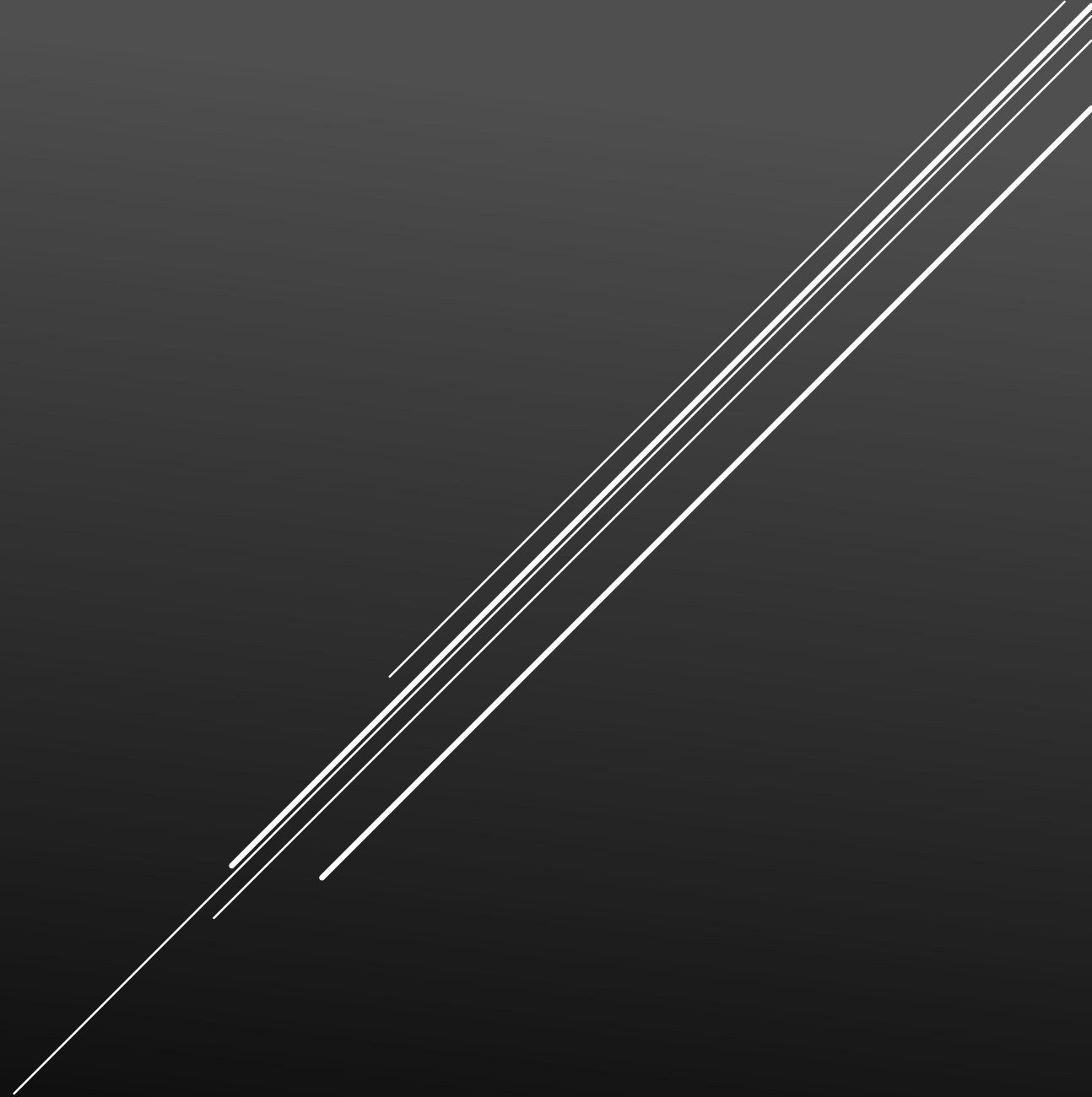


INTRODUÇÃO AO POLIMORFISMO





Polimorfismo



Poli = Muitos



Morfismo = formas



O Polimorfismo é um mecanismo por meio do qual selecionamos as funcionalidades utilizadas de forma dinâmica por um programa no decorrer de sua execução.



Com o Polimorfismo, os mesmos atributos e objetos podem ser utilizados em objetos distintos, porém, com implementações lógicas diferentes.

