



EducaCiência FastCode

Fala Galera,

Neste artigo, abordaremos um tema muito interessante.

- Artigo: 11/2020 Data: Março/2020
- Público Alvo: Desenvolvedores – Iniciantes ao Avançado
- Tecnologia: Java
- Tema: Orientação à Objeto – parte 1
- Link: <https://github.com/perucello/DevFP>

Desta vez , escolhi um tema onde amedronta e muito os Desenvolvedores principalmente os iniciantes que é Orientação à Objeto.

Optei em dividir o tema em três partes com intuito de facilitar o entendimento.

Vamos abordar nesse artigo, a “**parte 1**”, para obter mais informações, teremos a continuação no artigo 12 onde traremos a “**parte 2**” e na sequencia do artigo 13 nós traremos a “**parte 3**” e assim espero que facilite o entendimento quanto ao temido Java Orientado à Objeto.

Primeiramente, vamos explorar um pouco o Termo Orientação à Objeto !

"O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados."

"Programação Orientada a Objetos (também conhecida pela sua sigla POO) é um modelo de análise, projeto e programação de software baseado na composição e interação entre diversas unidades chamadas de 'objetos'. A POO é um dos 4 principais paradigmas de programação (as outras são programação imperativa, funcional e lógica). Os objetos são operados com o conceito de 'this', de forma que seus métodos (muitas vezes) modifiquem os dados da própria instância. Os programas são arquitetados através de objetos que interagem entre si. Dentre as várias abordagens da POO, as baseadas em classes são as mais comuns: objetos são instâncias de classes, o que em geral também define o tipo do objeto. Cada classe determina o comportamento (definido nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos..."

https://pt.wikipedia.org/wiki/Orienta%C3%A7%C3%A3o_a_objetos





Bom, feito uma breve explicação, vamos ao que interessa, vamos aos códigos!
Lembro que não trabalharemos com demonstração em “**desenhos**”, vou tentar explicar este conceito de “**parte 1**” puramente em código, que é o mais importante.

Cenário 1:

A ideia é termos um Objeto Chamado IDE e teremos dois atributos sendo “nome” que representa o nome da nossa IDE e outro atributo chamado “Fabricante” que informaremos o fabricante da nossa IDE.

Neste cenário, não abordaremos Banco de Dados e sim script em modo root para facilitar o entendimento.

Manipularemos o OO dentro de um mesmo Package neste cenário.

Descrição:

- a) Teremos nosso Projeto chamado **OrientadoObjetoParte1**
- b) Teremos apenas **um Package** (pacote neste exemplo)
- c) Teremos duas classes, sendo:
 - ⇒ **A Classe que teremos nosso método Main**
 - **MetodoMain.java** => Neste momento instanciaremos nosso Método Main com o Objeto e populando suas características/atributos.
 - ⇒ **A Classe que criará nosso Objeto**
 - **ObjetoIDE.java** => nesta Classe, criamos nosso objeto e seus atributos/características, também criamos um Método void onde adicionaremos o nosso SOUT sujo qual retornará na tela via Método Main.

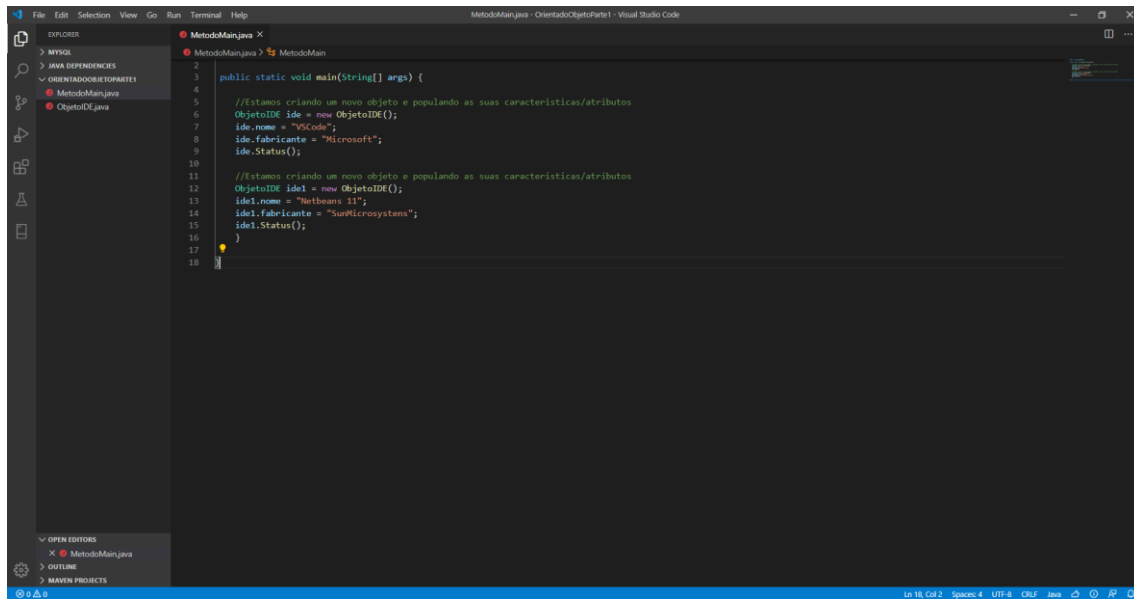
Após criarmos nossas duas classes como acima definidas, vamos aos códigos!

