

Java

Primeiro Programa em Java

Um programa em Java é composto de Pacotes; sendo assim, o primeiro comando a ser utilizado é “package primeiroprograma;”. (Nesse primeiro programa o comando é opcional).

```
package primeiroprograma;
```

O primeiro comando que toda Classe tem que ter é justamente o Cabeçalho da Classe, “public class PrimeiroPrograma {”.

```
package primeiroprograma;  
  
public class PrimeiroPrograma {
```

É importante fechar as chaves algumas linhas abaixo.

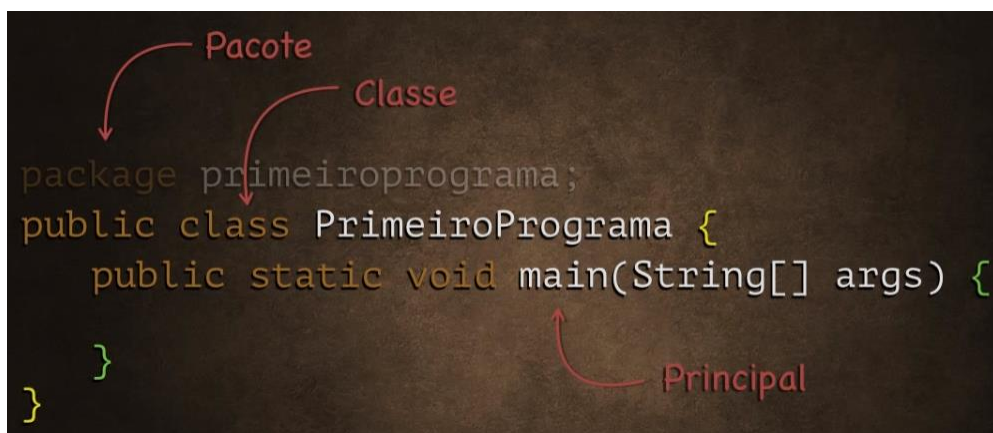
As Chaves representam Blocos, ou seja, uma Classe em Java é um Bloco.

```
package primeiroprograma;  
  
public class PrimeiroPrograma {  
  
  
  
}
```

Dentro da Classe vamos criar um Método.

```
package primeiroprograma;  
  
public class PrimeiroPrograma {  
  
    public static void main(String[]  
args) {  
  
    }  
}
```

Para melhor entendermos, vamos traduzir...



O diagrama mostra o código Java anterior com anotações em português explicando a estrutura:

- Pacote**: Uma seta vermelha aponta para a linha `package primeiroprograma;`.
- Classe**: Uma seta vermelha aponta para a linha `public class PrimeiroPrograma {`.
- Principal**: Uma seta vermelha aponta para a linha `public static void main(String[] args) {`.

O código completo exibido no diagrama é:

```
package primeiroprograma;  
public class PrimeiroPrograma {  
    public static void main(String[] args) {  
    }  
}
```

Logo, temos um Método Principal, dentro de uma Classe PrimeiroPrograma que faz parte do Pacote primeiroprograma.

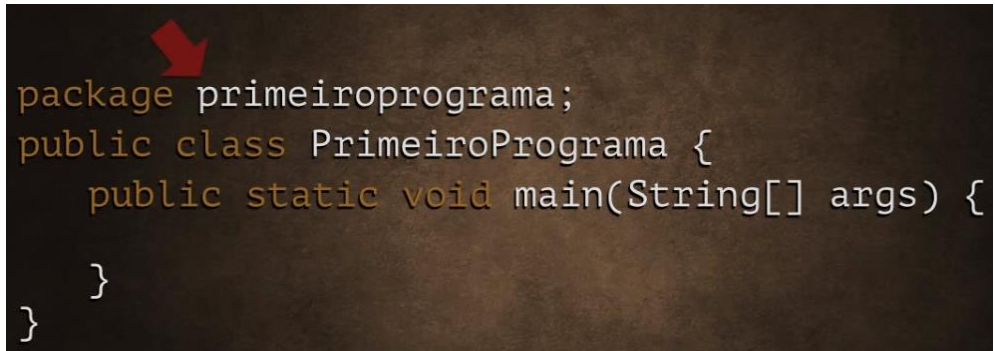
Sendo assim, temos nessa hierarquia Métodos; as Classes são compostas de Métodos e podemos ter várias Classes dentro dos Pacotes.

Novamente, temos um Método => dentro de uma Classe => e essa Classe faz parte de um Pacote de Classes.

