



## 4. POLIMORFISMO

PROGRAMAÇÃO ORIENTADA AOS OBJETOS

Desenvolvido por:

Carlos Urbano  
Catarina Reis  
José Magno  
Marco Ferreira  
Ricardo Antunes

4.1. CONCEITO

4.2. EXEMPLO

4.3. CASOS POSSÍVEIS

4.4. VARIÁVEIS POLIMÓRFICAS

# Polimorfismo

- significa muitas formas

proveniente do grego

- uma função com múltiplas formas de implementação

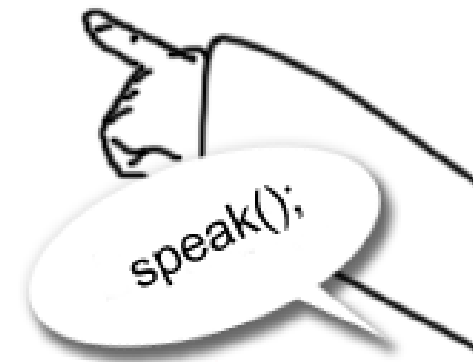
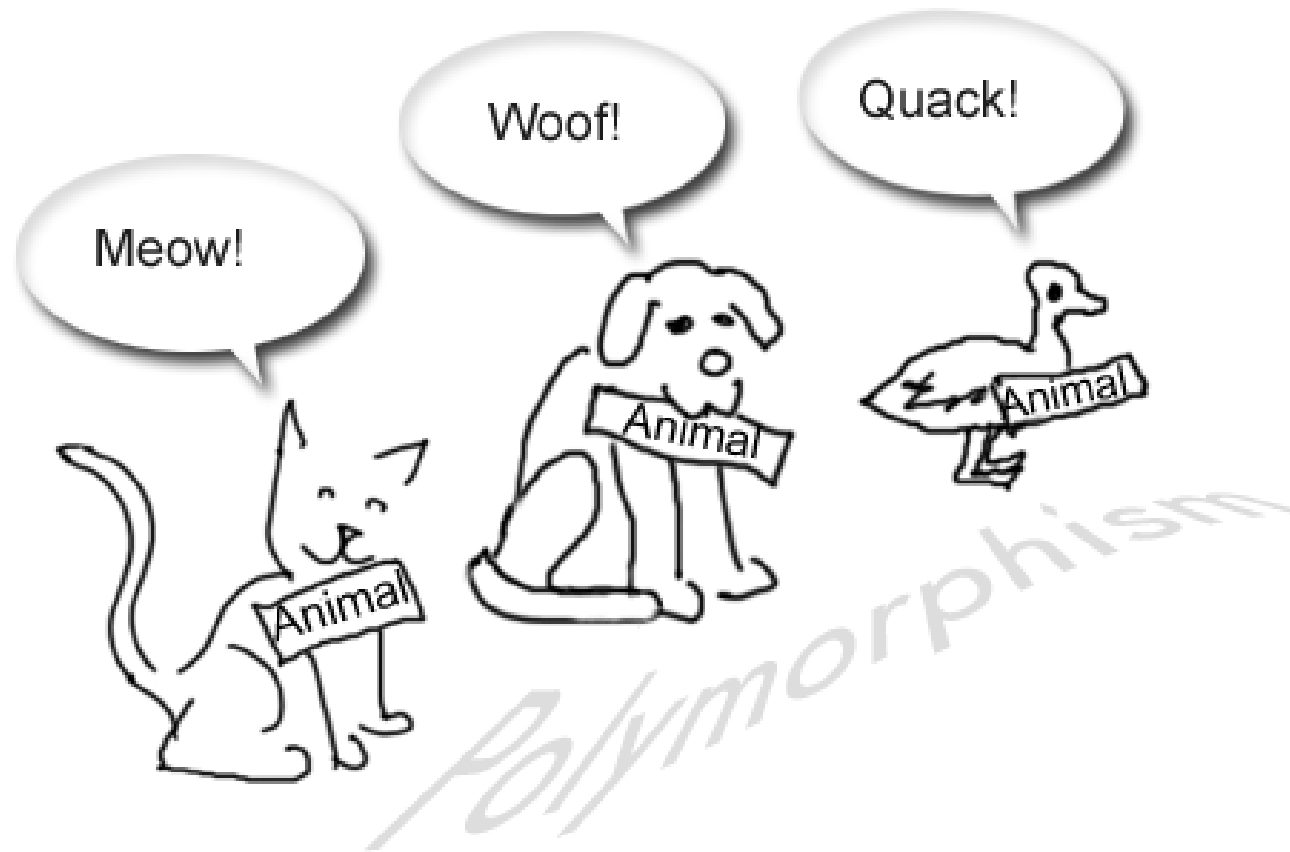
proveniente da matemática

