INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ	**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ** **Curso: ADS** **Disciplina: Programação Orientada a Objetos** **Professor: Ely**

Exercício 05

1. Atualize a implementação da classe Banco com os métodos apresentados em sala de aula:
    a. Consultar por índice, excluir, atualizar, sacar, depositar e transferir apresentadas em sala. Nota: o transferir deve ter como parâmetros os dois números de conta, não os objetos e um valor a ser transferido conforme slide **"Esqueleto" de um cadastro**.
    b. Crie um método transferir que recebe um array de contas destino e realiza transferências para cada uma delas;
    c. Crie 3 métodos: um que retorne a quantidade de contas, outro que retorne o total de dinheiro depositado em todas as contas. Por fim, crie um método que retorne a média do saldo das contas chamando os dois métodos anteriores.

2. Crie uma implementação que simule um migroblog:
    a. Crie uma classe Postagem e nela:
        a. Crie os atributos:
            1. id do tipo number, representando o identificador da postagem;
            2. texto do tipo string, representando um texto da postagem;
            3. quantidadeCurtidas do tipo number;
        b. Crie um método chamado curtir que incrementa a quantidade curtidas;
        c. Crie um método chamado toString que retorna a concatenação da postagem com a quantidade de curtidas;
    b. Crie uma classe Microblog e nela:
        a. Crie um array de classes Postagem;
        b. Crie um método que inclua uma postagem passada como parâmetro no array de postagens;
        c. Crie um método de excluir uma postagem que recebe um id passado por parâmetro. Para isso, efetue uma busca pelo id nas postagens do array e faça a exclusão de forma análoga à feita na classe Banco;
        d. Crie um método que retorna a postagem mais curtida;
        e. Crie um método curtir em que se passa um id como parâmetro e a classe microblog pesquisa a postagem e chama seu método curtir da própria postagem;

f. Crie um método toString que retorna a concatenação do “toString” de todas as postagens.

3. Coloque as classes de Banco e Conta em um arquivo chamado banco.ts e faça as exportações necessárias. Crie um arquivo chamado app.ts que tenha leitura de dados e um loop semelhante ao demonstrado abaixo e implemente todas as funcionalidades da classe banco.

   Além de instalar a biblioteca prompt-sync, instale também a versão tipada dela: npm install @types/prompt-sync

```
import prompt from "prompt-sync";
import { Conta, Banco } from "./banco";

let input = prompt();
let b: Banco = new Banco();
let opcao: String = '';

do {
    console.log('\nBem vindo\nDigite uma opção:');
    console.log('Contas:\n' +
    console.log('1 - Inserir       2 - Consultar    3 - Sacar\n' +
                '4 - Depositar     5 - Excluir      6 - Transferir\n' +
                '7 – Totalizações
    console.log('Clientes:\n');
    console.log('8 - Inserir       9 - Consultar    10 - Associar\n' +
    console.log('Sair\n');
    opcao = input("Opção:");

    switch (opcao) {
        case "1":
            inserir();
            break
        case "2":
            consultar();
            break
        //...
    }
    input("Operação finalizada. Digite <enter>");
} while (opcao != "0");
console.log("Aplicação encerrada");


function inserir(): void {
    console.log("\nCadastrar conta\n");
    let numero: string = input('Digite o número da conta:');

    let conta: Conta;
    conta = new Conta(numero, 0);
```

```
    b.inserir(conta);
}//...
```