

```

package td6;

import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.Scanner;

import td6.Pessoa;
import td6.PessoaFisica;
import td6.PessoaJuridica;
import td6.ColecaoPessoa;
import td6.Colecao;

public class App {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Calendar hoje = Calendar.getInstance();
        int op = 0, i = 0;
        Pessoa[] dados = new Pessoa[100];
        ColecaoPessoa cPessoa = new ColecaoPessoa();
        do {
            System.out.println("----- MENU -----");
            System.out.println("Digite a opção: \n1 - Inserir\n2 -
Remover\n3 - Atualizar\n4 - Pesquisar\n5 - Coleção vazia\n6 - Imprimir\n7 -
Retornar Objeto");

            op = scan.nextInt();

            if(op == 1) {
                cPessoa.inserir(dados[0]);
            } else if(op == 2) {
                System.out.println("Digite qual indice deseja
remover: ");

                int x = scan.nextInt();
                cPessoa.remover(x);
            } else if(op == 3) {
                System.out.println("Deseja atualizar qual
indice?");

                int l = scan.nextInt();
                cPessoa.atualizar(l ,dados[0]);
            } else if(op == 4) {
                cPessoa.pesquisar(dados[0]);
            } else if(op == 5) {
                cPessoa.colecaoEstaVazia();
            } else if(op == 6) {
                cPessoa.imprimirDadosColecao();
            } else if(op == 7) {
                System.out.println("Digite qual indice deseja
receber: ");

                int x = scan.nextInt();
                cPessoa.retornarObjeto(x);
            }
        }while(op == 1 || op == 2 || op == 3 || op == 4 || op == 5 ||
op == 6 || op == 7);

        for(int x = 0; x < dados.length; x++) {
            if(dados[x] == null) {

```

```

                break;
            } else {
                System.out.println("Nome: "+dados[x].getNome() +
", CPF: "+((PessoaFisica) dados[x]).getCpf());
            }
        }
    }
}

```

```

package td6;

```

```

import java.util.Calendar;

```

```

import java.util.GregorianCalendar;

```

```

public class PessoaJuridica extends Pessoa{

```

```

    private int dia;

```

```

    private int mes;

```

```

    private int ano;

```

```

    private String cnpj;

```

```

    private GregorianCalendar dataCriacao;

```

```

    public PessoaJuridica() {

```

```

        super();

```

```

    }

```

```

    public PessoaJuridica(int dia, int mes, int ano, String cnpj, GregorianCalendar dataCriacao)
{

```

```

        super();

```

```

        this.dia = dia;

```

```

        this.mes = mes;

```

```

        this.ano = ano;