Técnicas de Nomeação de Identificadores

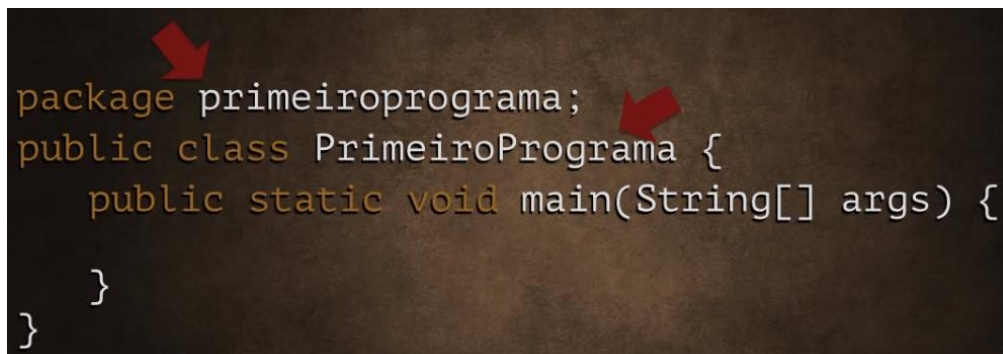
Java é case sensitive! Ou seja, irá diferenciar letras maiúsculas e minúsculas.

Observe que em package, “primeiroprograma” foi escrito todo em letra minúscula



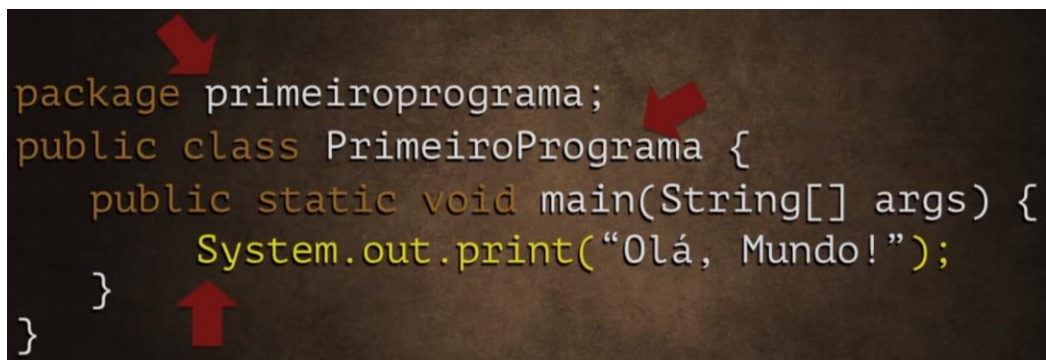
```
package primeiroprograma;  
public class PrimeiroPrograma {  
    public static void main(String[] args) {  
  
    }  
}
```

Já o Nome da Classe, “PrimeiroPrograma”, tem as primeiras maiúsculas.



```
package primeiroprograma;  
public class PrimeiroPrograma {  
    public static void main(String[] args) {  
  
    }  
}
```

O primeiro comando que iremos utilizar tem a função de escrever na tela; sendo este:
System.out.print(“Olá, Mundo!”);

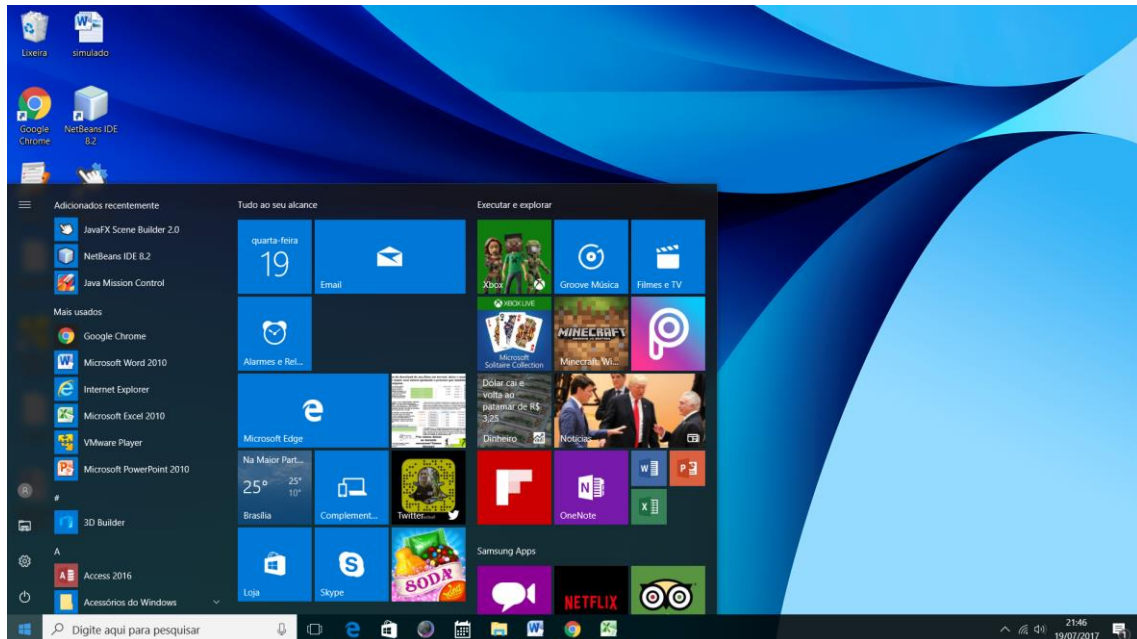


```
package primeiroprograma;  
public class PrimeiroPrograma {  
    public static void main(String[] args) {  
        System.out.print(“Olá, Mundo!”);  
    }  
}
```

