

Programação Orientada a Objetos

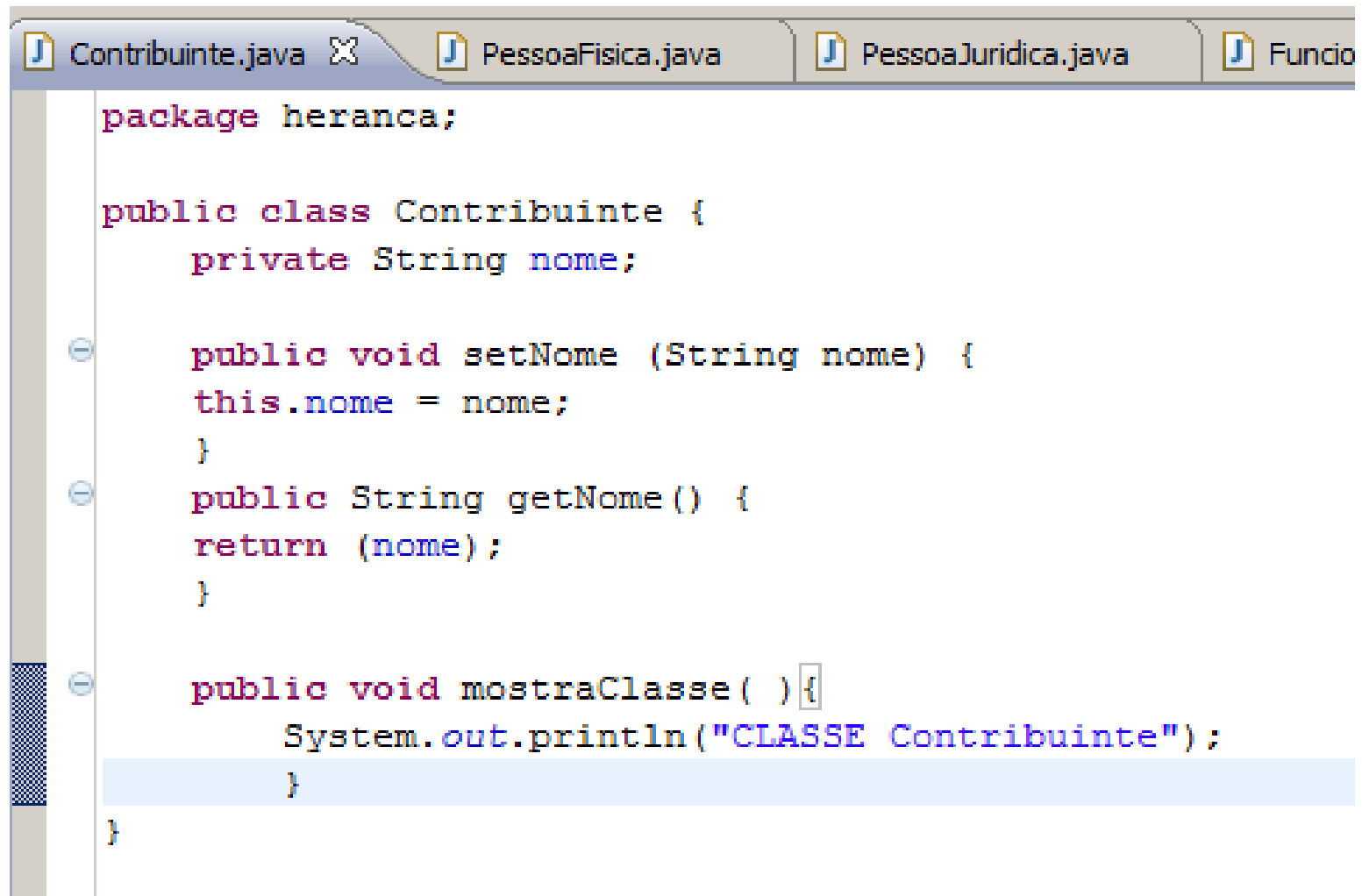
ITE - 003

profº Mauricio Conceição Mario

Polimorfismo

Polimorfismo é o princípio pelo qual duas ou mais **classes derivadas** de uma mesma **superclasse** podem invocar **métodos que têm a mesma identificação** (assinatura) mas **comportamentos distintos**, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. Esse mecanismo é fundamental na programação orientada a objetos, permitindo definir funcionalidades que operem genericamente com objetos, abstraindo-se de seus detalhes particulares quando esses não forem necessários.

