Terminal e Argumentos

Nem sempre executamos nosso programa Java pela IDE, já pensou, nossos clientes tendo que instalar o Eclipse ou VsCode para rodar o sistema em sua empresa ?

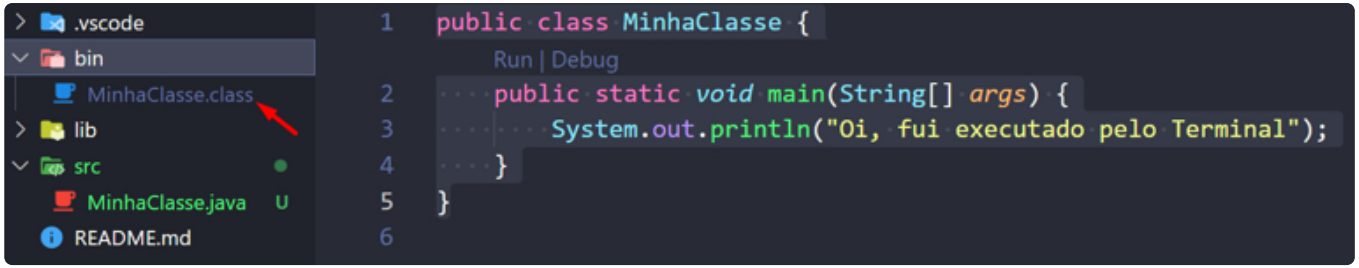
Com a JVM devidamente configurada, nós podemos criar um executável do nosso programa e disponibilizar o instalador para qualquer sistema operacional.

No nosso caso, iremos aprender como executar um programa Java via terminal, como MS - DOS ou terminal do VsCode.

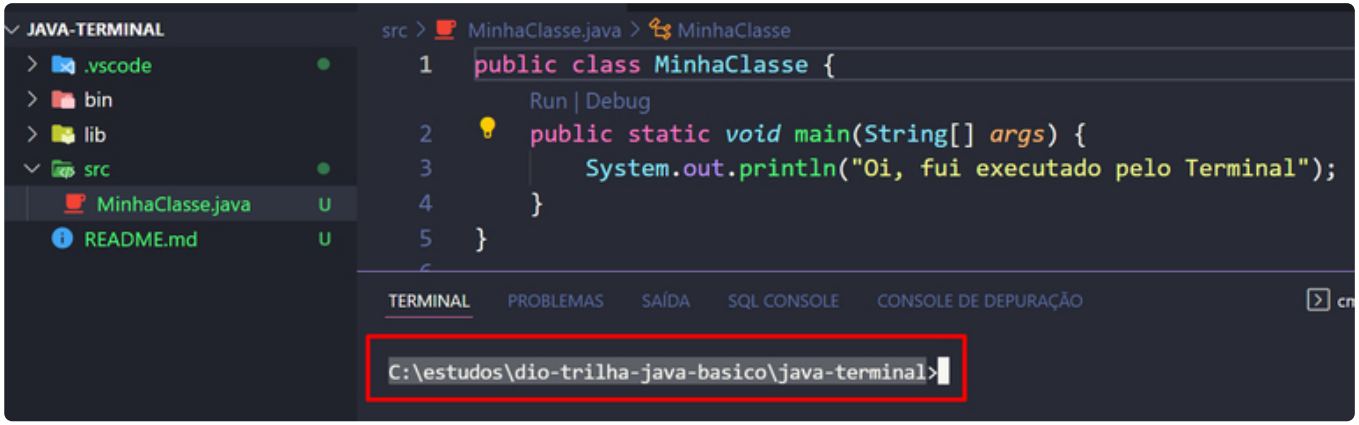
Vamos criar uma classe chamada `MinhaClasse.java` com o código abaixo:

```
public class MinhaClasse {
    public static void main(String[] args) {
        System.out.println("Oi, fui executado pelo Terminal");
    }
}
```

Observe que nosso projeto Java criado por uma IDE, terá uma pasta chamada **bin**. É nesta pasta que ficarão os arquivos **.class**, o nosso `bytecode`.



