

Programação Orientada a Objetos

classes abstratas e interfaces

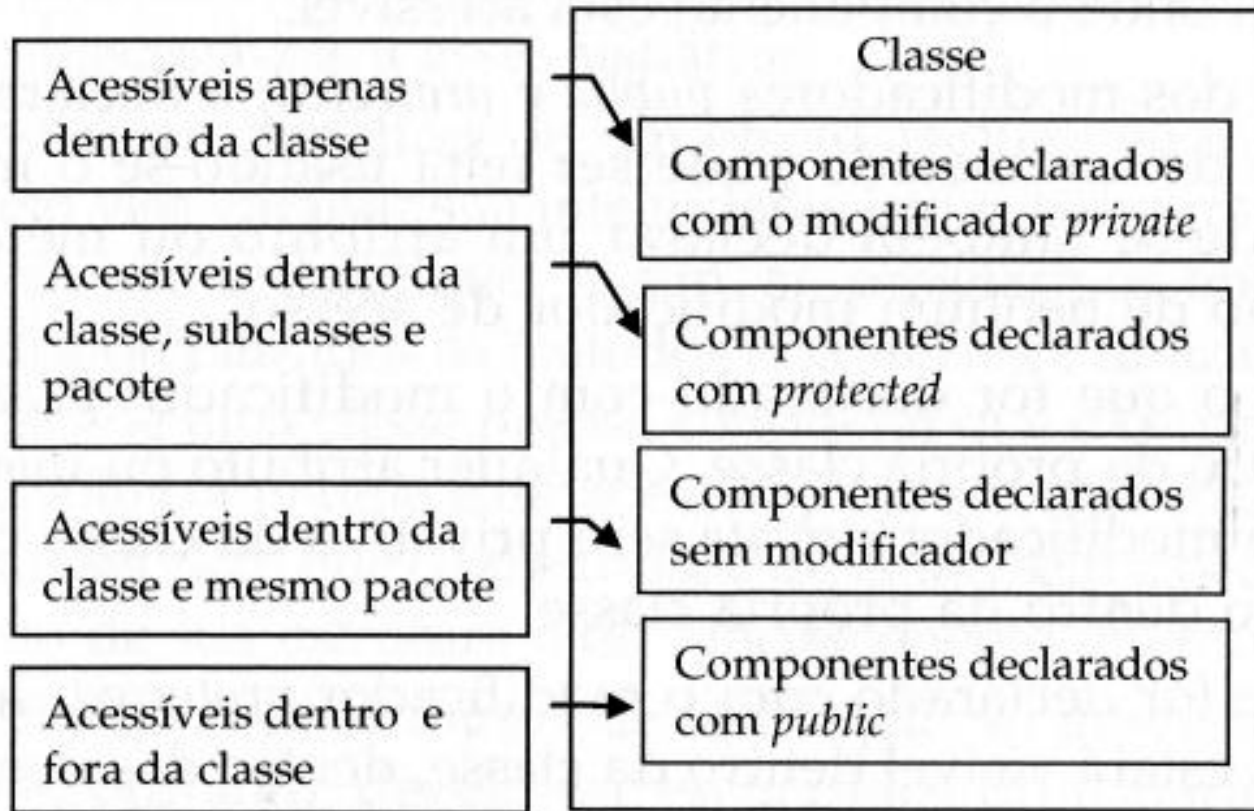
profº Mauricio Conceição Mario

Classe Abstrata

- Classe abstrata não permite a geração de instância a partir da mesma, ou seja, não permite a criação de objetos relacionados a classe.
- Uma classe abstrata serve de base para a elaboração de outras classes, ou seja, é um modelo para elaboração de outras classes, não de objetos.

(Furgeri, 2013)

Encapsulamento/Modificadores de acesso



Encapsulamento / Modificadores de acesso.

(Boratti, 2007)

Exemplo do uso de classe abstrata.

