

Lista 1

Nota: Deve utilizar todos os pilares da POO aprendidos até agora e deve criar um **main** para testar todas as implantações.

1. Implemente uma classe que modele um aluno de ginásio:
 - Atributos: Nome, data de nascimento, peso e altura
 - Métodos: Calcula Idade
2. Faça um programa que simule um televisor criando-o como um objecto:
 - O utilizador deve ser capaz de informar o número do canal e aumentar ou diminuir o volume.
 - Certifique-se de que o número do canal e o nível do volume permanecem dentro de faixas válidas
3. Crie uma classe que modele um funcionário
 - Um empregado tem um nome e um salário
 - Escreva um constructor com dois parâmetros (nome e salário)
 - Métodos para devolver nome, salário, aumentar salário (percentual de aumento) que aumente o salário do funcionário em uma certa percentagem.
 - Escreva um pequeno programa que teste a sua classe.
4. Implementar uma classe Caneta que deve possuir como características marca, cor e tamanho. Nesta classe devem ser implementados os métodos construtores, getters, setters. Em uma outra classe chamada CanetaTeste deverá ser criado um objeto do tipo Caneta, atribuir valores e exibir os dados deste objeto.
5. Implemente uma classe para definir os objectos que representarão os **clientes** de um banco. Essa classe deve declarar dois atributos: um para os nomes e outro para os códigos dos clientes. No final efectue um teste criando dois objectos da classe cliente e imprima seus dados.
6. Como sabemos qualquer empresa possui funcionários, crie uma classe chamada Funcionario para representa-los. Adicione os atributos nome e salário a essa classe, além disso crie dois métodos um para aumentar o salário e outro para consultar os dados dos funcionários.

7. Implementar uma classe **Lampada** que deve possuir como características tipo (led, fluorescente,), voltagem, cor, marca, preço, potência, status(boolean). Nesta classe devem ser implementados os métodos construtores, getters, setters, acender e apagar. Em uma outra classe chamada **LampadaTeste** devem ser criados dois objetos do tipo **Lampada**, atribuir valores e exibir os dados deste objeto. O programa deverá informar também qual das duas lâmpadas possui maior potência e também qual das lâmpadas é a mais cara.
8. Crie uma classe **Pessoa** que obedeça a descrição apresentada na representação abaixo.
- Possua os atributos **nome** e **idade**. E os métodos: **dizerONome**, **dizerAldade** e **fazerAniversario**.
 - O método **dizerONome** deve escrever na tela a mensagem: “Olá, meu nome é”, e o valor do atributo **nome**.
 - O método **dizerAldade** deve escrever na tela a mensagem: “Olá, eu tenho X anos”, onde X é o valor do atributo **idade**.
 - O método **fazerAniversario** deve adicionar 1 à idade da Pessoa
 - Crie também um método **main** que realize as seguintes operações:
 - Instancie um objeto do tipo Pessoa.
 - Atribua o seu nome ao atributo **nome**.
 - Atribua a sua idade ao atributo **idade**.
 - Chame o método **dizerONome**
 - Chame o método **dizerAldade**
 - Chame o método **fazerAniversario**
 - Chame o método **dizerAldade**
9. Implemente uma classe **Porta** com os seguintes atributos: aberta, cor, dimX, dimY, dimZ e com os seguintes métodos void open(), void close(), void paint(String cor), boolean isOpen(). Crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere as suas dimensões e use o método **isOpen()** para verificar se ela está aberta.
10. Implemente uma classe **Retângulo** que obedeça à descrição abaixo

Retangulo
+ lado1: float + lado2: float + área: float + perimetro: float
+ calcularArea(): void + calcularPerimetro(): void

Crie o método main e instancie a classe Retangulo, criando um objeto novoRetangulo do tipo Retangulo.

- Atribua o valor 10 ao atributo lado1
- Atribua o valor 5 ao atributo lado2
- Chame o método calcularArea.
- Chame o método calcularPerimetro.
- Atribua o valor 7 ao atributo lado2
- Chame o método calcularArea.
- Chame o método calcularPerimetro.

11. Crie uma classe **Circulo** que obedeça à descrição abaixo:

Circulo
+ raio: float + area: float + perimetro: float
+ calcularArea(): void + calcularPerimetro(): void

O perímetro é dado pela forma (**perimetro = 2 * 3.14 * raio**).

Crie o método main e instancie a classe Círculo, criando um objeto novoCirculo do tipo Circulo.

- Atribua o valor 10 ao atributo raio
- Chame o método calcularArea.
- Escreva no ecrã o valor da área.
- Chame o método calcularPerimetro.
- Escreva no ecrã o valor do perímetro.
- Atribua o valor 4 ao atributo raio
- Chame o método calcularArea.
- Escreva no ecrã o valor da área.
- Chame o método calcularPerimetro.
- Escreva no ecrã o valor do perímetro.