- uma funcionalidade com múltiplas formas de implementação

proveniente da POO

## 4.1. CONCEITO

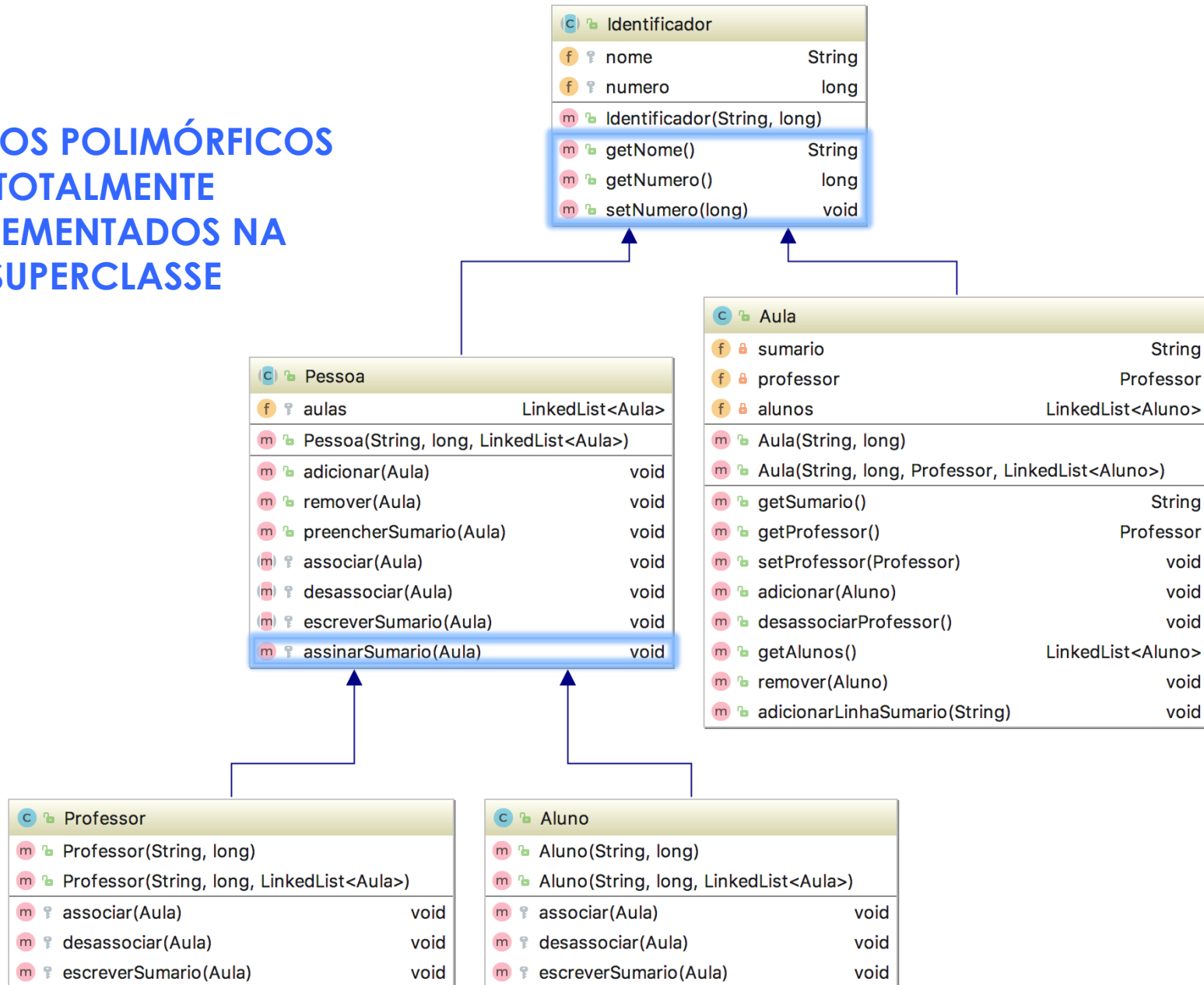
4