📁 interface_abstrata

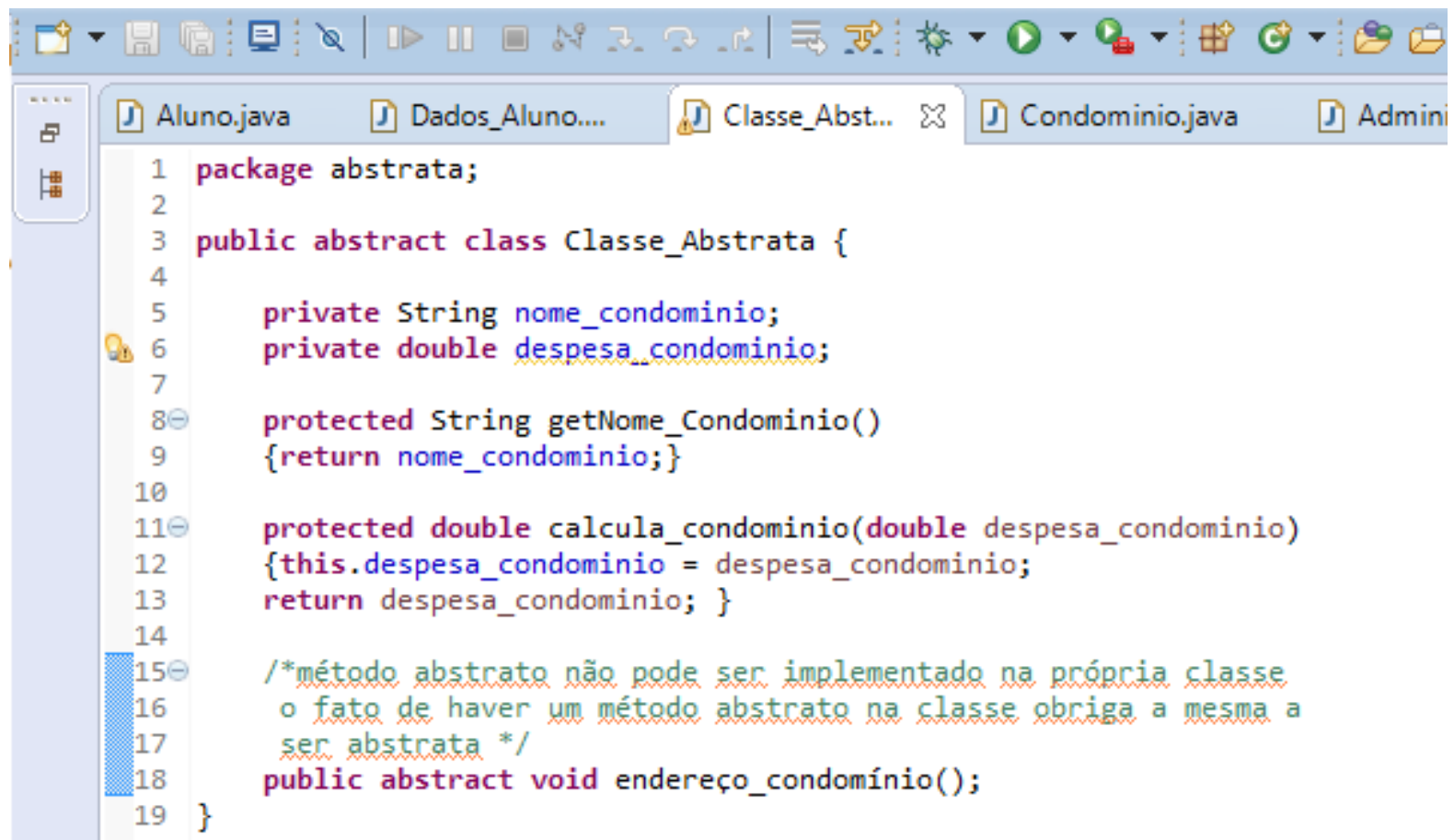
📁 src

📁 abstrata

▶ Administra_Condominio.java

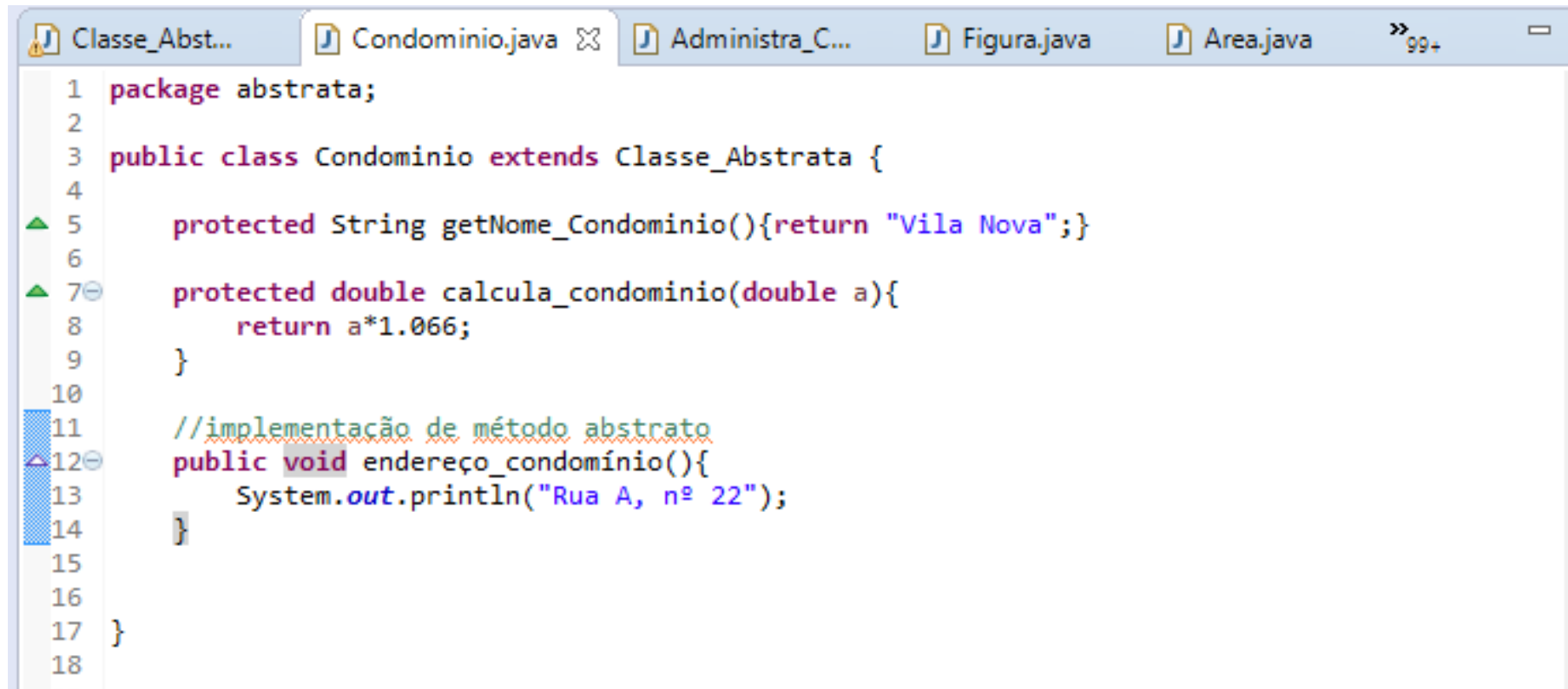
▶ Classe_Abstrata.java

▶ Condominio.java



```
1 package abstrata;
2
3 public abstract class Classe_Abstrata {
4
5     private String nome_condominio;
6     private double despesa_condominio;
7
8     protected String getNome_Condominio()
9     {return nome_condominio;}
10
11     protected double calcula_condominio(double despesa_condominio)
12     {this.despesa_condominio = despesa_condominio;
13     return despesa_condominio; }
14
15     /*método abstrato não pode ser implementado na própria classe
16     o fato de haver um método abstrato na classe obriga a mesma a
17     ser abstrata */
18     public abstract void endereco_condomínio();
19 }
```

Exemplo do uso de classe abstrata.