12. Crie uma classe **Funcionário** que obedeça à descrição abaixo:

Funcionario
+ nome: String + sobrenome: String + horasTrabalhadas: int + valorPorHora: float
+ nomeCompleto(): string + calcularSalario(): void + incrementarHoras(int): void

- O método `nomeCompleto` deve retornar o atributo `nome` concatenado ao atributo `sobrenome`
- O método `calcularSalario` faz o cálculo de quanto o funcionário irá receber no mês, multiplicando o atributo `horasTrabalhadas` pelo atributo `valorPorHora`. Em seguida, escreve o valor no ecrã.
- O método `incrementarHoras` adiciona um valor passado por parâmetro ao valor já existente no atributo `valorPorHora`.

Crie o método `main` e instancie a classe `Funcionário` criada, criando um objeto `novoFuncionario` do tipo `Funcionario`.

- Atribua o valor "Lionel" ao atributo `nome`
- Atribua o valor "Messi" ao atributo `sobrenome`
- Atribua o valor 10 ao atributo `horasTrabalhadas`
- Atribua o valor 25.50 ao atributo `valorPorHora`
- Chame o método `nomeCompleto`.
- Chame o método `calcularSalario`.
- Adicione 8 ao atributo `horasTrabalhadas` utilizando o método `incrementarHoras`
- Chame novamente o método `calcularSalario`.

13. Implemente a classe **Livros** em Java, que obedeça à descrição abaixo

Livro
- titulo: String - qtdPaginas: int - paginasLidas: int
+ getTitulo() : String + setTitulo(String) : void + getQtdPaginas() : int + setQtdPaginas(int) : void + getPaginasLidas() : int + setPaginasLidas(int) : void + verificarProgresso() : void

- O método **verificarProgresso** deverá calcular a percentagem de leitura do livro até o momento através da fórmula: **percentagem = paginasLidas * 100 / qtdPaginas**
O valor da percentagem deverá ser mostrado no ecrã conforme a mensagem "Você já leu X por cento do livro", onde o valor de **X** é o valor calculado pela fórmula apresentada anteriormente.

Crie uma classe **TestarLivros**, essa classe terá apenas o método **main** que servirá para testar a classe `Livros`. As seguintes ações devem ser realizadas:

- Crie um objeto `livrofavorito` do tipo `Livro`;
- Altere o atributo **titulo** para "Game of Thrones"
- Altere o atributo **qtdPaginas** para 500;

- Escreva no ecrã a mensagem: “O livro X possui Y páginas”, onde no lugar de X deverá aparecer o valor do atributo **titulo** e, no lugar de Y deverá aparecer o valor do atributo **qtdPaginas**.
- Altere a quantidade de paginasLidas para 50;
- Chame o método **verificarProgresso**.
- Altere a quantidade de paginasLidas para 250;
- Chame o método **verificarProgresso**.
- Implemente um método construtor na classe que vai os 3 atributos da classe como parâmetro
- Crie um livro ao seu critério utilizando o construtor.

14. O aluno deve criar um programa utilizando a linguagem Java, onde implementa uma lista de contactos. Para isso sugere-se que o aluno crie três classes:

- Exercicio01 - classe que contem o método main que instancia uma nova agenda, e exemplifica a adição de contactos e a sua impressão na consola.
- Agenda - classe que permite instanciar objectos que representam uma agenda, que são constituídos por uma lista de contactos.
- Contacto - classe que permite instanciar objectos que representam um contacto, contendo informação sobre o nome e o número de telefone desse contacto.

Apesar do código pedido neste exercício ser livre, desde que funcional, sugere-se que o código da classe Exercicio01 seja o seguinte:

```
public class Exercicio01 {

    public static void main (String [ ] args ) {

        Agenda agenda = new Agenda();
        agenda.addContacto("Eliane", 932345678);
        agenda.addContacto("Laura", 919876543);
        agenda.addContacto ("Adriel", 963452345);
        agenda.imprime();
    }
}
```

O código acima permite criar uma nova agenda e adicionar informação sobre novos contactos a mesma, terminando a execução do programa com a impressão da agenda, e de todos os seus contactos.

O output esperado é, portanto:

Agenda:
Nome: Eliane - Telefone: 932345678
Nome: Laura - Telefone: 919876543
Nome: Adriel - Telefone: 963452345
Fim da Lista de Contactos

Nota:

Lembre-se de seguir a convenção java, isso é importante. Isto é, **nomeDeAtributo**, **nomeDeMetodo**, **nomeDeVariavel**, **NomeDeClasse**, etc...

É também boa prática criar um ficheiro **.java** para cada classe