Polimorfismo



```
package heranca;

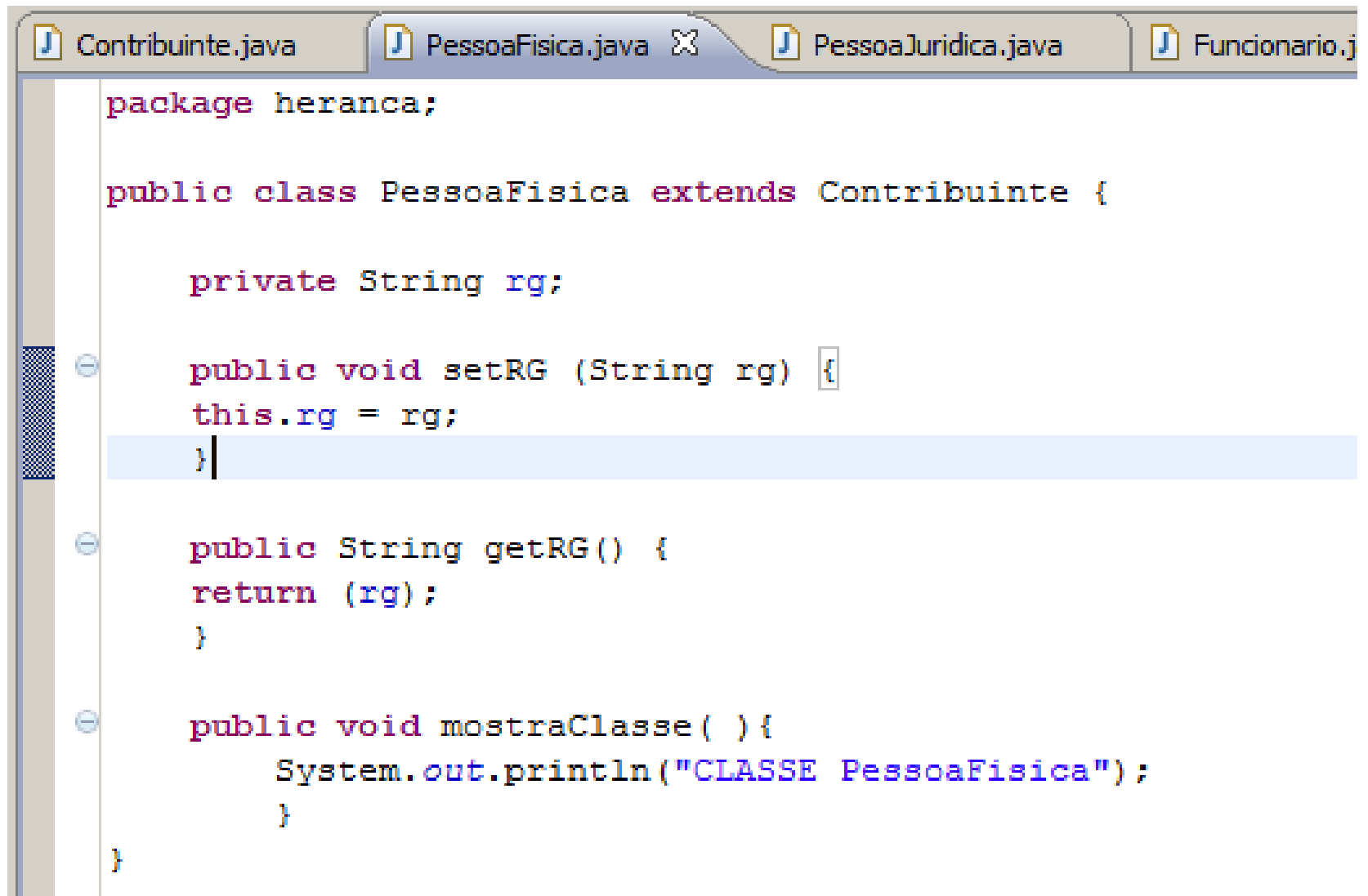
public class Contribuinte {
    private String nome;

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

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

    public void mostraClasse( ){
        System.out.println("CLASSE Contribuinte");
    }
}
```

