

Classe Coleção:

```
package TrabalhoDirigido6;

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

Classe ColeçãoPessoa:

```
package TrabalhoDirigido6;

public class ColecaoPessoa implements Colecao{
    private Pessoa[] dados = new Pessoa[100];
    private int posicaoCorrente = 0, insert = 0;

    public boolean inserir(Pessoa p) {
        for(int i = 0 ; i < dados.length ; i++) {
            if(this.dados[i] == null) {
                this.dados[i] = p;
                System.out.println("Pessoa " +p.getNome()+"
inserida");
                this.posicaoCorrente = posicaoCorrente + 1;
                insert++;
                return true;
            }
        }
        return false;
    }

    public boolean pesquisar(Pessoa p) {
        for(int i = 0 ; i < dados.length ; i++) {
            if(this.dados[i] != null &&
this.dados[i].equals(p)) {
                System.out.println(p);
                return true;
            }
        }
        System.out.println("Fora da coleção");
        return false;
    }
}
```

```

    }

    public void atualizar(int indice, Pessoa p) {
        this.dados[indice] = p;
    }

    public void imprimirDadosColecao() {
        for(int i = 0 ; i < dados.length ; i++) {
            if(this.dados[i] != null) {
                System.out.println(this.dados[i]);
            }
        }
    }

    public boolean remover() {
        for(int i = dados.length - 1 ; i >= 0 ; i--) {
            if(this.dados[i] != null) {
                this.dados[i] = null;
                System.out.println("Pessoa " + i + "
removida");
                insert--;
                return true;
            }
        }
        return false;
    }

    public boolean remover(int indice) {
        if(this.dados[indice] != null) {
            this.dados[indice] = null;
            System.out.println("Pessoa " + indice + "
removido");
            insert--;
            return true;
        }
        System.out.println("Pessoa" + indice + " nulo");
        return false;
    }

    public boolean colecaoEstaVazia() {
        for(int i = 0 ; i < dados.length ; i++) {
            if(this.dados[i] != null) {
                System.out.println("Coleção não esta vazia");
                return false;
            }
        }
        System.out.println("Coleção está vazia");
        return true;
    }

```

```

    }

    public Pessoa retornarObjeto(int indice) {
        if(indice > insert) {
            System.out.println("Pessoa desejada não
encontrada.");
            return null;
        }
        else {
            return this.dados[indice];
        }
    }
}

```

Classe Pessoa:

```

package TrabalhoDirigido6;

public abstract class Pessoa {
    private String nome;

    public String toString() {
        return String.format("Nome: %s", nome);
    }
    public String getNome() {
        return nome;
    }

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

    public abstract boolean ValidarDocumento();

    public abstract int calcularidade();
}

```

Classe PessoaJurídica:

```

package TrabalhoDirigido6;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class PessoaJurídica extends Pessoa{

```

```

private String cnpj;
private Calendar dataCriacao;

public String getCnpj() {
    return cnpj;
}

public void setCnpj(String cnpj) {
    this.cnpj = cnpj;
}

public Calendar getDataCriacao() {
    return dataCriacao;
}

public void setDataCriacao(Calendar dataCriacao) {
    this.dataCriacao = dataCriacao;
}

public boolean equals(PessoaJurídica person_j) {
    if(this.cnpj.equals(person_j.getCnpj())) {
        System.out.println(this.getNome() + " e "+
person_j.getNome() +"são a mesma pessoa");
        return true;
    }
    System.out.println(this.getNome()+" e
"+person_j.getNome()+" são pessoas diferentes");
    return false;
}

public String toString(){
    SimpleDateFormat sfd = new
SimpleDateFormat("dd/MM/yyyy");
    return String.format("Nome:%s\nCnpj:%s\nData de
criação:%s",this.getNome(),
this.getCnpj(),sfd.format(this.dataCriacao.getTime()));
}

public int calcularidade() {
    GregorianCalendar dataAtual = new GregorianCalendar();
    int age;
    age = dataAtual.get(Calendar.YEAR) -
this.dataCriacao.get(Calendar.YEAR);
    if(dataAtual.get(Calendar.MONTH) <
this.dataCriacao.get(Calendar.MONTH)) {
        age--;
        System.out.println(this.getNome()+" tem "+ age +"
anos.");
    }
}

```

```

        return age;
    }
    else if (dataAtual.get(Calendar.MONTH) >
this.dataCriacao.get(Calendar.MONTH)) {
        System.out.println(this.getNome()+" tem "+ age + "
anos.");
        return age;
    }
    else {
        if (this.dataCriacao.get(Calendar.DAY_OF_MONTH) <
dataAtual.get(Calendar.DAY_OF_MONTH)) {
            age--;
            System.out.println(this.getNome()+" tem "+
age + " anos.");
        }
    }
    return age;
}

public boolean ValidarDocumento() {
    if (this.getCnpj().length() == 18) {
        System.out.println("Cnpj de "+this.getNome()+"
está correto");
        return true;
    }
    else {
        System.out.println("CNPJ de "+this.getNome()+"
está incorreto");
        return false;
    }
}
}

```

Classe PessoaFísica:

```

package TrabalhoDirigido6;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class PessoaFísica extends Pessoa{
    private String cpf;
    private Calendar dataNascimento;

    public Calendar getDataNascimento() {
        return dataNascimento;
    }
}

```

```

    }

    public void setDataNascimento(Calendar dataNascimento) {
        this.dataNascimento = dataNascimento;
    }

    public String getCpf() {
        return cpf;
    }

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

    public boolean equals(PessoaFísica person) {
        if(this.cpf.equals(person.getCpf())) {
            System.out.println(this.getNome() + " e "+
person.getNome() +" são pessoas a mesma pessoa");
            return true;
        }
        System.out.println(this.getNome()+" e
"+person.getNome()+" são pessoas diferentes");
        return false;
    }

    public String toString() {
        SimpleDateFormat simpledate = new
SimpleDateFormat("dd/MM/yyyy");
        return String.format("Nome:%s\nCPF:%s\nData de
Nascimento:%s",this.getNome(),
this.getCpf(),simpledate.format(this.dataNascimento.getTime()));
    }

    public boolean ValidarDocumento() {
        if(this.getCpf().length() == 14) {
            System.out.println("CPF de "+this.getNome()+" está
correto");
            return true;
        }
        else {
            System.out.println("CPF de "+this.getNome()+" está
incorreto");
            return false;
        }
    }

    public int calcularidade() {
        GregorianCalendar Date_Current = new
GregorianCalendar();
        int age;

```

```

        age = Date_Current.get(Calendar.YEAR) -
this.dataNascimento.get(Calendar.YEAR);
        if(Date_Current.get(Calendar.MONTH) <
this.dataNascimento.get(Calendar.MONTH)) {
            age--;
            System.out.println(this.getNome()+" tem "+ age +"
anos.");
            return age;
        }
        else if(Date_Current.get(Calendar.MONTH)>
this.dataNascimento.get(Calendar.MONTH)) {
            System.out.println(this.getNome()+" tem "+ age +"
anos.");
            return age;
        }
        else {
            if(this.dataNascimento.get(Calendar.DAY_OF_MONTH)
< Date_Current.get(Calendar.DAY_OF_MONTH)) {
                age--;
                System.out.println(this.getNome()+" tem "+
age +" anos.");
                return age;
            }
        }
        return age;
    }
}

```

Classe App:

```

package TrabalhoDirigido6;

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

public class App {

    public static void main(String[] args) {
        ColecaoPessoa collection = new ColecaoPessoa();

        PessoaFísica Pf1    = new PessoaFísica();
        PessoaFísica Pf2    = new PessoaFísica();
        PessoaJurídica Pj1 = new PessoaJurídica();
        PessoaJurídica Pj2 = new PessoaJurídica();

        Pf1.setNome("Nilta Mota");
        Pf1.setCpf("497.145.237-58");
    }
}

```

```
Pf2.setNome("Chico Mota");
Pf2.setCpf("497.145.237-58");

Pj1.setNome("Lucas Mota");
Pj1.setCnpj("41.151.272/1401-12");

Pj2.setNome("Alisson Mota");
Pj2.setCnpj("71.752.959/0000-78-");

Calendar date_1 = new GregorianCalendar(1998,5,12);
Calendar date_2 = new GregorianCalendar(2015,7,10);
Calendar date_3 = new GregorianCalendar(2011,2,27);
Calendar date_4 = new GregorianCalendar(2004,4,12);

Pf1.setDataNascimento(date_1);
Pf2.setDataNascimento(date_2);

Pj1.setDataCriacao(date_3);
Pj2.setDataCriacao(date_4);

//Verificando o método inserir()
collection.inserir(Pj1);
collection.inserir(Pj2);
collection.inserir(Pf1);
collection.inserir(Pf2);

//Verificando o método equals()
Pf1.equals(Pf2);
Pj1.equals(Pj2);

//Verificando o método calcularidade()
Pf1.calcularidade();
Pf2.calcularidade();
Pj1.calcularidade();
Pj2.calcularidade();

//Verificando o método ValidarDocumento()
Pf1.ValidarDocumento();
Pf2.ValidarDocumento();
Pj1.ValidarDocumento();
Pj2.ValidarDocumento();

//Verificando método imprimirDadosColecao()
collection.imprimirDadosColecao();

//Verificando método remover()
collection.remover(1);
```



```
//Verificando método remover()
collection.remover();

//Verificando método atualizar()
collection.atualizar(1, Pf2);

//Verificando método pesquisar()
collection.pesquisar(Pj2);

//Verificando método colecaoEstaVazia()
collection.colecaoEstaVazia();

//Verificando método retornarObjeto()
collection.retornarObjeto(1);

//Verificando método imprimirDadosColecao()
collection.imprimirDadosColecao();
}
}
```