ObjetoIDE.java

```
ObjetoIDE.java - OrientadoObjetoParte1 - Visual Studio Code
1 //criado nossa Classe - Objeto - no caso chamado ObjetoIDE
2 public class ObjetoIDE{
3
4     String nome; //criado nosso atributo ou característica do Objeto - neste caso chamado NOME
5     String fabricante; //criado nosso atributo ou característica do Objeto - neste caso chamado FABRICANTE
6
7     //Criando nosso método Status onde manipularemos um SOUT para trazer de retorno na tela - instanciado na classe MetodoMain.java
8     void Status(){
9         //Neste momento vamos imprimir na tela o NOME e FABRICANTE que passaremos por argumento, neste caso referenciando pelo THIS
10        System.out.println("Você escolheu a IDE " + this.nome + " para trabalhar no desenvolvimento JAVA !");
11        System.out.println("O Fabricante : " + this.fabricante);
12    }
13 }
```



MetodoMain.java



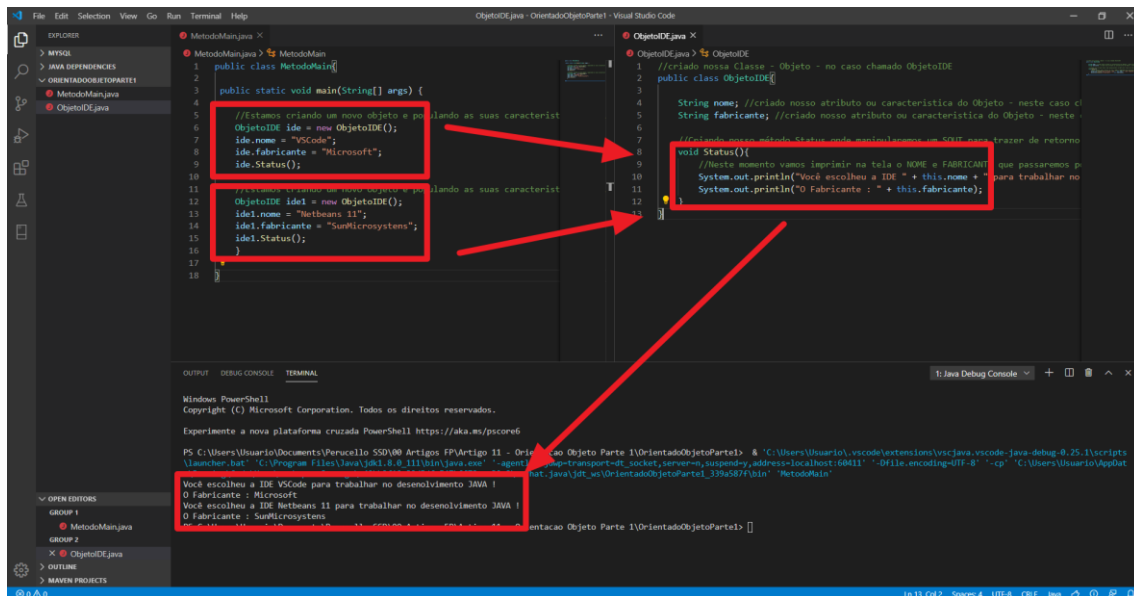
```

1 public static void main(String[] args) {
2
3     //Estamos criando um novo objeto e populando as suas caracteristicas/atributos
4     Ide ide = new Ide();
5     ide.nome = "VSCode";
6     ide.fabricante = "Microsoft";
7     ide.Status();
8
9
10    //Estamos criando um novo objeto e populando as suas caracteristicas/atributos
11    Ide1 ide1 = new Ide1();
12    ide1.nome = "Netbeans 11";
13    ide1.fabricante = "Sun Microsystems";
14    ide1.Status();
15
16 }
17
18

```

Após concluirmos nosso Desenvolvimento, teremos nosso Objeto IDE que criamos , com seus atributos devidamente populados e enfim colocaremos nosso código em execução.

De acordo com a proposta, deverá trazer em nossa tela, os dois objetos criados e populados IDE e IDE1 que criamos via código.



```

1 //criado nossa Classe - Objeto - no caso chamado Ide
2 public class Ide {
3
4     String nome; //criado nosso atributo no caracteristica do Objeto - neste caso o
5     String fabricante; //criado nosso atributo no caracteristica do Objeto - neste
6
7     //criado nosso atributo Status e populamos um Status para trazer de retorno
8     void Status() {
9         //Neste momento vamos imprimir na tela o NOME e FABRICANTE que passaremos p
10        System.out.println("Você escolheu a IDE : " + this.nome + " para trabalhar no
11        System.out.println("O Fabricante : " + this.fabricante);
12    }
13 }

```

```

PS C:\Users\Usuario\Documents\Perucello SSD\Artigos FP\Artigo 11 - Orientado a Objeto Parte 1> .\launcher.bat
Experimente a nova plataforma cruzada PowerShell https://aka.ms/powershell
Voc  escolheu a IDE VSCode para trabalhar no desenvolvimento 38VA
O Fabricante : Microsoft
Voc  escolheu a IDE Netbeans 11 para trabalhar no desenvolvimento 38VA
O Fabricante : Sun Microsystems

```

At  mais !

Espero ter ajudado !