

Checkpoint 4 – 2º Semestre 2023

Códigos de Alta Performance

Profª. Patrícia Magna

O objetivo deste trabalho é elaborar um sistema para priorizar o contato de um determinado banco com os clientes visando oferecer novos serviços e aplicações bancárias mais rentáveis.

Cada cliente do banco é cadastrado nesse sistema de priorização de contato pelo gerente da conta. Este deve selecionar um certo número de clientes que considera “de interesse” para o banco. Assim, o gerente inscreve o cliente preenchendo uma ficha com seus dados, conforme mostra registro a seguir:

Nome:
CPF/CNPJ:
Número da Conta:
Tipo de Conta (Física ou Jurídica)
Saldo em Aplicações:

O registro de cada cliente é inserido na árvore de busca binária (ABB), sendo que o **critério de organização dos registros na ABB é o saldo em aplicações dos clientes**.

De tempo em tempo, esse cadastro é usado para que um novo serviço e/ou aplicação possa ser oferecido para os clientes.

Para simulação deste sistema o programa em JAVA deve ter um menu principal com as seguintes opções:

1. Inscrição de cliente: essa opção é usada para ler do teclado todas as informações do cliente e realizar o armazenamento do registro do candidato na ABB correspondente ao tipo de conta, ou seja, ABB de pessoas físicas ou ABB de pessoas jurídicas.
2. Oferta de novo serviço e/ou aplicação: é usada para ler do teclado o saldo mínimo necessário para a oferta a ser feita e qual tipo de cliente (pessoa física ou jurídica). Em seguida, o programa deve percorrer a ABB do tipo de conta exigida pela oferta a fim de buscar todos os clientes que se adequem a oferta, gerando uma lista com os clientes que devem ser contactados (seus registros) em ordem decrescente de saldo em aplicações (obrigatoriamente a ordem deve ser gerada pelo percurso da ABB e não por ordenação posterior da lista).

Com a lista de clientes aptos para receber a oferta pronta o programa deve:

- I. Retirar o registro do cliente com maior saldo da lista gerada para que o setor de divulgação entre em contato com o cliente e este pode aceitar ou não a oferta do banco. Assim, se o cliente:
 - Aceitar a oferta, o programa deve apresentar uma mensagem na tela de saída para avisar este fato. Em seguida, o registro do cliente deve ser retirado também da ABB,

uma vez que apenas 1 oferta deve ser feita para cada cliente em um determinado período de tempo.

- Não aceitar a oferta, o programa deve apresentar uma mensagem na tela de saída para avisar a recusa por parte do cliente e o registro do cliente permanece na ABB.

II. Programa prossegue retirando o próximo cliente da lista que deve ser contactado, até esvaziar a lista.

3. Menu de consulta: nessa opção um submenu deve ser apresentado com as seguintes opções na ABB correspondente ao tipo de cliente:

- 1) Consulta cliente buscando pelo CPF ou CNPJ apresentando todas as informações do registro do cliente;
- 2) Atualização do saldo do cliente buscando pelo número da conta (apresentando todas as informações do registro do cliente após a alteração do saldo);
- 3) Apresenta a quantidade de clientes armazenado na ABB.
- 4) Apresenta a quantidade de clientes com saldo acima de um valor a ser consultado;
- 5) Voltar menu principal.

4. Encerra o programa: com essa opção se as ABBs ainda possuam registros de clientes, estas devem ser esvaziadas apresentando os registros de clientes que ainda estavam armazenados e só então o programa deve ser encerrado.

A aplicação já está parcialmente implementada e deve ser usada para realização do projeto.

```
public class DivulgaOfertas {  
  
    /*  
    * NOMES E RM dos alunos que compõem o grupo  
    */  
  
    public static void main(String[] args) {  
  
        Scanner le = new Scanner(System.in);  
  
        /*  
        * Cria a uma árvore de busca binária para cada tipo de conta  
        * (pessoa física ou jurídica)  
        */  
  
        int opcao, op, numeroConta;  
        String nome, cpfCnpj;  
        String tipoConta = null;  
        double saldo;  
        do {  
            System.out.println(" 0 - Encerrar o programa");  
            System.out.println(" 1 - Inscrição cliente");  
            System.out.println(" 2 - Oferta de novo serviço e/ou aplicação");  
            System.out.println(" 3 - Entrar no Submenu ");  
  
            opcao = le.nextInt();  
            switch (opcao) {  
                case 1:  
                    System.out.print("Digite nome: ");  
                    nome = le.next();  
                    System.out.print("Digite cpf: ");  
                    cpfCnpj = le.next();  
                    System.out.print("Digite número da conta: ");  
                    numeroConta = le.nextInt();  
                    do {
```

```

        System.out.print("Digite 1- Pessoa Física  2- Pessoa
Jurídica: ");

        op = le.nextInt();
        switch (op) {
            case 1:
                tipoConta = "Física";
                break;
            case 2:
                tipoConta = "Jurídica";
                break;
            default:
                System.out.println("Opção inválida ");

                op = -1;
            }
        } while (op == -1);
        System.out.print("Informe saldo em aplicações R$: ");
        saldo = le.nextDouble();

/*
 * Intancia um objeto da classe Cliente e insere na ABB correspondente
 * a tipo de conta
 */
        break;
    case 2:
        System.out.print("Qual tipo de conta a oferta se
destina? ");

        do {
            System.out.print("Digite 1- Pessoa Física  2-
Pessoa Jurídica: ");

            op = le.nextInt();
            switch (op) {
                case 1:
                    tipoConta = "Física";
                    break;
                case 2:
                    tipoConta = "Jurídica";
                    break;
                default:
                    System.out.println("Opção inválida ");
                    op = -1;
            }
        } while (op == -1);
        System.out.print("Qual o valor de saldo mínimo exigido:
R$ ");

        saldo = le.nextDouble();

/*
 * Fazendo uso de um método da classe ABB, desenvolvido para este
 * problema, uma lista de clientes aptos para a oferta é gerada.
 * Nesse trecho de programa que tentar fazer o contato com todos os
 * clientes presente na lista.
 */
        break;
    case 3:

/*
 * Implemente o submenu descrito no texto
 */
        break;
    }
    } while (opcao != 0);
    System.out.println("Clientes que não aceitaram ou não estavam
adequados para a oferta");

/*
 * Esvazia as ABBs apresentando todos os clientes que aguardam nova oportunidade
 */
    le.close();

}
}

```

Informações sobre a entrega

Data de Entrega: 16 de outubro de 2023

Grupo: mesmo grupo da *challenge* Kraft Heinz.

Entrega: upload do projeto JAVA zipado como entrega de trabalhos no portal da FIAP

Importante: Os nomes e RMs dos componentes do grupo devem constar como comentário do código fonte da classe `DivulgaOfertas`.