Mesmo usando uma IDE, nós sempre precisaremos identificar aonde se encontram as classes do nosso projeto, no meu caso está em: **C:\estudos\dio-trilha-java-basico\java-terminal.**

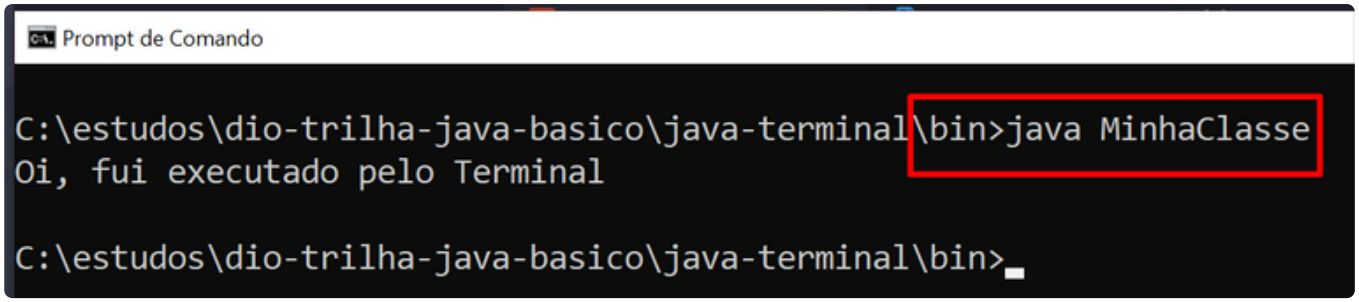


Terminal

Vamos ilustrar como executar uma classe, depois de compilada, sem precisar usar a IDE.

1. Abra o MS-DOS ou Power Shell;
2. Localize o diretório do seu projeto:
   `cd C:\estudos\dio-trilha-java-basico\java-terminal ;`
3. Acesse a pasta **** *bin*: ** `cd bin` ** `;`

4. Agora digite o comando:** `java MinhaClasse` ** *(nome da sua classe sem a extensão **.class**).*



Argumentos

Quando executamos uma classe, que contenha o método main, o mesmo permite que passemos um array `[]` de argumentos, do tipo String. Logo, podemos após a definição da classe a ser executada, informar estes parâmetros, exemplo:

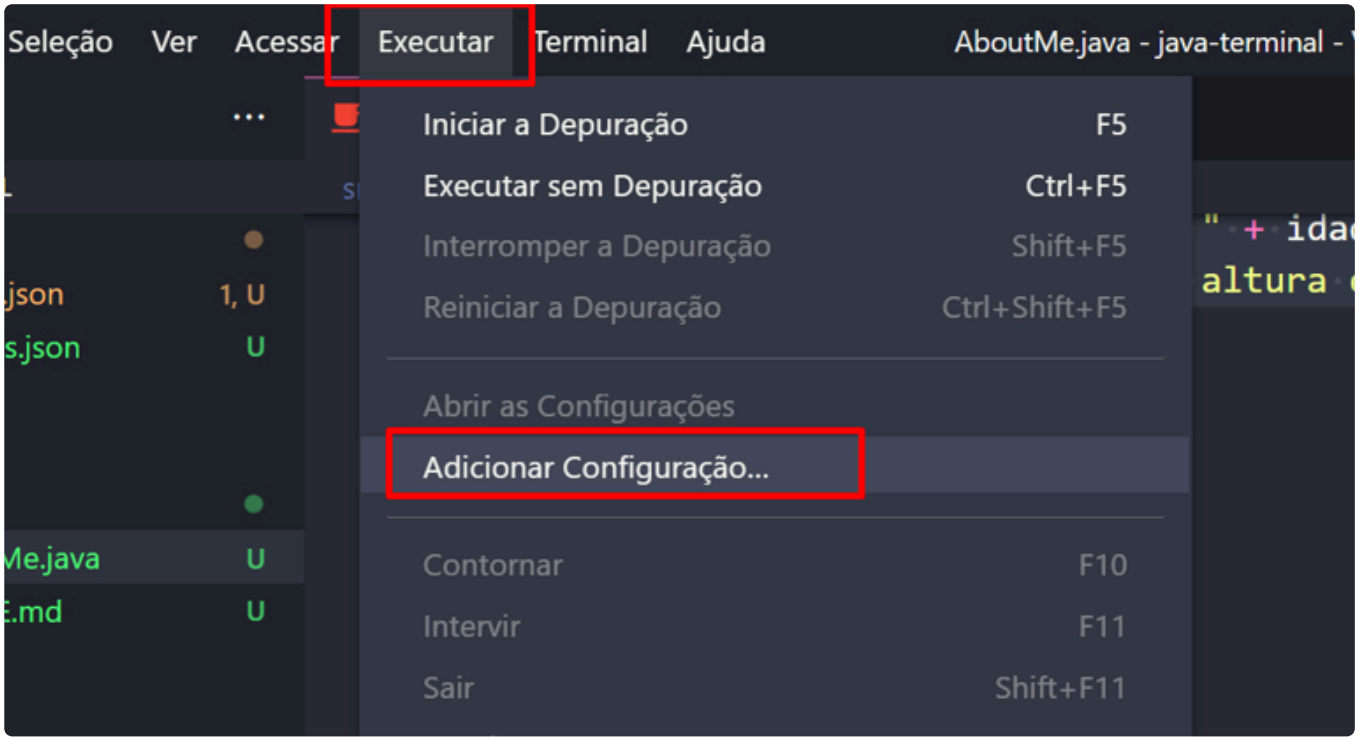
```
java MinhaClasse agumentoUm argumentoDois
```