```

```
        this.cnpj = cnpj;
        this.dataCriacao = dataCriacao;
    }
```

```
    boolean equals() {
        return true;
    }
```

```
    public boolean validarDocumento() {
        if(this.cnpj.length() == 18) {
            return true;
        } else {
            return false;
        }
    }
```

```
    public int CalculaIdade(Pessoa dados[], Calendar hoje) {
        int idade = 0;
        if(this.mes < hoje.get(Calendar.MONTH) + 1) {
            idade = hoje.get(Calendar.YEAR) - this.ano;
        } else if(this.mes > hoje.get(Calendar.MONTH) + 1) {
            idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;

        } else if(this.mes == hoje.get(Calendar.MONTH) + 1) {
            if(this.dia > hoje.get(Calendar.DAY_OF_MONTH)) {
                idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;

            } else {
                idade = hoje.get(Calendar.YEAR) - this.ano;
            }
        }
    }
```

```
        }  
    }  
    return idade;  
}
```

```
boolean equals(String cnpj) {  
    if(cnpj.equals(this.cnpj)) {  
        return true;  
    }  
    return false;  
}
```

```
public int getDia() {  
    return dia;  
}
```

```
public void setDia(int dia) {  
    this.dia = dia;  
}
```

```
public int getMes() {  
    return mes;  
}
```

```
public void setMes(int mes) {  
    this.mes = mes;  
}
```

```
    public int getAno() {  
        return ano;  
    }  
  
    public void setAno(int ano) {  
        this.ano = ano;  
    }  
  
    public String getCnpj() {  
        return cnpj;  
    }  
  
    public void setCnpj(String cnpj) {  
        this.cnpj = cnpj;  
    }  
  
    public GregorianCalendar getDataCriacao() {  
        return dataCriacao;  
    }  
  
    public void setDataCriacao(GregorianCalendar dataCriacao) {  
        this.dataCriacao = dataCriacao;  
    }  
}
```

```
package td6;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;

import java.util.Scanner;

public class PessoaFisica extends Pessoa {

    private int dia;

    private int mes;

    private int ano;

    private String cpf;

    private GregorianCalendar dataNascimento;

    Scanner scan = new Scanner(System.in);


    public boolean validarDocumento() {

        if(this.cpf.length() == 14) {

            return true;

        } else {

            return false;

        }

    }


    boolean equals(String cpf) {

        if(cpf.equals(this.cpf)) {

            return true;

        }

        return false;

    }

}
```

```
public int CalculaIdade(Pessoa dados[], Calendar hoje) {  
    int idade = 0;  
    if(this.mes < hoje.get(Calendar.MONTH) + 1) {  
        idade = hoje.get(Calendar.YEAR) - this.ano;  
    } else if(this.mes > hoje.get(Calendar.MONTH) + 1) {  
        idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;  
  
    } else if(this.mes == hoje.get(Calendar.MONTH) + 1) {  
        if(this.getDia() > hoje.get(Calendar.DAY_OF_MONTH)) {  
            idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;  
  
        } else {  
            idade = hoje.get(Calendar.YEAR) - this.ano;  
  
        }  
    }  
    return idade;  
}
```

```
public String getCpf() {  
    return cpf;  
}
```

```
public void setCpf(String cpf) {  
    this.cpf = cpf;  
}
```

```
public int getDia() {  
    return dia;  
}
```

```
}
```

```
public void setDia(int dia) {
```

```
    this.dia = dia;
```

```
}
```

```
public int getMes() {
```

```
    return mes;
```

```
}
```

```
public void setMes(int mes) {
```

```
    this.mes = mes;
```

```
}
```

```
public int getAno() {
```

```
    return ano;
```

```
}
```

```
public void setAno(int ano) {
```

```
    this.ano = ano;
```

```
}
```

```
public GregorianCalendar getDataNascimento() {
```

```
    return dataNascimento;
```

```
}
```

```
public void setDataNascimento(GregorianCalendar dataNascimento) {
```

```
    this.dataNascimento = dataNascimento;
```

```
}
```



```
}
```

```
package td6;
```

```
public interface Colecao {  
    public boolean inserir(Pessoa p);  
    public boolean remover(int indice);  
    public void atualizar(int indice, Pessoa p);  
    public boolean pesquisar(Pessoa p);  
    public boolean colecaoEstaVazia();  
    public void imprimirDadosColecao();  
    public Pessoa retornarObjeto(int indice);  
}
```

```
package td6;
```

```
import java.util.Arrays;  
import java.util.Calendar;  
import java.util.GregorianCalendar;  
import java.util.Scanner;  
public class ColecaoPessoa implements Colecao{  
    Scanner scan = new Scanner(System.in);  
    public ColecaoPessoa() {  
        super();  
    }  
    private Pessoa dados[] = new Pessoa[100];  
    private int posicaoCorrente;
```

```

public Pessoa[] getDados() {
    return dados;
}

public void setDados(Pessoa[] dados) {
    this.dados = dados;
}

public int getPosicaoCorrente() {
    return posicaoCorrente;
}

public void setPosicaoCorrente(int posicaoCorrente) {
    this.posicaoCorrente = posicaoCorrente;
}

int cont = 0;

public boolean inserir(Pessoa p) {
    int op = 0;
    System.out.println("Digite 1 para Pessoa Fisica e 2 para Pessoa Juridica:");
    op = scan.nextInt();
    scan.nextLine();
    if(op == 1) {
        dados[cont] = new PessoaFisica();
        System.out.println("Digite o nome ai: ");
        dados[cont].setNome(scan.nextLine());
        System.out.println("Digite o CPF: ");
        ((PessoaFisica) dados[cont]).setCpf(scan.nextLine());
        System.out.println("Digite a data de nascimento: dd mm aaaa: ");
        ((PessoaFisica) dados[cont]).setDia(scan.nextInt());
        ((PessoaFisica) dados[cont]).setMes(scan.nextInt());
        ((PessoaFisica) dados[cont]).setAno(scan.nextInt());
    }
}