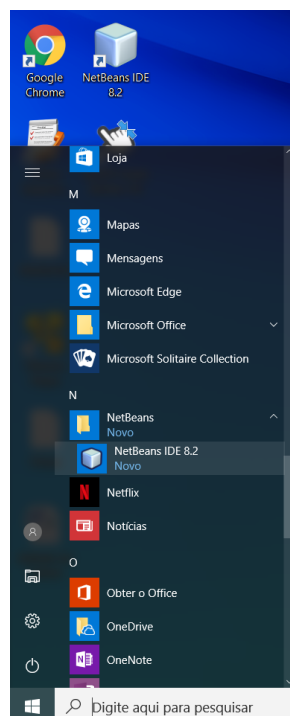
Tarefa 1

Desenvolver um programa que exiba o texto “Olá, Mundo!”.

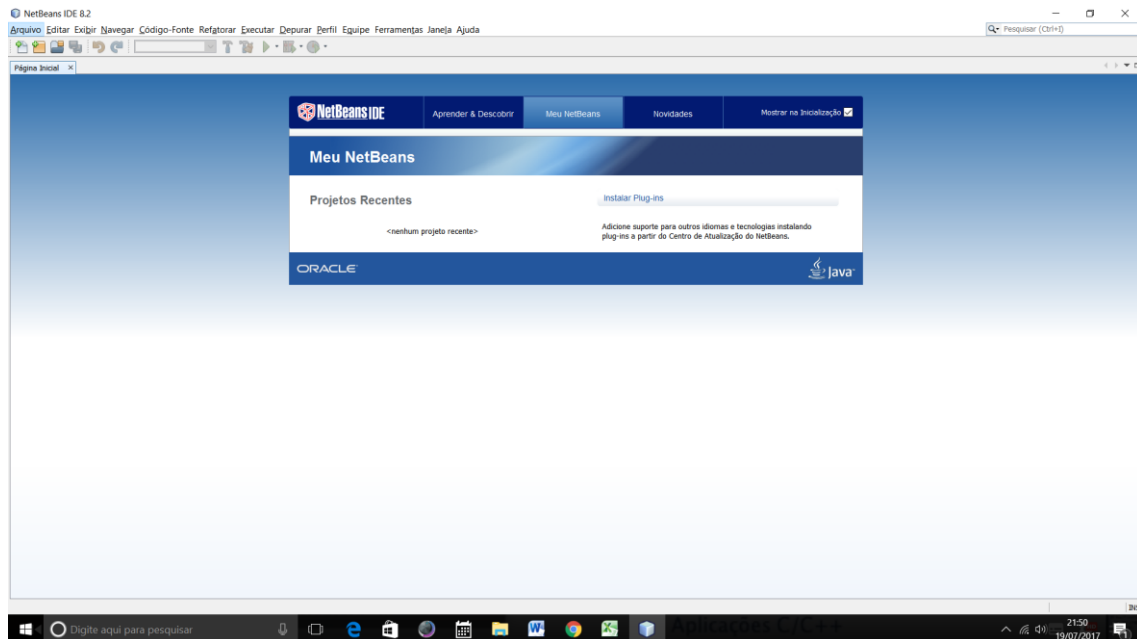
No Windows, abra o Netbeans, que instalamos na aula passada, clicando no botão “Iniciar”.



