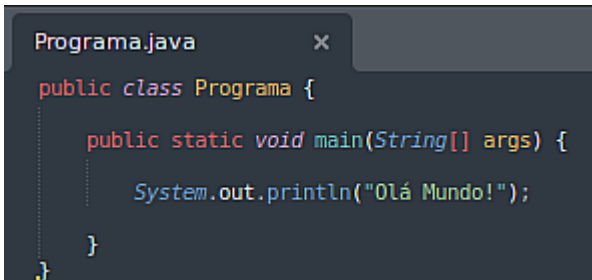


Será descrito o processo básico de criação, compilação e execução de um programa java usando um editor de texto e o terminal (no nosso caso usaremos o sublime e o terminal Linux).

Compilação e Execução de um Programa Java

Passo 1: Escrever o Código Fonte

Abra um editor de texto, como o Sublime, e crie um arquivo chamado "Programa.java" com o seguinte código:

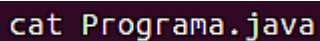


```
Programa.java x
public class Programa {
    public static void main(String[] args) {
        System.out.println("Olá Mundo!");
    }
}
```

OBS: A classe principal deve ter o mesmo nome do arquivo com a extensão ".java" no nosso caso a classe é Programa, logo o arquivo deve ser salvo com nome: Programa.java

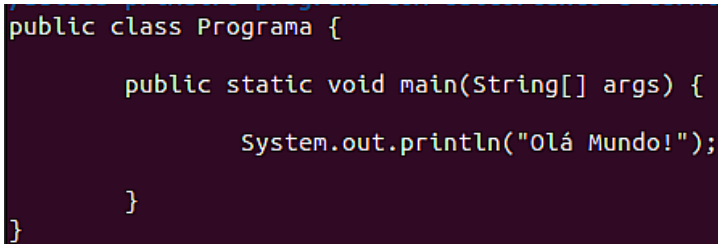
Passo 2: Verificar o Código Fonte

Navegue até o diretório onde o arquivo "Programa.java" está localizado e Use os comandos `cat` ou `less` no terminal para visualizar o conteúdo do arquivo e garantir que o código foi digitado corretamente.



```
cat Programa.java
```

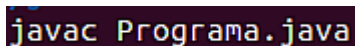
Após usar esse comando vai ser mostrado o conteúdo do arquivo:



```
public class Programa {
    public static void main(String[] args) {
        System.out.println("Olá Mundo!");
    }
}
```

Passo 3: Compilação do Código Fonte

No terminal, navegue até o diretório onde o arquivo "Programa.java" está localizado e execute o seguinte comando para compilar:

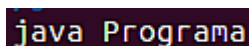


```
javac Programa.java
```

Isso criará um arquivo "Programa.class" que contém o bytecode Java.

Passo 4: Execução do Programa

Ainda no terminal, execute o seguinte comando para executar o programa:



```
java Programa
```

Ao digitar `java Programa` no terminal, está efetivamente chamando a Java Virtual Machine (JVM) para executar o programa Java. O comando `java` é utilizado para iniciar a JVM e executar a classe Java especificada após ele.

Ao fazer isso, a JVM carrega o `bytecode` do arquivo `Programa.class` gerado durante a compilação e executa o método `main` da classe `Programa`. Isso resulta na execução do código contido dentro desse método, que, no nosso caso, imprime "Olá Mundo" no console:

```
Olá Mundo!
```

Observação: Não inclua a extensão `".class"` ao executar o programa.

Possíveis Erros

Durante a compilação e execução, podem ocorrer erros. Aqui estão alguns exemplos:

Erro de Execução:

Certifique-se de **não** incluir `".class"` ao executar o programa, caso contrario, aparecerá esse erro:

```
/aula03-primeiro-programa-com-editortexto-e-terminal$ java Programa.class
Erro: Não foi possível localizar nem carregar a classe principal Programa.class
Causada por: java.lang.ClassNotFoundException: Programa.class
```

Erro de Compilação:

Verifique se todos os pontos e vírgulas estão corretos no código, caso contrario, aparecerá esse erro:

```
/aula03-primeiro-programa-com-editortexto-e-terminal$ javac Programa.java
Programa.java:5: error: ';' expected
        System.out.println("Olá Mundo!")
                        ^
1 error
```

Erro de Palavra-chave:

Certifique-se de que as palavras-chave, como `"public"` estejam em letras minúsculas, caso contrario, aparecerá esse erro:

```
/aula03-primeiro-programa-com-editortexto-e-terminal$ javac Programa.java
Programa.java:3: error: <identifier> expected
    Public static void main(String[] args) {
    ^
1 error
```

Compilação e Execução na JVM:

- Durante a compilação, ocorre verificação sintática.
- A compilação pode gerar erros, corrigíveis pelo desenvolvedor.
- A JVM lê e executa o Bytecode gerado pelo compilador.
- A compilação gera o Bytecode se não houver erros sintáticos.

Lembrando que o uso de uma IDE facilita a identificação de erros e a organização do código. Este procedimento manual é uma abordagem inicial e didática.