Polimorfismo



```
Contribuinte.java PessoaFisica.java X PessoaJuridica.java Funcionario.j
package heranca;

public class PessoaFisica extends Contribuinte {

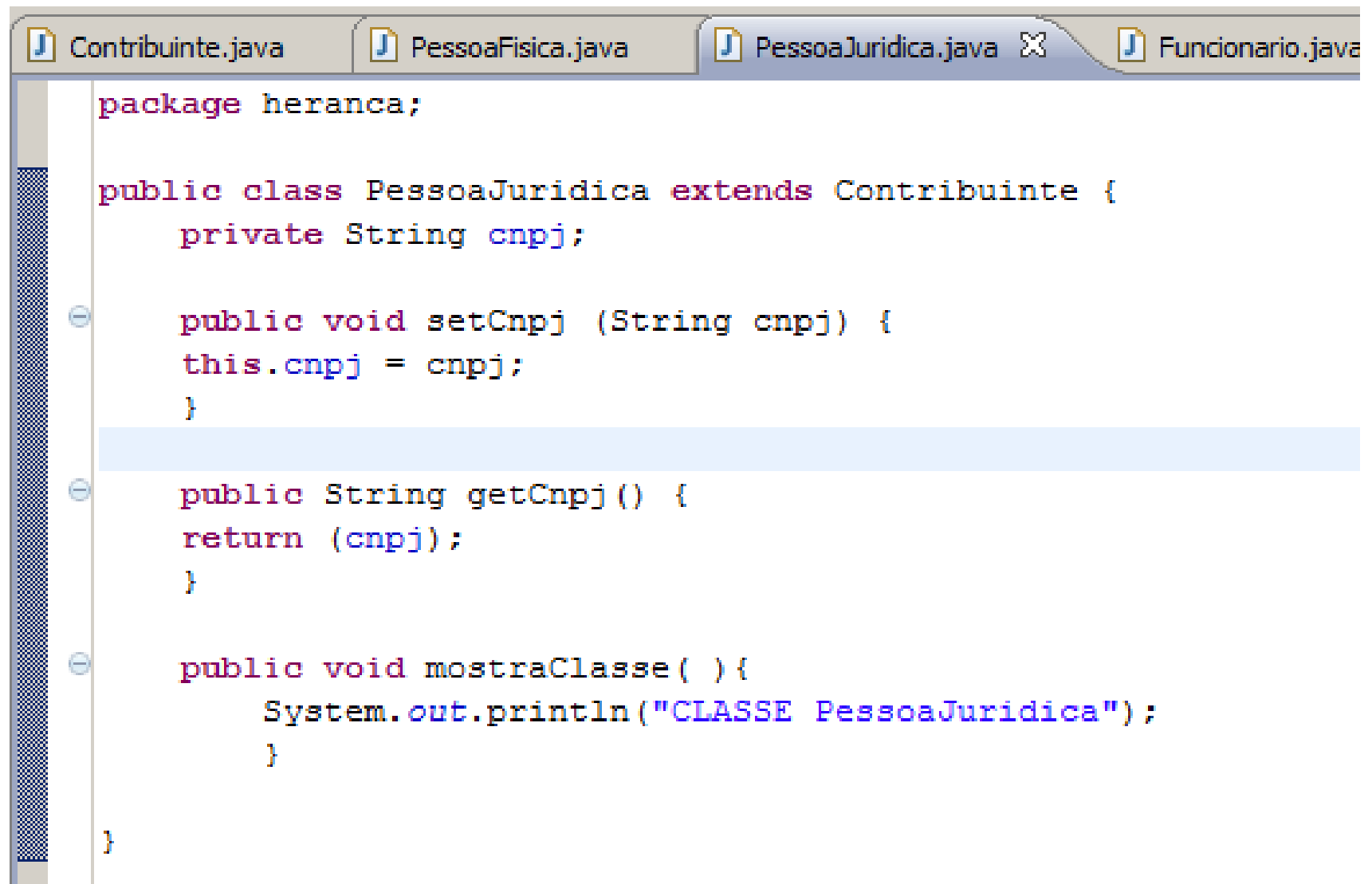
    private String rg;

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

    public String getRG() {
        return (rg);
    }

    public void mostraClasse( ){
        System.out.println("CLASSE PessoaFisica");
    }
}
```

