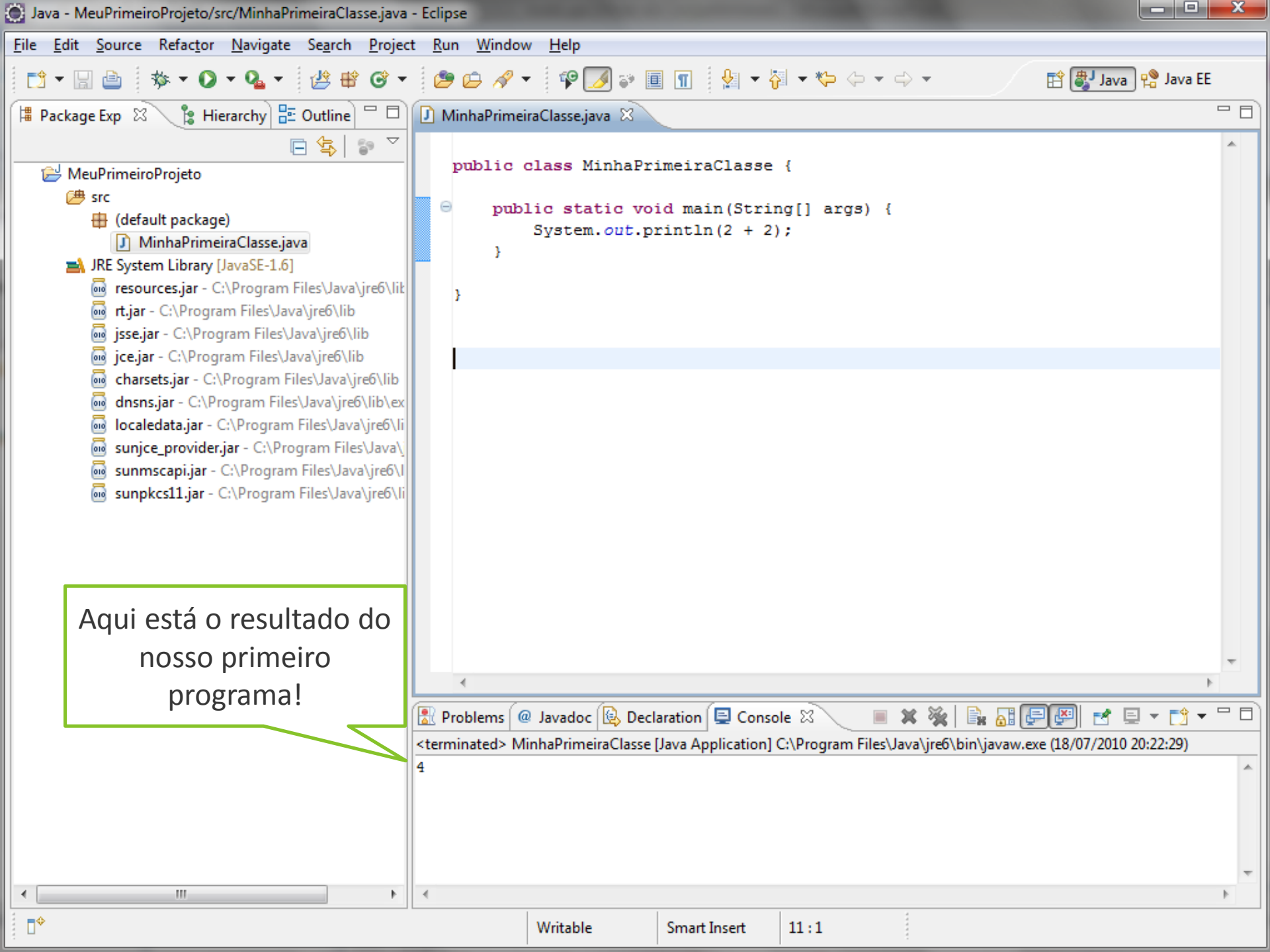
A expressão que colocamos entre parênteses é 2 + 2. O computador **avaliará** essa expressão e a imprimirá.

Programando

Vamos salvar as alterações (se necessário)...

...e rodar o código!

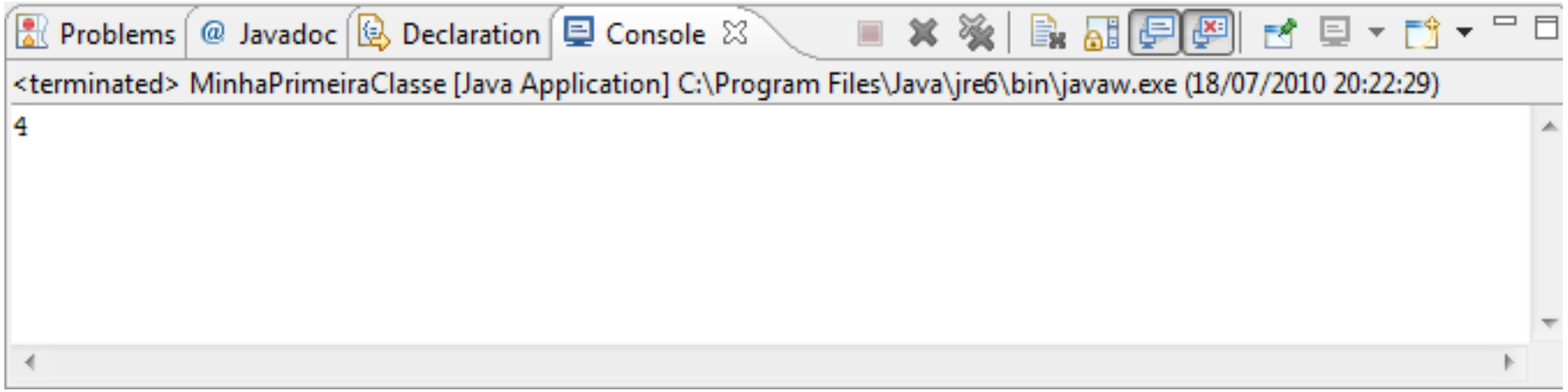




Aqui está o resultado do nosso primeiro programa!

Programando

- Viram o resultado? Ele aparece no **Console**. É no console que vemos o resultado em forma de texto dos programas. Aqui nós vemos o número 4 como um texto no **Console**, mas poderíamos vê-lo em uma **janela** também.



Referências

- http://en.wikipedia.org/wiki/Java_(software_platform)#History
- http://www.java.com/en/javahistory/timeline.jsp
- http://developer.android.com/guide/basics/what-is-android.html
- https://www.cs.auckland.ac.nz/references/java/java1.5/tutorial/getStarted/intro/definition.html
- https://www.cs.auckland.ac.nz/references/java/java1.5/tutorial/getStarted/intro/cando.html
- http://www.eclipse.org/