

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

### Exercício 05

1. Considere o trecho de código abaixo e descreva o que acontece com a execução do código abaixo:

```
import java.util.Scanner;
package br.ifpi.poo.ex05.cadastro;

public class ExecutaBanco {
    ....
}
```

O que há de errado com as declarações e importações de pacotes?

2. O código abaixo representa um possível programa em java que utiliza a classe Banco para realizar cadastro de contas usando a entrada e saída padrões.

```
import java.util.Scanner;
public class ExecutaBanco {
    public static void main(String[] args) {
        Banco b = new Banco();
        Scanner sc = new Scanner(System.in);
        String opcao = "";
        do {
            System.out.println("1-Cadastrar 2-Alterar 3-Excluir ...
9-Sair");
            opcao = sc.next();
            switch (opcao) {
                "1" : Conta c = new Conta();
                        System.out.println("Digite o número");
                        c.numero = sc.next ();

                        c.saldo = sc.nextDouble();
                        b.inserir(c);
                        break;
                "2" : //...
            }
        } while (!opcao.equals("9") );
    }
}
```

Complemente esta implementação usando as classes do exercício 04 e crie opções para todas os métodos da classe banco: inserir, alterar, excluir, consultar, creditar, debitar e transferir. Forneça, sempre que possível uma resposta ao usuário em caso de sucesso ou falha da operação.

3. Crie uma classe Calculadora que tenha dois atributos privados (operando1 e 2) e:
  - a. Crie 2 métodos públicos, cada um representando uma operação básica;
  - b. Crie um construtor onde são passados os operandos;
  - c. Crie uma classe que teste a classe calculadora. Tente acessar os atributos diretamente e verifique o que acontece.

Para os exercícios seguintes, caso considere necessário, crie uma classe de testes.

4. Crie uma classe chamada ContaImposto com um atributo saldo e número.
  - a. Modifique os atributos para privado;
  - b. Crie um método público chamado getSaldo( ) que retorna o saldo atual;
  - c. Crie um construtor onde são passados os 2 parâmetros para setar os atributos;
  - d. Implemente os métodos de crédito e débito como públicos;
  - e. Crie um método privado chamado retemImposto(double valorDebito) que seja chamado toda vez que houver um débito e retire 0,38% do valor do débito;
5. Crie uma classe Hora que:
  - a. Tenha 3 atributos privados e definidos no construtor chamados hora, minutos e segundos;
  - b. Crie métodos para ler hora, minuto e segundo de forma individual;
  - c. Crie um método público para imprimir a hora no formato hh:mm:ss.
6. Altere as implementações da classe Banco das aulas anteriores para que:
  - a. O seu atributo índice e contas tenham visibilidade privada;
  - b. O método consulta por índice seja privado;
  - c. Os demais métodos sejam públicos.
7. Altere também a sua classe Conta dos exercícios anteriores para:
  - a. Ter atributos privados e métodos públicos;
  - b. Verifique se sua implementação da classe Banco e da aplicação principal precisarão ser adaptadas para ter métodos de leitura e escrita, visto que os atributos que agora são privados.