

Classe Abstrata

Carlos Arruda Baltazar
UNIP – Cidade Universitária

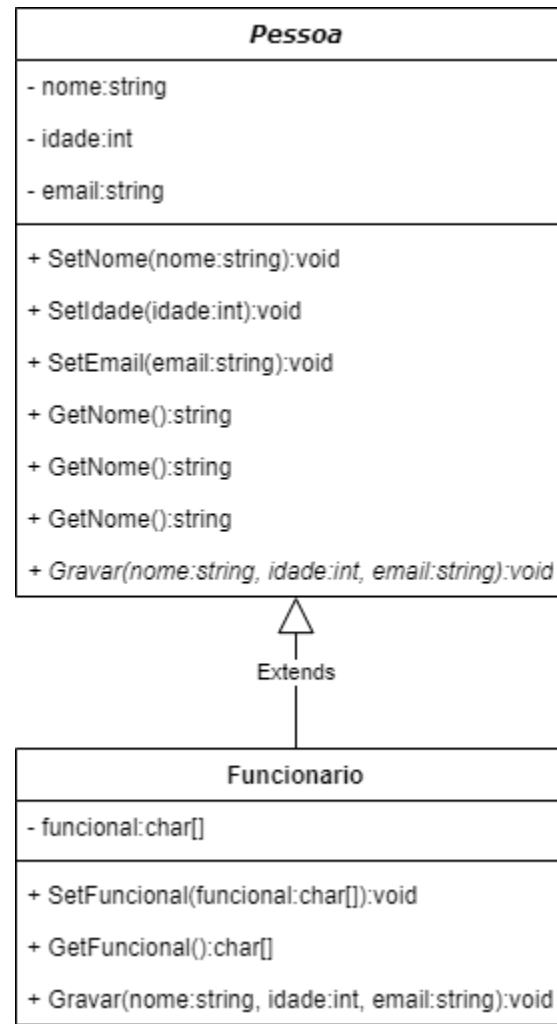
As classes abstratas são as que não permitem realizar qualquer tipo de instância. São classes feitas especialmente para serem modelos para suas classes derivadas. As classes derivadas, via de regra, deverão sobrescrever os métodos para realizar a implementação dos mesmos. As classes derivadas das classes abstratas são conhecidas como classes concretas.

Como medida de segurança, as classes abstratas somente podem ser estendidas, sendo que a criação de um objeto a partir da mesma é um procedimento evitado. Além disso, caso um ou mais métodos abstratos estejam presentes nessa classe abstrata, a classe filha será, então, forçada a definir tais métodos, pois, caso contrário, a classe filha também se tornará abstrata.

A funcionalidade dos métodos abstratos que são herdados pelas classes filha normalmente é atribuída de acordo com o objetivo ou o propósito dessas classes. É possível, porém, não atribuirmos uma funcionalidade a esses métodos abstratos. Neste caso, faz-se necessário, pelo menos, declarar tais métodos.

Os métodos abstratos estão presentes somente em classes abstratas, e são aqueles que não possuem implementação.

```
public abstract void metodo1();
```

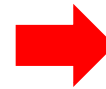


```
package Pck_teste2;

public abstract class Pessoa
{
    private String nome;
    private int idade;
    private String email;

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public int getIdade() {
        return idade;
    }
    public void setIdade(int idade) {
        this.idade = idade;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }

    public abstract void Gravar(String nome, int idade, String email);
}
```



```
package Pck_teste2;

public class Funcionario extends Pessoa
{
    private char[] funcional;

    private char[] getFuncional() {
        return funcional;
    }

    private void setFuncional(char[] funcional) {
        this.funcional = funcional;
    }

    public void Gravar(String nome, int idade, String email)
    {
        super.setNome(nome);
        super.setIdade(idade);
        super.setEmail(email);
    }
}
```