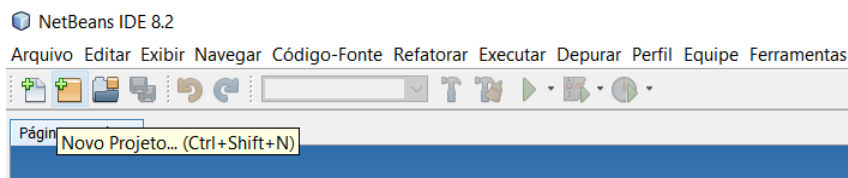
Após, mova a Barra de Rolagem para baixo, expanda a opção Netbeans e clique no ícone do Netbeans.



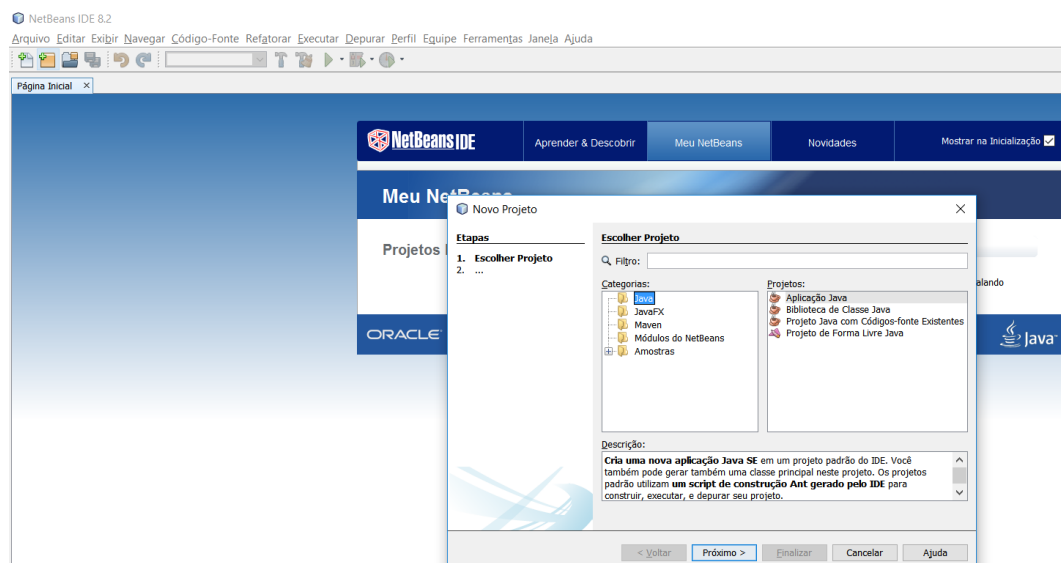
O Netbeans será aberto já com as opções de idioma do seu computador. Neste caso, em Português Brasil.



Na Barra de Ferramentas, clique em Novo Projeto.

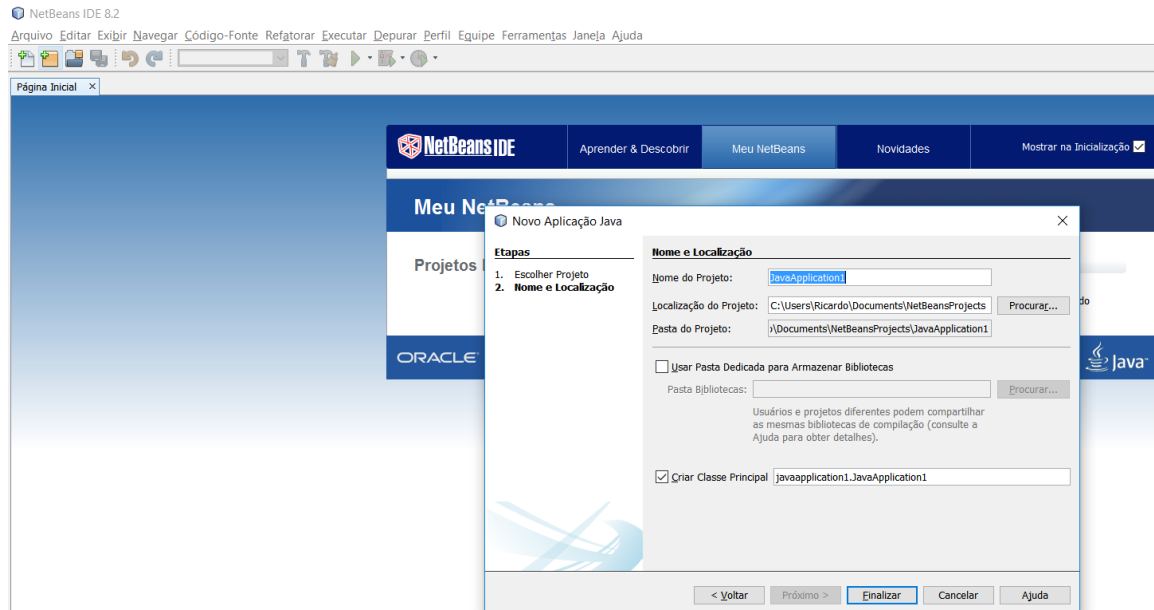


Na etapa “Escolher Projeto”, mantenha selecionada a Categoria “Java” e em Projetos mantenha selecionado “Aplicação Java”. Clique no botão “Próximo”.