The screenshot shows an IDE window with several tabs: 'Classe_Abst...', 'Condominio.java', 'Administra_C...', 'Figura.java', and 'Area.java'. The 'Condominio.java' tab is active, displaying the following Java code:

```
1 package abstrata;
2
3 public class Condominio extends Classe_Abstrata {
4
5     protected String getNome_Condominio(){return "Vila Nova";}
6
7     protected double calcula_condominio(double a){
8         return a*1.066;
9     }
10
11     //implementação de método abstrato
12     public void endereço_condomínio(){
13         System.out.println("Rua A, nº 22");
14     }
15
16
17 }
18
```

Exemplo do uso de classe abstrata.

```
Aluno.java  Dados_Aluno...  Classe_Abst...  Condominio.java  Administra_C... ✕

1 package abstrata;
2
3 public class Administra_Condominio {
4
5     public static void main(String args[]){
6
7         Condominio b = new Condominio();
8         Classe_Abstrata c = new Condominio(); ←
9
10        //variável do tipo final equivale a uma constante
11        final double a = 234.08;
12
13        System.out.println("NOME CONDOMINIO " + b.getNome_Condominio());
14        System.out.println("VALOR DO CONDOMINIO = R$ " + b.calcula_condominio(a));
15
16        c.endereço_condomínio();
17
18        /*variável final não pode ser alterada
19        a = a + 1.0; "erro"*/
20
21        /*um método declarado como final não pode ser sobrescrito
22        uma classe declarada como final não pode ser herdada*/
23
24    }
25
26 }
```

Console ✕

<terminated> Administra_Condominio [Java Application] C:\Arquivos de Programas\Java\jdk1.8.0_45\bin\javaw.exe (15 de out de 2017)

NOME CONDOMINIO Vila Nova

VALOR DO CONDOMINIO = R\$ 249.52928000000003

Rua A, nº 22

Interface

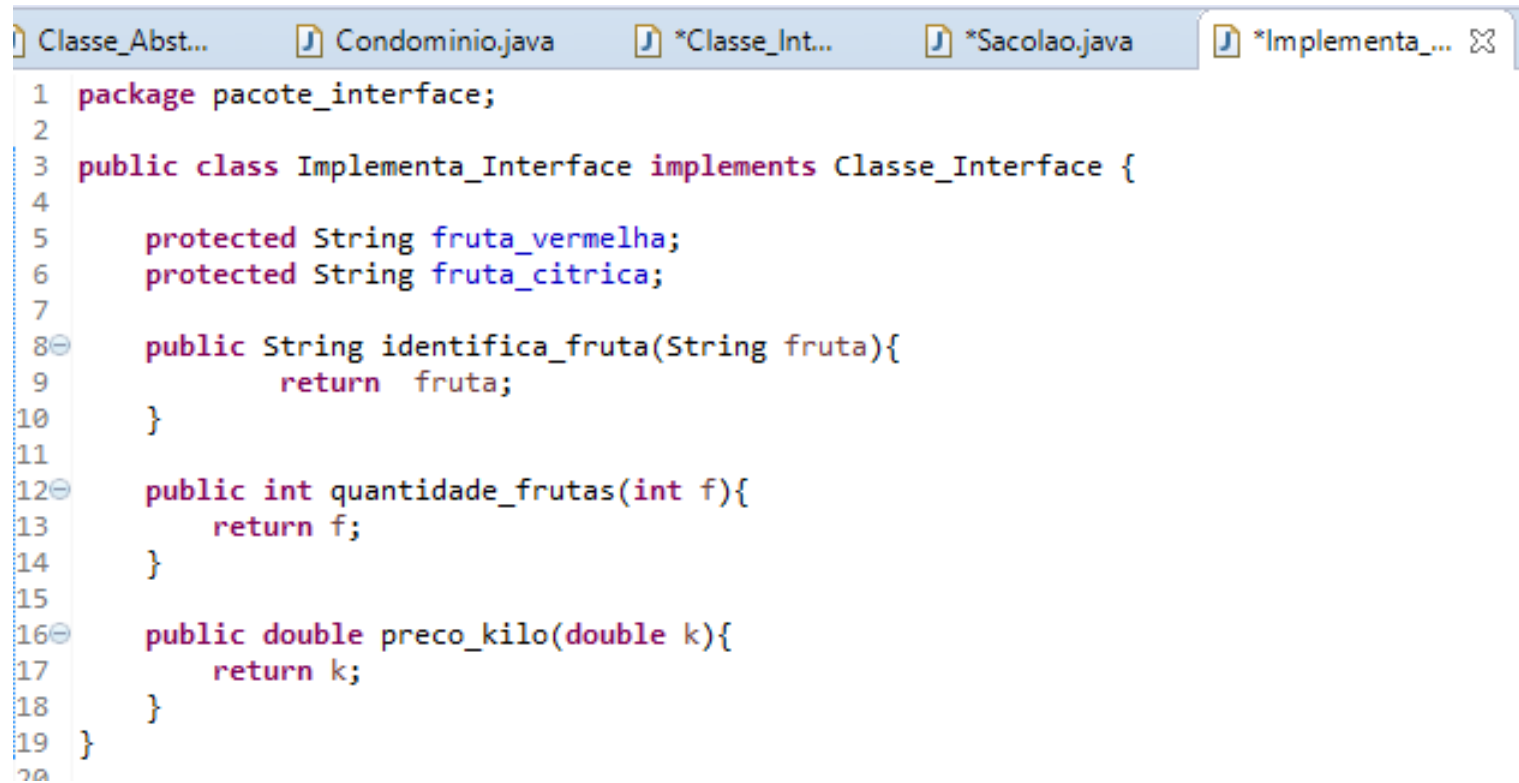
- Na orientação a objetos uma interface se refere a um mecanismo que determina uma série de requisitos que uma classe deva conter. Diz-se então, que esta classe implementa esta interface. Uma interface é similar a uma classe abstrata, porém não permite a implementação de métodos, contendo apenas a especificação destes.