Para estender o exemplo anterior, vamos imaginar que na empresa que está comprando esse sistema existem: diretores, gerentes, secretarias e etc. Contudo, independente do cargo e apesar de suas particularidades, todos são funcionários da empresa. Então vamos incluir um sistema de bonificações na classe Funcionário.


```
package Pck_teste2;

public class Funcionario extends Pessoa
{
    private char[] funcional;
    private double salario;

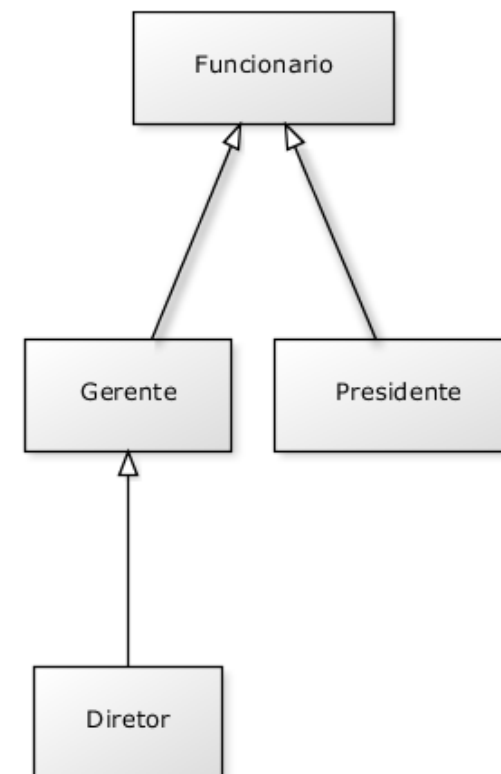
    private char[] getFuncional() {
        return funcional;
    }

    private void setFuncional(char[] funcional) {
        this.funcional = funcional;
    }

    public void Gravar(String nome, int idade, String email)
    {
        super.setNome(nome);
        super.setIdade(idade);
        super.setEmail(email);
    }

    public double GetBonificacao()
    {
        return this.salario * 1.2;
    }
}
```

Porém, ao estabelecermos a política de bonificação dessa forma, a mesma bonificação será aplicado independente do cargo. Pois independente do cargo a classe funcionário seria herdada. Desta forma temos:



Então, a melhor alternativa seria transformar a classe funcionário também em uma classe abstrata e o método de bonificação em um método abstrato.

```
package Pck_teste2;

public abstract class Funcionario extends Pessoa
{
    private char[] funcional;
    private double salario;

    public char[] getFuncional() {
        return funcional;
    }

    public void setFuncional(char[] funcional) {
        this.funcional = funcional;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public void Gravar(String nome, int idade, String email)
    {
        super.setNome(nome);
        super.setIdade(idade);
        super.setEmail(email);
    }

    public abstract double GetBonificacao();
}
```

```
package Pck_teste2;

public abstract class Funcionario extends Pessoa
{
    private char[] funcional;
    private double salario;

    public char[] getFuncional() {
        return funcional;
    }

    public void setFuncional(char[] funcional) {
        this.funcional = funcional;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public void Gravar(String nome, int idade, String email)
    {
        super.setNome(nome);
        super.setIdade(idade);
        super.setEmail(email);
    }

    public abstract double GetBonificacao();
}
```

```
package Pck_teste2;
```

```
public class Secretaria extends Funcionario
{
    private static double taxaBonificacao = 1.4;

    public double GetBonificacao()
    {
        return super.getSalario() * this.taxaBonificacao;
    }
}
```

```
public class Presidente extends Funcionario
{
    private static double taxaBonificacao = 2;
    @Override
    public double GetBonificacao() {
        // TODO Auto-generated method stub
        return super.getSalario() * this.taxaBonificacao + 3000;
    }
}
```

```
package Pck_teste2;

public abstract class Funcionario extends Pessoa
{
    private char[] funcional;
    private double salario;

    public char[] getFuncional() {
        return funcional;
    }

    public void setFuncional(char[] funcional) {
        this.funcional = funcional;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public void Gravar(String nome, int idade, String email)
    {
        super.setNome(nome);
        super.setIdade(idade);
        super.setEmail(email);
    }

    public abstract double GetBonificacao();
}
```



```
package Pck_teste2;

public class Gerente extends Funcionario
{
    private static double taxaBonificacao = 1.4;
    private String departamento;

    public String getDepartamento() {
        return departamento;
    }

    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }

    @Override
    public double GetBonificacao() {
        // TODO Auto-generated method stub
        return super.getSalario() * this.taxaBonificacao + 1000;
    }
}
```

```
package Pck_teste2;

public class Gerente extends Funcionario
{
    private static double taxaBonificacao = 1.4;
    private String departamento;

    public String getDepartamento() {
        return departamento;
    }
    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }

    @Override
    public double GetBonificacao() {
        // TODO Auto-generated method stub
        return super.getSalario() * this.taxaBonificacao + 1000;
    }
}
```



```
package Pck_teste2;

public class Diretor extends Gerente
{
    private static double taxaBonificacao = 1.8;

    public double GetBonificacao()
    {
        return super.getSalario() * this.taxaBonificacao + 1000;
    }
}
```

OBRIGADO