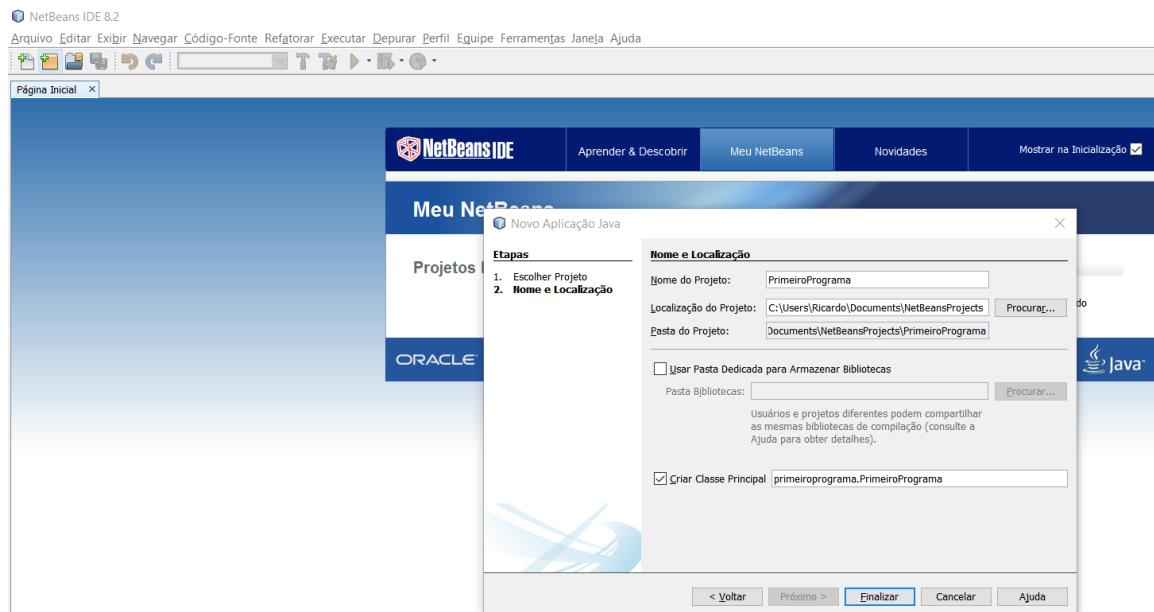


Na Caixa de Texto “Nome do Projeto”, digite “PrimeiroPrograma”.

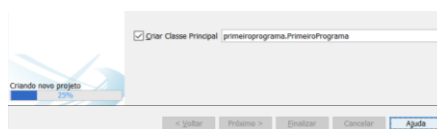
Não pode haver espaço entre as palavras do Nome do Projeto.



Observe que ele já criará a Classe Principal. Mantenha a Caixa de Seleção “Criar Classe Principal” selecionada.



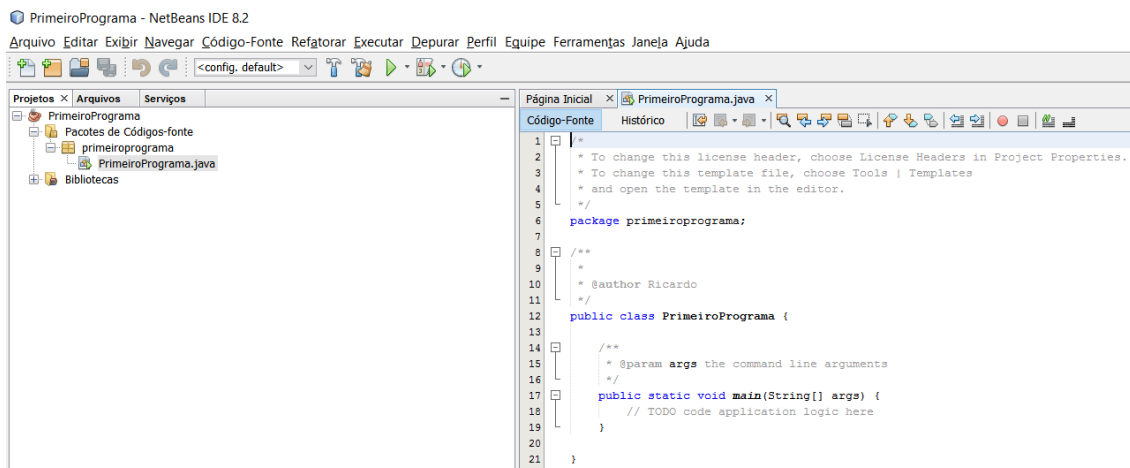
Será exibido o processo de criação do sistema de arquivos. Aguarde a finalização.