Polimorfismo



```
Contribuinte.java  PessoaFisica.java  PessoaJuridica.java  Funcionario.java

package heranca;

public class PessoaJuridica extends Contribuinte {
    private String cnpj;

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

    public String getCnpj() {
        return (cnpj);
    }

    public void mostraClasse( ){
        System.out.println("CLASSE PessoaJuridica");
    }
}
```

Polimorfismo

Contribuinte.java PessoaFisica.java PessoaJuridica.java Funcionario.java X

```
package heranca;

public class Funcionario extends PessoaFisica {
    private String cartao;

    public void setCartao (String cartao) {
        this.cartao = cartao;
    }

    public String getCartao() {
        return (cartao);
    }

    public void mostraClasse( ){
        System.out.println("CLASSE Funcionario");
    }
}
```

Polimorfismo

```
package heranca;
```

```
import javax.swing.*;
```

```
public class Polimorfismo {
```

```
    public static void main(String args[]){
```

```
        /*declaração de objeto do tipo da superclasse  
        inicializado como "vazio"  
        */
```

```
        Contribuinte pessoa = null;
```

```
        //leitura de variável que definirá como o objeto será instanciado  
        int tipo = Integer.parseInt(JOptionPane.showInputDialog  
            ("digitar opção: valor numérico entre 1 e 4"));
```

```
        //definição de como o objeto será instanciado
```

```
        switch(tipo){
```

```
        case 1:
```

```
            | pessoa = new Contribuinte(); break; //instância pela superclasse
```

```
        case 2:
```

```
            pessoa = new PessoaFisica(); break; //instância pela classe PessoaFisica
```

```
        case 3:
```

```
            pessoa = new PessoaJuridica(); break; //instância pela classe PessoaJuridica
```

```
        case 4:
```

```
            pessoa = new Funcionario(); break; //instância pela classe Funcionario
```

```
            default: {System.out.println("tipo não pertence à família");
```

```
                System.exit(0);}
```

```
        } //fim do switch()
```

```
        pessoa.mostraClasse();
```

```
    } //fim do void main()
```

