

FuturoDEV

Exercícios da Semana

COMO REALIZAR OS EXERCÍCIOS

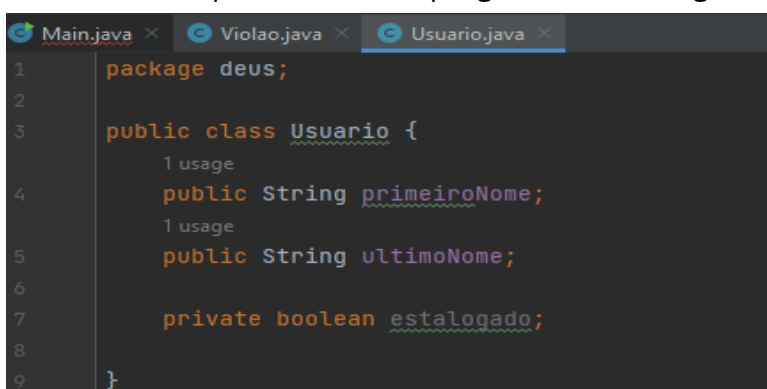
Recomenda-se criar uma pasta no computador para armazenar as respostas dos exercícios semanais. No primeiro módulo do curso, os alunos irão receber os enunciados através de um documento texto, após (segundo módulo), os exercícios serão disponibilizados semanalmente em um quadro do *Trello*, onde os alunos irão movimentar os cartões de exercícios/tarefas, seguindo a metodologia *Kanban*, e armazenar as respostas em um repositório no *GitHub*.

[M1S02] Ex 1 - Imprima propriedades do objeto

Imagine um sistema que vende acessórios de computador, então suponha que você tenha uma série de **objetos**, que você quer trazer do mundo real para o mundo computacional. Que tipo de **propriedades e comportamentos** você consegue imaginar a respeito desses objetos? Implemente em **Java** uma **classe** com um desses objetos e **instancie** esse **objeto** na classe **Main**, após isso **imprima na tela** os dados referentes a esse objeto criado.

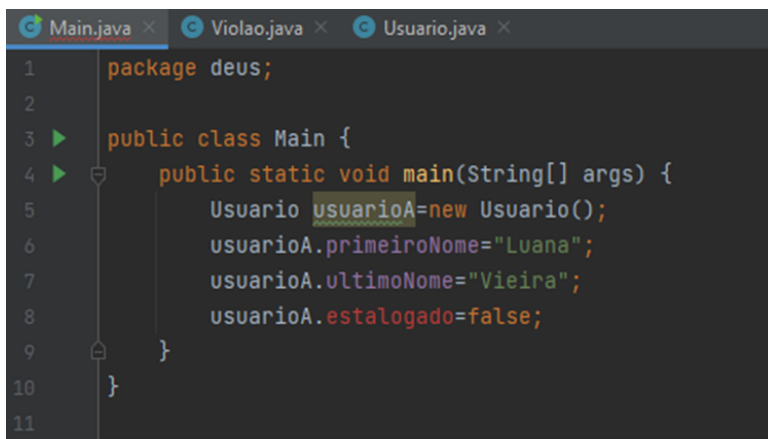
[M1S02] Ex 2 - Visibilidade privada

Considerando que eu tenha os programas das imagens 1 e 2 abaixo:



```
1 package deus;
2
3 public class Usuario {
4     1 usage
5     public String primeiroNome;
6     1 usage
7     public String ultimoNome;
8
9     private boolean estalogado;
10 }
11 }
```

Imagem 1. Classe Usuário



```

1 package deus;
2
3 public class Main {
4     public static void main(String[] args) {
5         Usuario usuarioA=new Usuario();
6         usuarioA.primeiroNome="Luana";
7         usuarioA.ultimoNome="Vieira";
8         usuarioA.estalogado=false;
9     }
10 }
11

```

Imagem 2. Classe Main

A propriedade “estalogada” foi definida como **privada** na classe Usuario, quais são as consequências ao tentar realizar o acesso dessa propriedade na Classe Main?

[M1S02] Ex 3 - Método Construtor

Através do método construtor é possível criar o **objeto** em **memória**. Com base nisso, defina uma classe chamada Funcionario, com ao menos 2 **atributos** e crie o **método construtor** padrão para essa mesma **classe**.

[M1S02] Ex 4 - Sobrecarga de Métodos

O conceito de **sobrecarga de métodos** flexibiliza o código, permitindo que o **objeto** seja **instanciado** de 2 ou mais formas. Sendo assim, crie 3 métodos usando sobrecarga de métodos com base em uma **classe ImpostoRenda** e que execute os seguintes cálculos:

1. Método para calcular o imposto de renda para quem tem uma renda entre R\$6.677,56 a R\$9.922,28.
2. Método para calcular o imposto de renda para quem tem uma renda entre R\$9.922,29 a R\$13.167,00.
3. Método para calcular o imposto de renda para quem tem uma renda entre R\$13.167,01 a R\$16.380,38.