

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ Curso: ADS Disciplina: Programação Orientada a Objetos Professor: Ely</p>
--	--

Exercício 03 - extra

1. Qual o resultado da execução da classe abaixo?

```
public class Teste {
    public static void executa(String[] args) {
        System.out.println("ola");
    }
}
```

2. Considerando a classe Dummy abaixo e uma classe de testes em nela seja apenas instanciado um objeto Dummy e seu metodo3 seja invocado, qual a saída em tela após isso? Justifique.

```
public class Dummy {
    int atributo1;
    int atributo2;

    int metodo1() { return ++atributo1; }
    int metodo2() { return ++atributo2; }

    void metodo3() {
        if (atributo1 == 0 && (metodo1() == 0) && metodo2() == 0) {
            System.out.print(atributo1 + atributo2);
        } else {
            System.out.print(2*atributo1 - atributo2);
        }
    }
}
```

3. Qual o resultado da compilação do código abaixo?

```
public class Aplicacao {
    public static void main(String args[]) {
        int i = 1, j;
        if (j != 0) {
            j = 1;
            i = i + j;
        }
        System.out.println(i);
    }
}
```

4. Dadas as classes abaixo:

```
public class NumberUtils {
```

```
public class TestaAluno {
```

<pre> int i; int dobra() { return i * 2; } </pre>	<pre> public static void main(String[] args){ NumberUtils n = new NumberUtils(); System.out.println(n.dobra()); } </pre>
--	--

Podemos afirmar que ocorrerá um erro de compilação relacionado ao atributo **i** não inicializado? Justifique.

5. Considere as duas classes abaixo:

<pre> public class Post { int countLikes; Post(int initialLikes) { countLikes = initialLikes; } } </pre>	<pre> public class TestPost { public static void main(String[] args){ Post p = new Post(); p.countLikes = 1; } } </pre>
--	---

Explique o que acontece com a compilação da classe TestPost e, se houver problemas, proponha uma solução.

- Implemente um método para a classe acima chamado addLike() que incrementa o atributo countLikes.
- Dado o seguinte trecho de código de acumulação de um atributo **quantidade**, explique a problemática envolvida e reescreva o método com uma possível solução:

```

public class Produto {
    int quantidade;
    void acumula (int quantidade) {
        quantidade++;
    }
}

```

- Considerando a classe abaixo, implemente um construtor que inicialize todos os atributos e o método clonado() para que ele retorne verdadeiro caso um outro objeto passado como parâmetro para o método possua o mesmo chassi e placa do objeto atual.

```

public class Carro {
    String placa;
    long chassi;
    String modelo;
    String ano;
    boolean clonado(Carro c) {...}
}

```

- Implemente uma classe **ImpostoDeRenda** que:
 - Tenha um atributo real chamado **rendaBruta** que represente a renda bruta anual de uma pessoa e crie um construtor que inicialize esse atributo com um valor passado como parâmetro;
 - Crie um método chamado **calcularImposto()** que retorna o valor do imposto a ser pago segundo as regras:

Renda em R\$	% de Imposto
0 a 1000,00	5%
Acima de 1000,00	10%

Ex: renda bruta = 2000,00 imposto (10% da renda bruta) = 200,00

- c. Crie um método chamado **calculaRendaLiquida()** que retorna o valor do líquido da renda: renda bruta – imposto.

10. Uma classe chamada Jogador e nela:

- Crie um atributo inteiro chamado vidas, representando a quantidade de vidas de um jogador;
- Crie um construtor em que é inicializado o atributo vida com 5 vidas;
- Crie um método isLimiteMaxVidas que retorne true se o número de vidas for igual a 5. Falso caso contrário;
- Crie um método isLimiteMinVidas que retorne true se o número de vidas for igual a 0. Falso caso contrário;
- Crie dois métodos que retornam void: um que incrementa a quantidade de vidas e outro que decrementa tal quantidade;
- Altere o os métodos das alternativas d. e e. para que não incremente caso o limite máximo seja atingido e não decremente as vidas caso o limite mínimo tenha atingido. Use os métodos das alternativas c) e d)
- Crie um método doar vida, onde um jogador recebe como parâmetro outro jogador e uma vida sua é decrementada e no outro jogador a quantidade de vidas é incrementada;
- Altere o método da classe anterior para não realizar a operação caso o doador tenha atingido o limite mínimo e o receptor tenha atingido o limite máximo.