```

```

        scan.nextLine();

        ((PessoaFisica) dados[cont]).setDataNascimento(new
GregorianCalendar());

        ((PessoaFisica) dados[cont]).getDataNascimento().set(((PessoaFisica)
dados[cont]).getAno(), ((PessoaFisica) dados[cont]).getMes(), ((PessoaFisica)
dados[cont]).getDia());

        cont++;
    } else if (op == 2) {
        dados[cont] = new PessoaJuridica();

        System.out.println("Digite o nome da empresa: ");

        dados[cont].setNome(scan.nextLine());

        System.out.println("Digite o CNPJ: ");

        ((PessoaJuridica) dados[cont]).setCnpj(scan.nextLine());

        System.out.println("Digite a data de Criacao: dd mm aaaa: ");

        ((PessoaJuridica) dados[cont]).setDia(scan.nextInt());

        ((PessoaJuridica) dados[cont]).setMes(scan.nextInt());

        ((PessoaJuridica) dados[cont]).setAno(scan.nextInt());

        scan.nextLine();

        ((PessoaJuridica) dados[cont]).setDataCriacao(new GregorianCalendar());

        ((PessoaJuridica) dados[cont]).getDataCriacao().set(((PessoaJuridica)
dados[cont]).getAno(), ((PessoaJuridica) dados[cont]).getMes(), ((PessoaJuridica)
dados[cont]).getDia());

        cont++;
    }

    return true;
}

public boolean remover(int indice) {

    dados[indice] = null;

    return true;
}

```

```

public void atualizar(int indice, Pessoa p) {
    dados[indice] = new PessoaFisica();
    System.out.println("Digite o novo nome: ");
    dados[indice].setNome(scan.nextLine());
    System.out.println("Digite o novo CPF: ");
    ((PessoaFisica) dados[indice]).setCpf(scan.nextLine());
    System.out.println("Digite a nova data de nascimento: dd mm aaaa: ");
    ((PessoaFisica) dados[indice]).setDia(scan.nextInt());
    ((PessoaFisica) dados[indice]).setMes(scan.nextInt());
    ((PessoaFisica) dados[indice]).setAno(scan.nextInt());
    scan.nextLine();
    ((PessoaFisica) dados[indice]).setDataNascimento(new GregorianCalendar());
    ((PessoaFisica) dados[indice]).getDataNascimento().set(((PessoaFisica)
dados[indice]).getAno(), ((PessoaFisica) dados[indice]).getMes(), ((PessoaFisica)
dados[indice]).getDia());
}

public boolean pesquisar(Pessoa p) {
    System.out.println("Deseja pesquisar uma Pessoa Fisica ou Juridica? 1 pra Fisica 2
pra Juridica");
    int op = scan.nextInt();
    scan.nextLine();
    if(op == 1) {
        System.out.println("Digite o CPF: ");
        String aux = scan.nextLine();
        for(int i = 0; i < dados.length; i++) {
            if(dados[i] == null) {
                break;
            } else {

```

```

        if(aux.equals(((PessoaFisica) dados[i]).getCpf())){
            retornarObjeto(i);
            return true;
        } else {
            System.out.println("Não existe");
            return false;
        }
    }
}

return false;
}

```

```

public boolean colecaoEstaVazia() {
    if(dados[0] == null) {
        System.out.println("Coleção está vazia.");
        return true;
    } else {
        System.out.println("Coleção não está vazia.");
        return false;
    }
}

```

```

public void imprimirDadosColecao() {
    for(int i = 0; i < dados.length; i++) {
        if(dados[i] == null) {
            break;
        } else {

```

```
                System.out.println("Nome: "+dados[i].getNome() + ", CPF: "+((PessoaFisica) dados[i]).getCpf()+ ", Data de Nascimento: "+((PessoaFisica) dados[i]).getDia()+ "/" +((PessoaFisica) dados[i]).getMes()+ "/" +((PessoaFisica) dados[i]).getAno());
```

```
            }
```

```
        }
```

```
    }
```

```
    public Pessoa retornarObjeto(int indice) {
```

```
        if(dados[indice] == null) {
```

```
            System.out.println("Esse objeto não existe.\n");
```

```
            return null;
```

```
        }
```

```
        System.out.println("Nome: "+ dados[indice].getNome() +" CPF: "+ ((PessoaFisica) dados[indice]).getCpf());
```

```
        return dados[indice];
```

```
    }
```

```
}
```

```
package td6;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
import java.util.Scanner;
```

```
public abstract class Pessoa{
```

```
    private String nome;
```

```
    private Calendar hoje = Calendar.getInstance();
```

```
Scanner scan = new Scanner(System.in);
```

```
public Pessoa() {  
    super();  
    this.nome = nome;  
    this.hoje = hoje;  
}
```

```
public String getNome() {  
    return nome;  
}
```

```
public void setNome(String nome) {  
    this.nome = nome;  
}
```

```
public boolean validarDocumento() {  
    return true;  
}
```

```
public int calcularIdade() {  
    return 1;  
}
```

```
public Calendar getHoje() {  
    return hoje;  
}
```

```
public void setHoje(Calendar hoje) {  
    this.hoje = hoje;  
}
```

```
}
```

