

 <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>
--	--

Exercícios

1. Corrija as seguintes afirmações:

Considere uma classe chamada Avaliacao. Para compilarmos e executarmos a classe devemos executar os seguintes comandos:

```
javac Avaliacao
javac Avaliacao.java
```

2. Qual o resultado da compilação do código abaixo e justifique sua resposta:

```
public class Aplicacao {
    public static void main {
        int contador;
        System.out.println(contador++);
    }
}
```

3. Explique detalhadamente o resultado após a execução abaixo do código:

```
public class Exemplo {
    public static void main(String[] args) {
        int a = 9;
        int b = 0;
        if ((a > 11) && (++b <= 100)){
            System.out.println(a*b);
        } else {
            System.out.println(a-b);
        }
    }
}
```

4. Explique detalhadamente qual a saída da execução do trecho de código abaixo e justifique:

```
int op = 1;
switch (op) {
    case 1: System.out.println(op);
    case 2: System.out.println(op);
    default: System.out.println(op);
}
```

5. Corrija a classe abaixo para que a mesma exiba corretamente a soma de a e b.

```
public class Exemplo {
    public static void main(String args[]) {
```

```

        int a = 10;
        int b = 1;
        System.out.println("Soma: " + a + b);
    }
}

```

6. Crie uma classe em que o método **main()** exiba uma mensagem de saudação usando um "println".
7. Crie uma classe chamada **Saudacao** que contenha um atributo chamado **texto** e outro chamado **destinatario**, ambos String. Crie um método **obterSaudacao()** que retorne a concatenação dos dois atributos. Ex. de saída: "Bom dia, João".
8. Crie uma classe **TestaSaudacao** para testar a classe **Saudacao**. Nela crie um método **main()** que instancie um objeto do tipo **Saudacao** e inicialize os dois atributos. Após isso, exiba (println) o resultado do **método obterSaudacao()**.
9. Compare as implementações da questão 6 e com a implementação das questões 7 e 8. Qual é a abordagem mais "estruturada" e qual a versão mais Orientada a Objetos? Qual a mais modularizada e reutilizável? Entretanto, qual a mais burocrática devido ao problema a ser resolvido?

A partir da questão seguinte, crie além da classe pedida uma classe para testes e execução da classe em questão. Nessa classe, inicialize os atributos e teste todos os métodos de cada classe. Ex: Para a classe "X.java", crie uma classe chamada "TesteX.java"

10. Crie uma classe **Soma** que tenha dois atributos numéricos e que tenha um método **calculaSoma()**, que retorna a soma dos dois atributos.
11. Crie uma classe chamada **DecimalNumber** que tenha um atributo double. Esta classe deve ter dois métodos: um que exiba a parte inteira do atributo e outro que retorne parte decimal. Use cast.
12. Crie uma classe chamada **Desconto** que tenha como atributos dois valores referentes a um valor original de uma conta e um desconto em porcentagem. Crie um método **calcula()** que retorne um valor calculando o desconto da seguinte forma: $\text{valorCalculado} = \text{valorOriginal} * (1 - \text{desconto}/100)$.
Ex: valor: 120 desconto: 10. Valor calculado: 108
13. Crie uma classe chamada **NumberUtils** que tenha um atributo numérico **n** os seguintes métodos:
 - a. **isPair()**: retorna verdadeiro se o número for par. Aproveitando a implementação desse método, crie outro método "isOdd()" que retorne false se o número for ímpar;
 - b. **isPrime()** que retorna true ou false se o número é primo ou não.;
 - c. **printCount()** que retorne a contagem crescente em forma de String de zero até o valor do atributo. Ex: "0, 1, 2, 3, 4";
 - d. **printReverseCount()** que retorna uma String com contagem decrescente;
14. Crie uma classe Equipamento com:
 - a. um atributo ligado (tipo boolean)

- b. dois métodos `liga()` e `desliga()`. O método `liga` torna o atributo `ligado` `true` e o método `desliga` torna o atributo `ligado` `false`.
- c. Crie um método chamado `inverte()`, que muda o status atual (se `ligado`, `desliga`...se `desligado`, `liga`)
- d. Crie um método `estaLigado()` que retorna o valor do atributo `ligado`
- e. Altere o comportamento dos métodos `liga` para caso o equipamento já esteja `ligado`, não ligue novamente. Faça o mesmo com o método `desligar`.

15. Crie uma classe chamada `TestaEquipamentos` e em seu método `main`:

- a. Instancie 2 objetos da classe `Equipamento`;
- b. Ligue o primeiro e desligue o segundo;
- c. Chame o método `inverte()` em cada um dos equipamentos;
- d. Faça um `if` para cada um dos equipamentos testando se o mesmo está `ligado`. Imprima: `"ligado"` ou `"desligado"` dependendo da situação de ambos;