

09

Mão na massa: Trabalhando com interfaces

- 1) Se não tiver aberto ainda, abra o projeto sobre **Funcionario**.
- 2) Crie uma classe chamada **SistemaInterno** que tenha como atributo uma senha e implemente o método `autentica` conforme abaixo:

```
public class SistemaInterno{  
  
    private int senha = 2222;  
  
    public void autentica(Gerente g){  
  
        boolean autenticou = g.autentica(this.senha);  
  
        if(autenticou){  
            System.out.println("Pode entrar no sistema");  
        } else {  
            System.out.println("Não pode entrar no sistema");  
        }  
    }  
}
```

- 3) Crie uma classe de testes chamada **TesteSistema** com o método `main` já gerado.
- 4) Instancie um gerente, utilize o método `setSenha` e coloque a senha correta, além disso, instancie o **SistemaInterno** e tente autenticar.

O código abaixo deve estar dentro do método `main`:

```
Gerente g = new Gerente();  
g.setSenha(2222);  
  
SistemaInterno si = new SistemaInterno();  
  
si.autentica(g);
```

- 5) Crie uma classe **Administrador** que deve herdar da classe **Funcionario**. Perceba que o Eclipse já nos entrega inclusive o método `getBonificacao`
- 6) Crie uma classe abstrata **FuncionarioAutenticavel** que herda de **Funcionario**
- 7) Complete o código de **FuncionarioAutenticavel** com o código abaixo:

```
public abstract class FuncionarioAutenticavel extends Funcionario{  
  
    private int senha;
```

```

public void setSenha(int Senha){
    this.senha = senha;
}

public boolean autentica(int senha){
    if(this.senha == senha){
        return true;
    } else {
        return false;
    }
}
}

```

8) Modifique o código de **Gerente** fazendo com que herde de **FuncionarioAutenticavel**

```

public class Gerente extends FuncionarioAutenticavel{

//resto do código

}

```

9) Em **SistemaInterno** troque a referência do método **autentica** para que receba um **FuncionarioAutenticavel**, veja abaixo:

```

public class SistemaInterno{

private int senha = 2222;

public void autentica(FuncionarioAutenticavel fa){

    boolean autenticou = fa.autentica(this.senha);

    if(autenticou){
        System.out.println("Pode entrar no sistema");
    } else {
        System.out.println("Não pode entrar no sistema");
    }
}
}

```

10) Perceba que se precisarmos que um cliente seja autenticável, teremos um problema, já que a classe **Cliente** precisaria herdar de **FuncionarioAutenticavel** e isso acarretaria em um cliente ter bonificação já que a classe **FuncionarioAutenticavel** herda de **Funcionario**

11) Para resolver isso, faça uso das **Interfaces**, veja a interface **Autenticavel** abaixo:

```

public abstract interface Autenticavel{

    public abstract void setSenha(int senha);

    public abstract boolean autentica(int senha);
}

```

12) Na classe **Cliente** utilize a palavra reservada **implements** para utilizar a interface criada, veja abaixo:

```
public class Cliente implements Autenticavel{  
  
}
```

13) Complete o código da classe **Cliente**, implementando os métodos que faltam:

```
public class Cliente implements Autenticavel{  
  
    private int senha;  
  
    @Override  
    public void setSenha(int senha){  
        this.senha = senha;  
    }  
  
    @Override  
    public boolean autentica(int senha){  
  
        if(this.senha == senha){  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
}
```

14) Faça com que a classe **Funcionario** implemente a interface, não se esqueça de implementar os métodos e declarar o atributo privado senha!

```
public class Funcionario implements Autenticavel{  
  
    private int senha;  
  
    public double getBonificacao(){  
        System.out.println("Chamando o método de bonificação do GERENTE");  
        return super.getSalario();  
    }  
  
    @Override  
    public void setSenha(int senha){  
        this.senha = senha;  
    }  
  
    @Override  
    public boolean autentica(int senha){  
  
        if(this.senha == senha){  
            return true;  
        }  
    }  
}
```

```
    } else {
        return false;
    }
}
```

15) No **SistemaInterno** modifique o método autentica para que agora receba um **Autenticavel**

```
public class SistemaInterno{

    private int senha = 2222;

    public void autentica(Autenticavel fa){

        boolean autenticou = fa.autentica(this.senha);

        if(autenticou){
            System.out.println("Pode entrar no sistema");
        } else {
            System.out.println("Não pode entrar no sistema");
        }
    }
}
```

16) Repare que agora podemos usar **Autenticavel** como um tipo, veja o exemplo abaixo:

```
Autenticavel referencia = new Gerente();

Autenticavel referencia = new Cliente();

Autenticavel referencia = new Administrador();
```

17) Faça com que as classes **Gerente** e **Administrador** implementem a interface **Autenticavel**, não se esqueça do **atributo privado senha**

18) Teste tudo e veja se está tudo ok!