Exemplo do uso de interface.

- pacote_interface
 - Classe_Interface.java
 - Implementa_Interface.java
 - Sacolao.java

```
Classe_Abst...  Condominio.java  *Classe_Int...  *Sacolao
1 package pacote_interface;
2
3 public interface Classe_Interface {
4
5     public String identifica_fruta(String fruta);
6     public int quantidade_frutas(int f);
7     public double preco_kilo(double k );
8 }
9
10
```


Exemplo do uso de interface.



The image shows a screenshot of an IDE with several tabs at the top: 'Classe_Abst...', 'Condominio.java', '*Classe_Int...', '*Sacolao.java', and '*Implementa_...' (which is the active tab). The code in the active tab is as follows:

```
1 package pacote_interface;
2
3 public class Implementa_Interface implements Classe_Interface {
4
5     protected String fruta_vermelha;
6     protected String fruta_citrica;
7
8     public String identifica_fruta(String fruta){
9         return fruta;
10    }
11
12    public int quantidade_frutas(int f){
13        return f;
14    }
15
16    public double preco_kilo(double k){
17        return k;
18    }
19 }
```

Exemplo do uso de interface.

```
Classe_Abst...  Condominio.java  *Classe_Int...  *Sacolao.java  *Implementa...  » 33

1 package pacote_interface;
2
3 public class Sacolao {
4
5     public static void main(String args[]){
6         Implementa_Interface a = new Implementa_Interface();
7         a.fruta_citrica = "laranja";
8         a.fruta_vermelha = "maçã";
9
10        System.out.println ("Fruta Cítrica: " + a.identifica_fruta(a.fruta_citrica));
11        System.out.println ("Fruta Vermelha: " + a.identifica_fruta(a.fruta_vermelha));
12        System.out.println ("quantidade de frutas = " + a.quantidade_frutas(12));
13        System.out.println ("preco kilo da fruta: R$ " + a.preco_kilo(5.66));
14    }
15
16 }
17
```

Problems Javadoc Declaration Console Debug

<terminated> Sacolao [Java Application] C:\Arquivos de Programas\Java\jdk1.8.0_45\bin\javaw.exe (27 de out de 2016 11:14:21)

Fruta Cítrica: laranja
Fruta Vermelha: maçã
quantidade de frutas = 12
preco kilo da fruta: R\$ 5.66

- **Exercício classe abstrata:** Construir e executar o exemplo de aplicação com classe abstrata.
- **Exercício interface:** Construir e executar o exemplo de aplicação com interface.

Referências Bibliográficas

- Boratti, I. C.. “*Programação Orientada a Objetos em Java*”. Editora Visual Books. ISBN 85-7502-199-0. Florianópolis, 2007.
- Furgeri, S.. “*Modelagem de Sistemas Orientados a Objetos*”. Editora Érica. ISBN 978-85-365-0461-2. São Paulo, 2013.