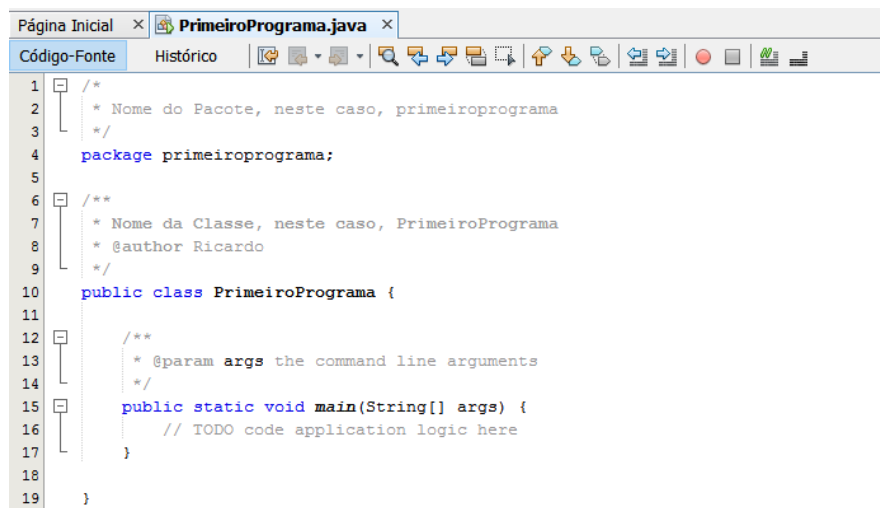
Ao término do processo, será exibida a seguinte tela, contendo o seu programa.

Observe em “PrimeiroPrograma.java”, no Pannel “Projetos” a esquerda, que “.java” representa uma Classe, conforme mostrado no Pannel “Código-Fonte”, “public class PrimeiroPrograma”.

Logo, temos nossa Classe Primeiro.Programa e temos nosso Pacote primeiroprograma visíveis no Pannel esquerdo.



No Pannel “Código-Fonte”, todo o texto com a cor cinza são comentários.

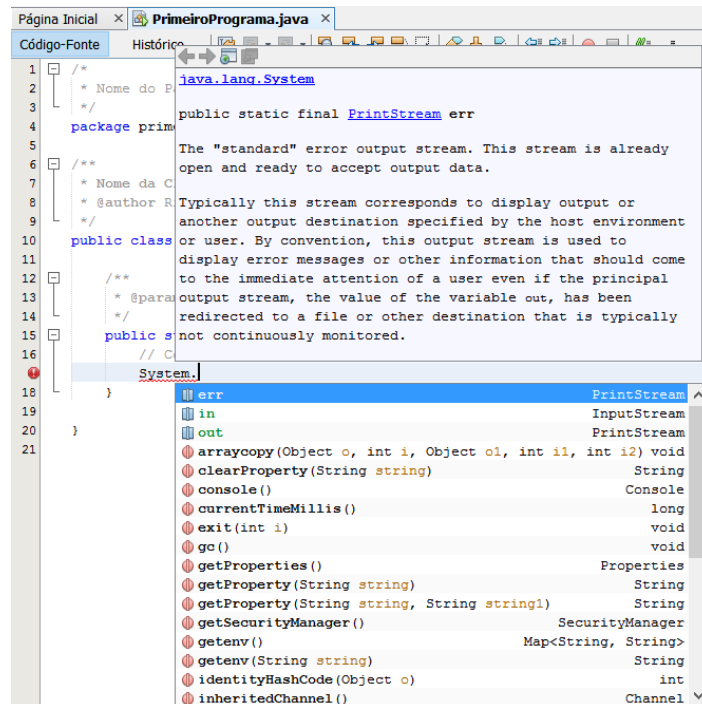


Inserindo comandos no Código Fonte

Não foi necessário digitar todos os comandos listados acima devido a IDE.