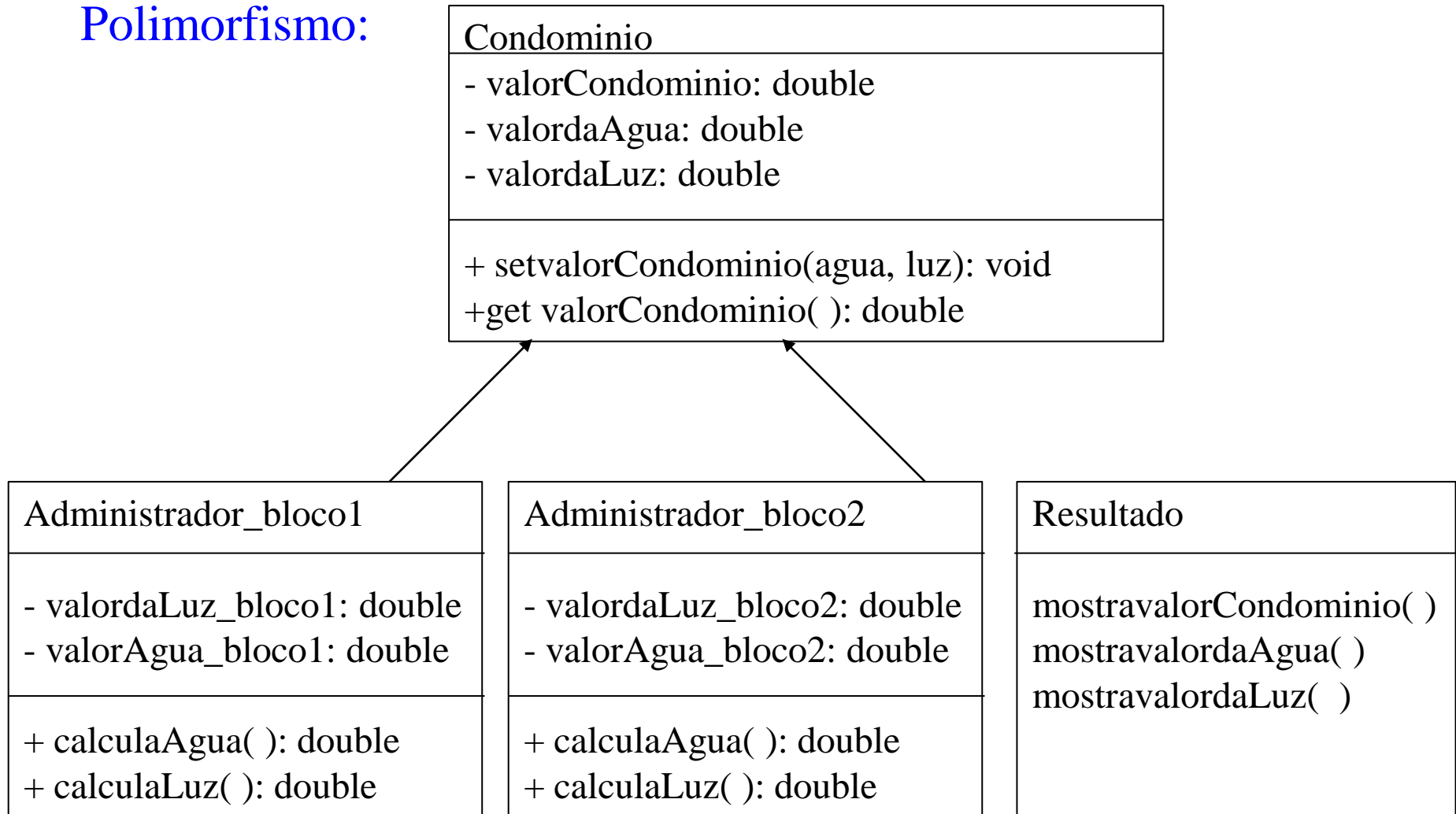
```
}
```

Exercício 9:

Construir as classes Contribuinte, PessoaFisica, PessoaJuridica, Funcionario e Polimorfismo. Executar Polimorfismo com as diferentes opções de tipo e verificar os resultados.

Exercício 10: (para ser entregue como parte da lista do bimestre, juntamente com os primeiros exercícios de herança).

Implementar a aplicação conforme o diagrama de classes e utilizar os conceitos de Sobrecarga de Métodos e Polimorfismo:



Proposta de aplicativo para calcular área utilizando sobrecarga de métodos:

```
Aluno.java  CentralSubject.java  Mensagem.java  ObserverMain.java  Area_Sobrecarga.java X

public class Area_Sobrecarga {

    public static void main (String args[]){

        System.out.println("área do quadrado..." + area(3));
        System.out.println("área do retângulo..." + area(3,2));
        System.out.println("área do cubo..." + area(3,2,5));

    }

    public static double area(int x){
        return x*x;
    }

    public static double area(int x, int y){
        return x*y;
    }

    public static double area(int x, int y, int z){
        return x*y*z;
    }

}
```

```
Problems  @ Javadoc  Declaration  Console X

<terminated> Area_Sobrecarga [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (24/03/2015 07:53:49)

área do quadrado...9.0
área do retângulo...6.0
área do cubo...30.0
```