Exemplo:

```java
public class AboutMe {
    public static void main(String[] args) {
        //os argumentos começam com indice 0
        String nome = args [0];
        String sobrenome = args [1];
        int idade = Integer.valueOf(args[2]); //vamos falar sobre Wrappers
        double altura = Double.valueOf(args[3]);

        System.out.println("Ola, me chamo " + nome + " " + sobrenome);
        System.out.println("Tenho " + idade + " anos ");
        System.out.println("Minha altura é " + altura + "cm ");
    }
}
```

Passando valores aos argumentos pelo VsCode:




```
{
    "version": "0.2.0",
    "configurations": [
        {
            "type": "java",
            "request": "launch",
            "mainClass": "AboutMe",
            "args": ["GLEYSON", "SAMPAIO", "28", "1.58"]
        }
    ]
}
```

Executando o programa agora no terminal:

```
cd C:\estudos\dio-trilha-java-basico\java-terminal
cd bin

java AboutMe GLEYSON SAMPAIO 28 1.58
```

Scanner

Nos exemplos anteriores, percebemos que podemos receber, dados digitados pelo usuário do nosso sistema, porém, tudo precisa estar em uma linha e também é necessário informar os valores nas posições correspondentes. Esta abordagem pode deixar margens de execução, com erro do nosso programa. Para isso, com a finalidade de deixar as nossas entradas de dados mais seguras, agora vamos receber estes dados via **Scanner**.

A classe **Scanner,** permite que o usuário tenha, uma interação mais assertiva com o nosso programa, veja como vamos mudar o nosso programa `AboutMe` para deixar mais intuitivo aos usuários:

```java
import java.util.Locale;
import java.util.Scanner;

public class AboutMe {
    public static void main(String[] args) {
        //criando o objeto scanner
        Scanner scanner = new Scanner(System.in).useLocale(Locale.US);

        System.out.println("Digite seu nome");
        String nome = scanner.next();

        System.out.println("Digite seu sobrenome");
        String sobrenome = scanner.next();

        System.out.println("Digite sua idade");
        int idade = scanner.nextInt();

        System.out.println("Digite sua altura");
        double altura = scanner.nextDouble();


        //imprimindo os dados obtidos pelo usuario
        System.out.println("Ola, me chamo " + nome + " " + sobrenome);
        System.out.println("Tenho " + idade + " anos ");
        System.out.println("Minha altura é " + altura + "cm ");


    }
}
```

https://code.visualstudio.com/docs/java/java-debugging
code.visualstudio.com

Pass parameters when debugging vscode java. [Note]

linuxtut.com

Previous
Documentação

Next
Conceito

Last updated 1 year ago