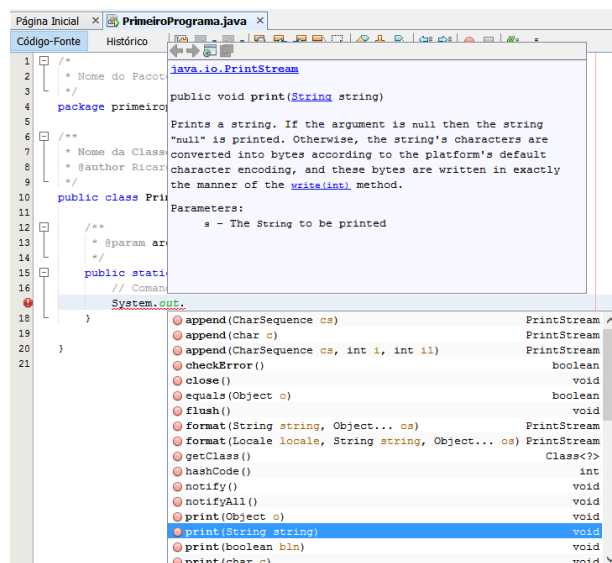
A IDE, também, irá ajudar ao digitar o comando “System.out.print”.

Na linha abaixo de “public static”, digite “System.” e observe que haverá uma ajuda do próprio sistema para completar o comando. No nosso caso, basta clicar em “out”.

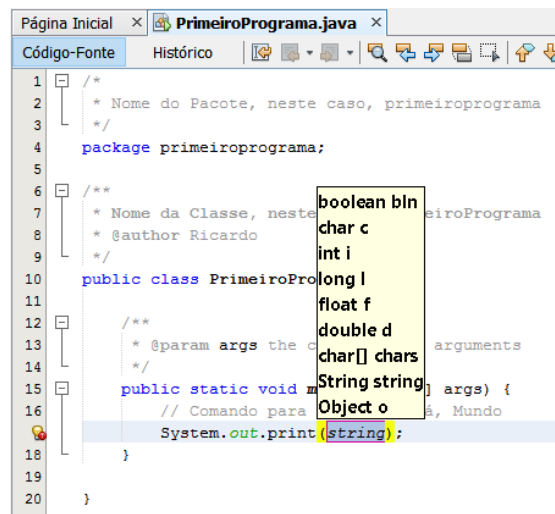


Após clicar em “out”, digite “.” (ponto) e novamente o Netbeans irá fornecer uma ajuda para completar o comando, conforme imagem que segue.

Clique em print(String string), conforme a imagem.

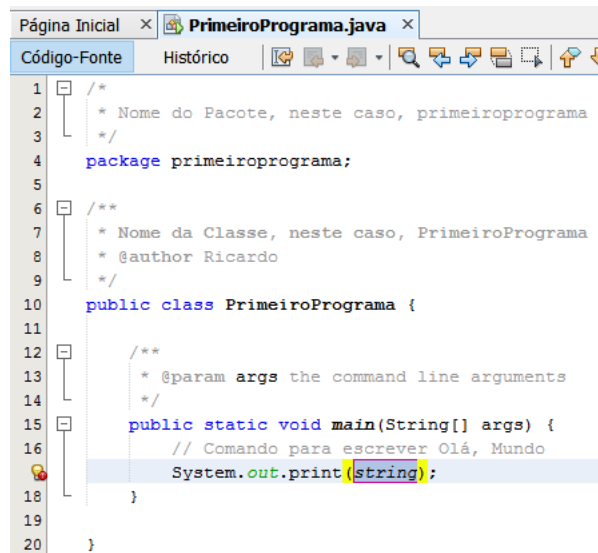


O programa questionará o formato do dado, sugerindo que seja uma “string”, ou seja, texto.



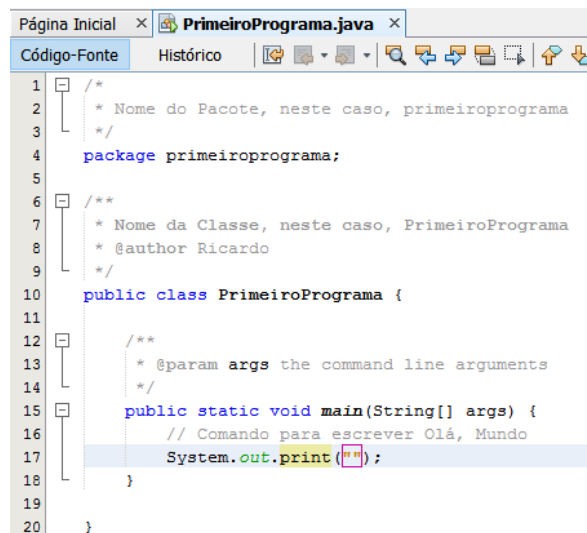
```
1  /*
2  * Nome do Pacote, neste caso, primeiroprograma
3  */
4  package primeiroprograma;
5
6  /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         // Comando para escrever Olá, Mundo
17         System.out.print(string);
18     }
19 }
20
```

