



```
package fintech;
```

```
public class Pessoa {
```

```
    public String  nome;
    public String  email;
    public String  nascimento;
    public String  endereco;
    public String  bairro;
    public String  cidade;
    public String  estado;
    public String  cep;
```

```
    public Pessoa() {
        // TODO Auto-generated constructor stub
    }
```

```
public String getNome() {  
    return nome;  
}  
  
public void setNome(String nome) {  
    this.nome = nome;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getNascimento() {  
    return nascimento;  
}  
  
public void setNascimento(String nascimento) {  
    this.nascimento = nascimento;  
}  
  
public String getEndereco() {  
    return endereco;  
}  
  
public void setEndereco(String endereco) {  
    this.endereco = endereco;  
}  
  
public String getBairro() {  
    return bairro;  
}  
  
public void setBairro(String bairro) {  
    this.bairro = bairro;  
}  
  
public String getCidade() {  
    return cidade;  
}  
  
public void setCidade(String cidade) {  
    this.cidade = cidade;  
}
```

```
public String getEstado() {  
    return estado;  
}  
  
public void setEstado(String estado) {  
    this.estado = estado;  
}  
  
public String getCep() {  
    return cep;  
}  
  
public void setCep(String cep) {  
    this.cep = cep;  
}  
}
```

```
package fintech;
```

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

```
import fintech.PessoaPF;
```

```
public class PessoaPF extends Pessoa {  
  
    private String  cpf;  
    private String  rg;  
  
    public PessoaPF() {  
        // TODO Auto-generated constructor stub  
    }  
  
    public String getCpf() {  
        return cpf;  
    }  
  
    public void setCpf(String cpf) {  
        this.cpf = cpf;  
    }  
  
    public String getRg() {  
        return rg;  
    }  
}
```

```
public void setRg(String rg) {  
    this.rg = rg;  
}
```

```
public PessoaPF setCadastroPessoaPF(Scanner scanner) {
```

```
    PessoaPF pf = new PessoaPF();  
    textoNL("Preencha os dados abaixo corretamente!");  
    texto("Nome: "); setNome(scanner.nextLine());  
    texto("Rg: "); setRg(scanner.nextLine());  
    texto("Cpf: "); setCpf(scanner.nextLine());  
    texto("Email: "); setEmail(scanner.nextLine());  
    texto("Data nascimento: "); setNascimento(scanner.nextLine());  
    texto("Endereço: "); setEndereco(scanner.nextLine());  
    texto("Bairro: "); setBairro(scanner.nextLine());  
    texto("Cidade: "); setCidade(scanner.nextLine());  
    texto("Estado: "); setEstado(scanner.nextLine());  
    texto("Cep: "); setCep(scanner.nextLine());
```

```
    return pf;
```

```
}
```

```
public void consultarPessoaPF(List<PessoaPF> dadosPF, Scanner scanner) {
```

```
    String busca;  
    int encontrado;
```

```
    if (dadosPF.size() == 0) {  
        textoNL("Não existem cadastros!");  
    }  
    else {
```

```
        texto("Digite o CPF do cadastro: ");  
        busca = scanner.next();
```

```
        encontrado = -1;  
        PessoaPF pf = dadosPF.get(0);  
        for (int i = 0; i < dadosPF.size(); i++) {
```

```
            dadosPF.get(i);  
            if (busca.equals(pf.getCpf())) {  
                textoNL("Cadastro encontrado.");  
                pf.getDadosPessoaPF();  
                encontrado = 1;  
            }  
        }
```

```
    }  
    if (encontrado < 0)
```

```

        textoNL("Cadastro não foi encontrado, tente novamente!");
    }

}

public void getDadosPessoaPF() {

    textoNL("Nome: " + getNome());
    textoNL("Rg: " + getRg());
    textoNL("Cpf: " + getCpf());
    textoNL("Email: " + getEmail());
    textoNL("Data nascimento: " + getNascimento());
    textoNL("Endereço: " + getEndereco());
    textoNL("Bairro: " + getBairro());
    textoNL("Cidade: " + getCidade());
    textoNL("Estado: " + getEstado());
    textoNL("Cep: " + getCep());

}

public void deletarPessoaPF(List<PessoaPF> dadosPF, Scanner scanner) {

    String busca;
    int encontrado;

    if (dadosPF.size() == 0) {
        textoNL("Não existem cadastros!");
    }
    else {

        texto("Digite CPF do cadastro que deseja deletar: ");
        busca = scanner.next();

        encontrado = -1;
        PessoaPF pf = dadosPF.get(0);
        for (int i = 0; i < dadosPF.size(); i++)
        {
            dadosPF.get(i);
            if (busca.equals(pf.getCpf())){
                System.out.println("Cadastro deletado!");
                dadosPF.remove(i);
                encontrado = 1;
            }
        }
        if (encontrado < 0) {
            textoNL("Cadastro não foi encontrado, tente novamente!");
        }
    }
}

