

Exercícios:

- 1) Crie um projeto chamado projeto-aula02.
- 2) No projeto acima, crie a classe Caneta com os atributos cor, marca, tamanho e tampa e uma classe de teste para instanciar (criar) ao menos 2 objetos da classe Caneta, atribuir valores a seus atributos e imprimir as informações em tela.

```
public class Caneta {
    String cor;
    String marca;
    int tamanho;
    boolean tampa;
}

public class TestaCanetas {
    public static void main(String[] args) {
        Caneta umaCaneta = new Caneta();
        umaCaneta.cor = "Azul";
        umaCaneta.marca = "Pilot";
        umaCaneta.tamanho = 10;
        umaCaneta.tampa = true;

        Caneta outraCaneta = new Caneta();
        outraCaneta.cor = "Preta";
        outraCaneta.marca = "Pilot";
        outraCaneta.tamanho = 10;

        System.out.println("###INFORMAÇÕES DE umaCaneta");
        System.out.println("Esta é uma caneta de cor "+umaCaneta.cor
            +" e da marca "+umaCaneta.marca);

        System.out.println("###INFORMAÇÕES DE outraCaneta");
        System.out.println("Esta já é uma outra caneta de cor "
            +"outraCaneta.cor
            +" e da marca "+outraCaneta.marca);
    }
}
```

Exercícios (parte 2):

- 1) No mesmo projeto do exercício acima, crie a classe Conta com os atributos numero, titular e saldo, além dos métodos saca e deposita. Lembre-se que cada conta já deve ser criada com 500,00 de saldo. Crie uma classe chamada TestaContas que contenha um método main. Crie pelo menos duas contas em memória faça depósitos, saques e imprima as informações das contas. Não esqueça de simular situações de saque inválido e depósito inválido.
- 2) Não se esqueça de respeitar os princípios da alta coesão e o princípio da responsabilidade única.
- 3) Crie um método para transferir dinheiro de uma conta para outra. Utilize esse método na classe TestaContas. Melhore a legibilidade deste método.
- 4) Crie um método mostraInformacoes cujo objetivo é exibir os dados de uma Conta. Utilize esse método na classe TestaContas.

Exercícios (parte 3):

- 1) Nova regra. Nossas contas agora precisam ter nome, cpf e e-mail do titular. Esses atributos são da conta ou do cliente que é titular da conta? Providencie as mudanças necessárias.

- 2) Depois de fazer as devidas alterações, modifique a classe de testes para que tudo continue funcionando. Teve problemas com NullPointerException? Garanta que uma Conta jamais será criada sem um objeto do tipo Cliente instanciado em memória e apontado pelo atributo titular de Conta.
- 3) O objetivo dos métodos é garantir a integridade dos dados. De acordo com as regras do nosso banco ou de qualquer banco da vida real só deve ser possível alterar o saldo de uma conta por meio de operações bancárias. Logo, o código abaixo não pode ser permitido. Cuide para que esse código não possa ser executado.

```
conta1.saldo = -5000;
```

- 4) Partindo da mesma premissa, cuide para que todos os atributos de Conta e Cliente só possam ser acessados de forma indireta e controlada. Se estiver difícil, peça a ajuda do professor.