Apague o texto “string” e digite “Olá, mundo!”, com as aspas.



```
1  /*
2  * Nome do Pacote, neste caso, primeiroprograma
3  */
4  package primeiroprograma;
5
6  /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         // Comando para escrever Olá, Mundo
17         System.out.print(Olá, mundo!);
18     }
19 }
20
```

Ao abrir aspas, automaticamente o programa já fechará.



```
1  /*
2  * Nome do Pacote, neste caso, primeiroprograma
3  */
4  package primeiroprograma;
5
6  /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         // Comando para escrever Olá, Mundo
17         System.out.print(\"Olá, mundo!\");
18     }
19 }
20
```

Basta digitar Olá, Mundo!

```
15 public static void main(String[] args) {  
16     // Comando para escrever Olá, Mundo  
17     System.out.print("Olá, Mundo!");  
18 }  
19  
20 }
```

Outra vantagem na utilização da IDE; agora basta clicar no botão Executar Projeto para testar. Automaticamente ele irá salvar e exibir a Saída do programa; sendo em verde exibida a mensagem “CONSTRUÍDO COM SUCESSO”.

```
15 public static void main(String[] args) {  
16     // Comando para escrever Olá, Mundo  
17     System.out.print("Olá, Mundo!");  
18 }  
19  
20 }  
21
```

primeiroprograma.PrimeiroPrograma > main >

Saída - PrimeiroPrograma (run) x

run:
Olá, Mundo!CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

Reconhecimento de abreviações de comandos

Abreviando o Comando System.out.println

Visualize novamente o comando “System.out.print(“Olá, Mundo!”);”.

```
15  [ ] public static void main(String[] args) {  
16      // Comando para escrever Olá, Mundo  
17      System.out.print("Olá, Mundo!");  
18  }  
19  
20 }
```

Apague esse comando para que possamos inseri-lo novamente de forma abreviada.

```
15  [ ] public static void main(String[] args) {  
16      // Comando para escrever Olá, Mundo  
17      |  
18  }  
19  
20 }
```

Digite no lugar do comando anterior, que acabamos de apagar, somente: sout.

```
15  [ ] public static void main(String[] args) {  
16      // Comando para escrever Olá, Mundo  
17      sout  
18  }  
19  
20 }
```

Aperte a tecla Tab do seu teclado e o comando completo já será inserido.

Haverá uma diferença do “print”, inserido anteriormente, para “println” que está atualmente em sua tela. A diferença é que com “println” insere o texto e pula o cursor para próxima linha.

```
15  [ ] public static void main(String[] args) {  
16      // Comando para escrever Olá, Mundo  
17      System.out.println(|);  
18  }  
19  
20 }
```

Vamos realizar outro exercício de abreviação.

Abreviando o Comando System.out.println

Desta vez apague o bloco que inicia em “public static void main (String[] args) { ... }”, conforme a imagem.

```
4 package primeiroprograma;
5
6 /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15
16
17
18 }
```

Digite no lugar do bloco, que acabamos de apagar, somente “psvm”.

```
4 package primeiroprograma;
5
6 /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15
16     psvm
17
18 }
```

Aperte a tecla Tab do seu teclado e basta.

```
4 package primeiroprograma;
5
6 /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         |
17     }
18
19
20 }
```

Digite novamente “sout” e aperte a tecla Tab para reconstruirmos o nosso programa.

```
4 package primeiroprograma;
5
6 /**
7  * Nome da Classe, neste caso, PrimeiroPrograma
8  * @author Ricardo
9  */
10 public class PrimeiroPrograma {
11
12     /**
13     * @param args the command line arguments
14     */
15     public static void main(String[] args) {
16         System.out.println("");
17     }
18
19
20 }
```

Digite novamente “Olá, mundo!”.

```
15     public static void main(String[] args) {
16         System.out.println("Olá, Mundo!");
17     }
```

Aperte o botão “Executar Projeto” e note que ele já irá escrever “CONSTRUÍDO COM SUCESSO” na linha de baixo.