Tipos básicos de polimorfismo

- Polimorfismo Estático ou Sobrecarga
- Polimorfismo Dinâmico ou Sobreposição

1) Sobreposição: é um conceito do polimorfismo que nos permite reescrever um método, ou seja, podemos reescrever nas classes filhas métodos criados inicialmente na classe pai, os métodos que serão sobrepostos, diferentemente dos sobrecarregados, devem possuir o mesmo nome, tipo de retorno e quantidade de parâmetros do método inicial, porém o mesmo será implementado com especificações da classe atual, podendo adicionar um algo a mais ou não.

2) Sobrecarga: pode ser classificada como polimorfismo de inclusão. Quando um método sobrescreve um método herdado de uma classe, temos uma sobrescrita de método. Este método de sobrescrita tem que ser idêntico ao método da classe herdada, ou seja, eles precisam ter o mesmo nome, valor de retorno e argumentos. Portanto, temos que uma classe filha fornece apenas uma nova implementação para o método herdado e não um novo método.

Assinatura do método:

Quantidades e os tipos dos parâmetros

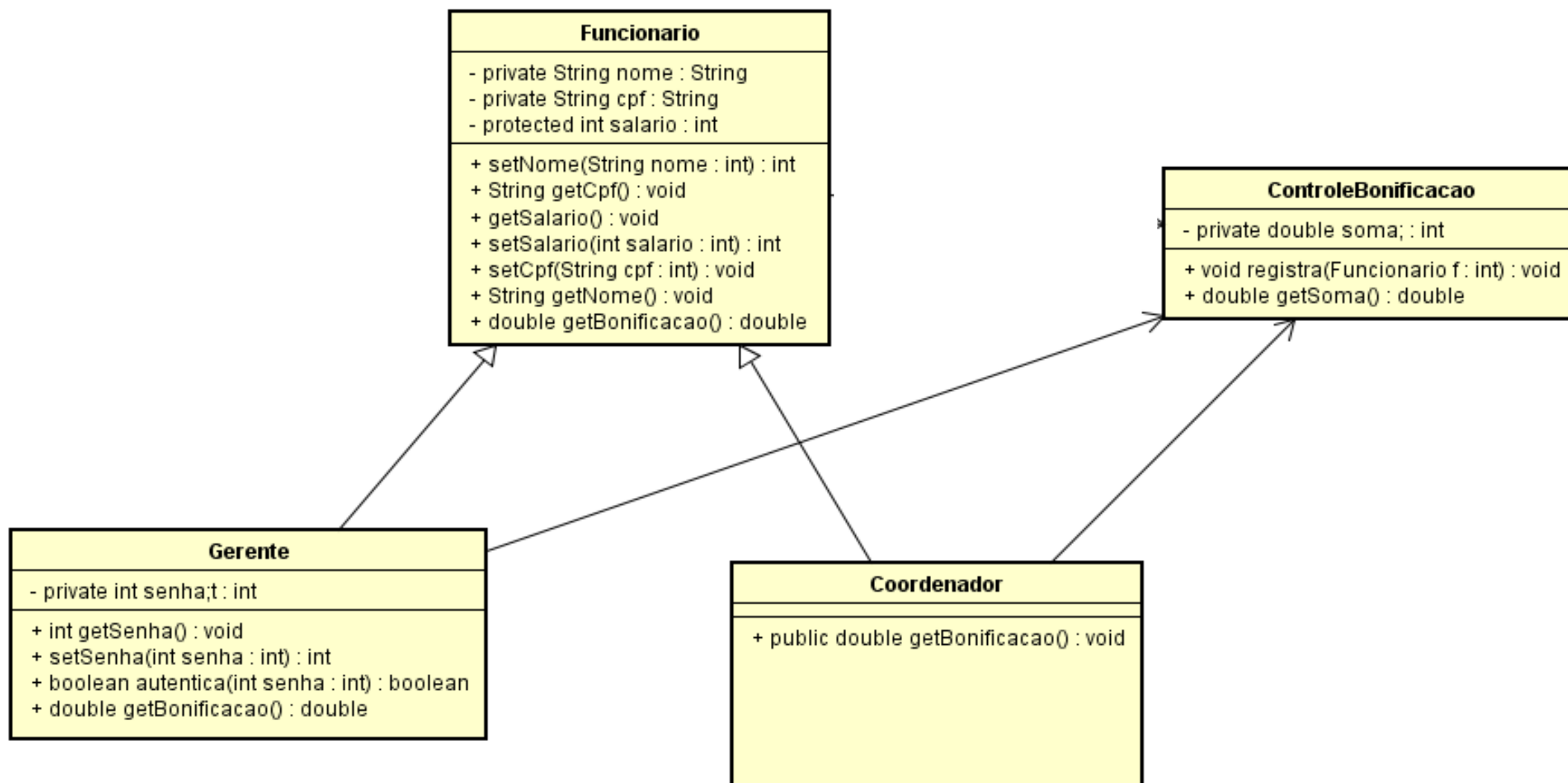
Exemplos

```
public function calcMedia(n1:Real,n2:Real):Real
```

```
public function calcMedia(v1:inteiro,v2:inteiro):Inteiro
```

```
public function calcMedia(bim:inteiro,n1:Real,n2:Real):Real
```

Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.



```
2 public class Funcionario {
3
4     private String nome;
5     private String cpf;
6     protected int salario;
7
8     public String getNome() {
9         return nome;
10    }
11
12    public void setNome(String nome) {
13        this.nome = nome;
14    }
15
16    public String getCpf() {
17        return cpf;
18    }
19
20    public void setCpf(String cpf) {
21        this.cpf = cpf;
22    }
23
24    public int getSalario() {
25        return salario;
26    }
27
28    public void setSalario(int salario) {
29        this.salario = salario;
30    }
31
32    // Método que retorna o valor da bonificação do funcionario
33    public double getBonificacao() {
34
35        return salario * 0.10;
36
37    }
38
39 }
40
```

```
2 public class Gerente extends Funcionario {
3
4     private int senha;
5
6     public int getSenha() {
7         return senha;
8     }
9
10    public void setSenha(int senha) {
11        this.senha = senha;
12    }
13
14    public boolean autentica(int senha) {
15
16        if (this.senha == senha) {
17
18            return true;
19        } else
20            return false;
21
22    }
23
24    // Método que retorna o valor da bonificação do gerente
25    @Override
26
27    public double getBonificacao() {
28
29        return super.salario * 0.15;
30    }
31
32 }
33
```

```
2 public class Coordenador extends Funcionario {  
3     // Método que retorna o valor da bonificação do funcionario  
4     public double getBonificacao() {  
5  
6         return 2000;  
7     }  
8 }  
9  
10 }  
11
```



```
2 //Classe Para controlar os gastos da empresa com Bonificação
3 public class ControleBonificacao {
4     //atributo soma, server para soma cada uma das bonificações
5     private double soma;
6
7
8     //Método que recebe como parametro um objeto do tipo funcionario
9     public void registra(Funcionario f) {
10         //executando o getBonificação especifico de cada funcionario e armazenado o valor dentro
11         double boni = f.getBonificacao();
12
13         // somando a variavel boni dentro da variavel soma
14         this.soma = this.soma + boni;
15     }
16
17
18     //método getSoma();
19     public double getSoma() {
20
21         return this.soma;
22     }
23
24 }
25
```

```

30 public static void main(String[] args) {
4     // TODO Auto-generated method stub
5
6     Funcionario f1 = new Funcionario();
7     f1.setNome("Doni");
8     f1.setCpf("44444.3333.4");
9     f1.setSalario(2500);
10    System.out.println(f1.getNome());
11    System.out.println(f1.getCpf());
12    System.out.println(f1.getSalario());
13
14    Gerente g1 = new Gerente();
15    g1.setNome("PEDRO");
16    g1.setCpf("32323232");
17    g1.setSalario(2500);
18    g1.setSenha(2222);
19    System.out.println("-----");
20    System.out.println(g1.getNome());
21    System.out.println(g1.getCpf());
22    System.out.println(g1.getSalario());
23    System.out.println(g1.autentica(2222));
24    Coordenador c1 = new Coordenador();
25    c1.setNome("Coord");
26    c1.setCpf("32323232");
27    c1.setSalario(3600);
28    System.out.println("-----");
29    System.out.println(c1.getNome());
30    System.out.println(c1.getCpf());
31    System.out.println(c1.getSalario());
32
33    System.out.println("-----");
34    ControleBonificacao control = new ControleBonificacao();
35
36    control.registra(f1);
37
38    control.registra(g1);
39
40    control.registra(c1);
41
42    System.out.println("Gastos com bonificação "+control.getSoma());
43
44
45
46 }
47
48 }

```

<terminated> Principal (26) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (22 de jul de 2020 13:43:50 – 13:43:54)

Doni

44444.3333.4

2500

PEDRO

32323232

2500

true

Coord

32323232

3600

Gastos com bonificação 2625.0

AGORA É A SUA VEZ...