```

```

        }

    }

    public void textoNL(String frase) {
        System.out.println(frase);
    }

    public void texto(String frase) {
        System.out.print(frase);
    }
}

```

```

package fintech;

```

```

import java.util.List;
import java.util.Scanner;

```

```

public class PessoaPJ extends Pessoa{

    private String cnpj;
    private String nire;

    public PessoaPJ() {
        // TODO Auto-generated constructor stub
    }

    public String getCnpj() {
        return cnpj;
    }

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

    public String getNire() {
        return nire;
    }

    public void setNire(String nire) {
        this.nire = nire;
    }

    public PessoaPJ setCadastroPessoaPJ(Scanner scanner) {

        PessoaPJ pj = new PessoaPJ();
    }
}

```

```

        textoNL("Preencha os dados abaixo corretamente!");
        texto("Razão social: "); setNome(scanner.nextLine());
        texto("NIRE: "); setNire(scanner.nextLine());
        texto("Cnpj: "); setCnpj(scanner.nextLine());
        texto("Email: "); setEmail(scanner.nextLine());
        texto("Data abertura: "); setNascimento(scanner.nextLine());
        texto("Endereço: "); setEndereco(scanner.nextLine());
        texto("Bairro: "); setBairro(scanner.nextLine());
        texto("Cidade: "); setCidade(scanner.nextLine());
        texto("Estado: "); setEstado(scanner.nextLine());
        texto("Cep: "); setCep(scanner.nextLine());

        return pj;
    }

    public void consultarPessoaPJ(List<PessoaPJ> dadosPJ, Scanner scanner) {

        String busca;
        int encontrado;

        if (dadosPJ.size() == 0) {
            textoNL("Não existem cadastros!");
        }
        else {

            texto("Digite o CNPJ do cadastro: ");
            busca = scanner.next();

            encontrado = -1;
            PessoaPJ pj = dadosPJ.get(0);
            for (int i = 0; i < dadosPJ.size(); i++) {

                dadosPJ.get(i);
                if(busca.equals(pj.getCnpj())) {
                    textoNL("Cadastro encontrado.");
                    pj.getDadosPessoaPJ();
                    encontrado = 1;
                }
            }
            if (encontrado < 0)
                textoNL("Cadastro não foi encontrado, tente novamente!");
        }
    }

    public void getDadosPessoaPJ() {

        textoNL("Razão social: " + getNome());
    }

```

```

        textoNL("Cnpj: " + getCnpj());
        textoNL("Nire: " + getNire());
        textoNL("Email: " + getEmail());
        textoNL("Data nascimento: " + getNascimento());
        textoNL("Endereço: " + getEndereco());
        textoNL("Bairro: " + getBairro());
        textoNL("Cidade: " + getCidade());
        textoNL("Estado: " + getEstado());
        textoNL("Cep: " + getCep());
    }

    public void deletarPessoaPJ(List<PessoaPJ> dadosPJ, Scanner scanner) {

        String busca;
        int encontrado;

        if (dadosPJ.size() == 0) {
            textoNL("Não existem cadastros!");
        }
        else {

            texto("Digite CNPJ do cadastro que deseja deletar: ");
            busca = scanner.next();

            encontrado = -1;
            PessoaPJ pj = dadosPJ.get(0);
            for (int i = 0; i < dadosPJ.size(); i++)
            {
                dadosPJ.get(i);
                if (busca.equals(pj.getCnpj())){
                    System.out.println("Cadastro deletado!");
                    dadosPJ.remove(i);
                    encontrado = 1;
                }
            }
            if (encontrado < 0) {
                textoNL("Cadastro não foi encontrado, tente novamente!");
            }
        }
    }

    private void textoNL(String frase) {
        System.out.println(frase);
    }
}

```



```
private void texto(String frase) {  
    System.out.print(frase);  
}  
}
```
