Estrutura de dados ADO 1

Aluno: Davi da Silva Patricio

Classe main:

```
import java.util.Scanner;

public class App1 {
    public static void main(String[] args) throws Exception {

// Criação de vetor de objetos para criar e manipular dados de componentes
                        e periféricos de computador
  /*esse programa ajuda a registrar dados de componentes e periféricos de
   * computador(com atributos como, nome do produto, marca, cor e valor)
                    usando as funções passadas nas aulas para
   * manipular os dados como função para adicionar, função para aummentar
                    capacidade, função para mostrar a quantidade
   * de objetos criados, função para adicionar no inicio, entre outras*/

                    VetorObjeto vetor = new VetorObjeto(5);
                    Scanner scan = new Scanner (System.in);
                                int opcao = 0;
                                    do {
                System.out.println("====Menu de Opções====");
                     System.out.println("1. Adicionar");
                System.out.println("2. Visualizar quantidade");
                 System.out.println("3. Buscar por posição");
            System.out.println("4. Verificar se componente existe");
                 System.out.println("5. Adicionar no inicio");
                      System.out.println("6. Remover");
                       System.out.println("0 Sair");
                          opcao = scan.nextInt();
                             scan.nextLine();

   Componentes c1 = new Componentes ("Pen Drive","San Disk","preto", 40);
Componentes c2 = new Componentes ("Teclado Mecanico", "Logitech", "Preto",
                                        800);
     Componentes c3 = new Componentes ("Mouse", "Razer", "Preto", 350);
    Componentes c4 = new Componentes ("Monitor", "LG", "Branco", 2000);
  Componentes c5 = new Componentes ("HeadSet", "HyperX", "Vermelho", 400);
                                switch (opcao) {
                                    case 1:
                                     try {
                                vetor.adicionar(c1);
                                vetor.adicionar(c2);
                                vetor.adicionar(c3);
                                vetor.adicionar(c4);
                                vetor.adicionar(c5);
                                        }
```

```
                    catch (Exception e){
                    e.printStackTrace();
                              }
                            break;

                    case 2:
            System.out.println(vetor.getTamanho());
                            break;

                    case 3:
          System.out.println("qual a posição no vetor?");
                  int posicao = scan.nextInt();
             System.out.println(vetor.busca(posicao));
                              ;
                            break;

                    case 4:
            System.out.println("qual o objeto?");
                String posicao1 = scan.next();
                   vetor.busca1(posicao1);
                            break;

                    case 5:
                Object elemento = new Object();
                    int posicaoInicio = 0;
          vetor.adicionarInicio(posicaoInicio, elemento);
                            break;

                    case 6:
            System.out.println("qual a posição?");
                     int posExcluir = 0;
                 posExcluir = scan.nextInt();
            vetor.remove(vetor.setTamaho(posExcluir));
                            break;
                           default:
  System.out.println("opcao invalida, tente novamente!");
                            break;
                              }
                    } while (opcao != 0);
                        scan.close();
                              }
                              }
```

Classe Componentes:

```
public class Componentes {
// atributos: modelo, marca, cor, valor

    private String modelo;
    private String marca;
    private String cor;
    private double valor;

public Componentes (String modelo, String marca, String cor, double valor)
    {
    this.modelo = modelo;
    this.marca = marca;
    this.cor = cor;
    this.valor = valor;
    }

    public String getModelo() {
        return modelo;
    }
public void setModelo (String modelo) {
    this.modelo = modelo;
    }

    public String getMarca() {
        return this.marca;
    }
public void setMarca (String marca) {
    this.marca = marca;
    }

    public String getCor() {
        return this.cor;
    }
public void setCor (String cor) {
    this.cor = cor;
    }

    public double getValor() {
        return this.valor;
    }
public void setValor (double valor) {
    this.valor = valor;
    }

    @override
    public String toString() {
```

```
return "Componentes (" + "modelo=" + modelo + ", marca=" + marca + ", cor="
                    + cor + ", valor=" + valor + ',';
                                    }
                                    }
```

Classe vetorObjeto:

```
                        public class VetorObjeto {
                        private Object[] elementos;
                            private int tamanho;

                    public VetorObjeto (int capacidade) {
                        this.elementos = new Object[5];
                                this.tamanho = 0;
                                    }
                     public void aumentarCapacidade () {
                if (this.tamanho == this.elementos.length) {
        Object[] novoElemento = new Object [this.elementos.length * 2];
               for (int i = 0; i < this.elementos.length; i++) {
                     novoElemento[i] = this.elementos[i];
                                    }
                        this.elementos = novoElemento;
                                    }
                                    }

          public void adicionar (Object elemento) throws Exception {
                            this.aumentarCapacidade();
                  if (this.tamanho < this.elementos.length) {
                    this.elementos[this.tamanho] = elemento;
                                this.tamanho++;
                                    }
                                  else {
      throw new Exception ("Não é possivel adicionar novos elementos, vetor
                                    cheio");
                                    }
                                    }

                            public int getTamanho() {
                               return this.tamanho;
                                    }

                        public int setTamaho(int tamanho) {
                           return this.tamanho = tamanho;
                                    }

              public Object busca (int posicao) throws Exception {
                    if (posicao >=0 && posicao < tamanho) {
```

```
                    return elementos[posicao];
                                }
                            else {
                throw new Exception ("posicao invalida");
                                }
                                }

                  public int busca1 (Object elemento) {
                    for (int i = 0; i < tamanho;i++) {
                   if (elementos[i].equals(elemento)) {
                                return i;
                                }
                                }
                            return - 1;
                                }

public boolean adicionarInicio (int posicao, Object elemento) throws
                                Exception {
                        this.aumentarCapacidade();
                 if (posicao >=0 && posicao < tamanho) {
              for (int i = this.tamanho-1; i > posicao; i--) {
                 this.elementos[i + 1] = this.elementos[i];
                                }
                    this.elementos[posicao] = elemento;
                              this.tamanho++;
                                }
                            else {
                throw new Exception ("posição invalida");
                                }
                            return true;
                                }

          public void remove (int posicao) throws Exception {
                if (posicao >=0 && posicao < tamanho) {
           for (int i = posicao; i < this.tamanho - 1; i++) {
                 this.elementos[i] = this.elementos[i + 1];
                                }
                            this.tamanho--;
                                }
                            else {
                throw new Exception ("posicao invalida");
                                }
                                }
                                }
```