# 4.1. CONCEITO

5

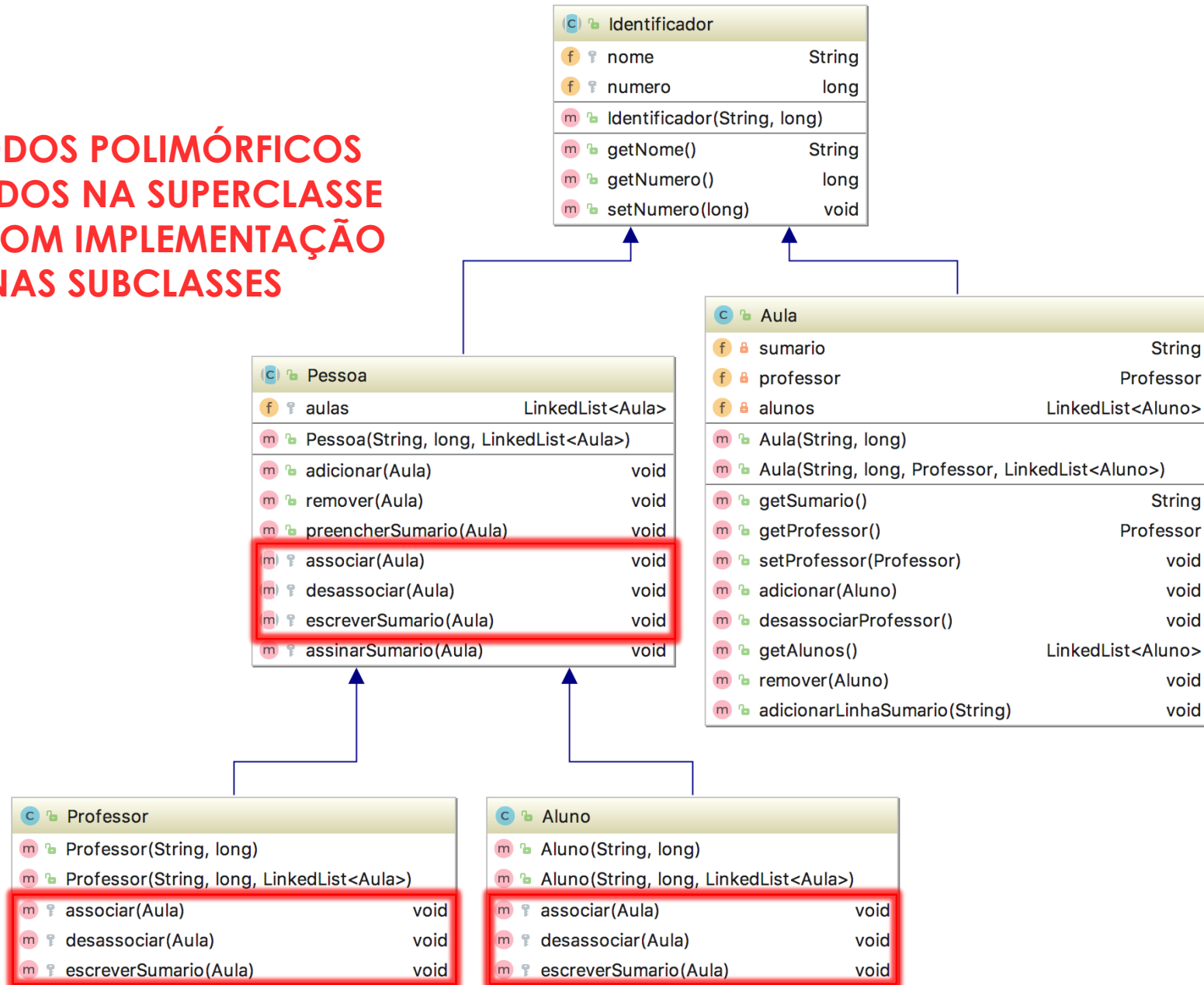
## MÉTODOS POLIMÓRFICOS TOTALMENTE IMPLEMENTADOS NA SUPERCLASSE



# 4.1. CONCEITO

6

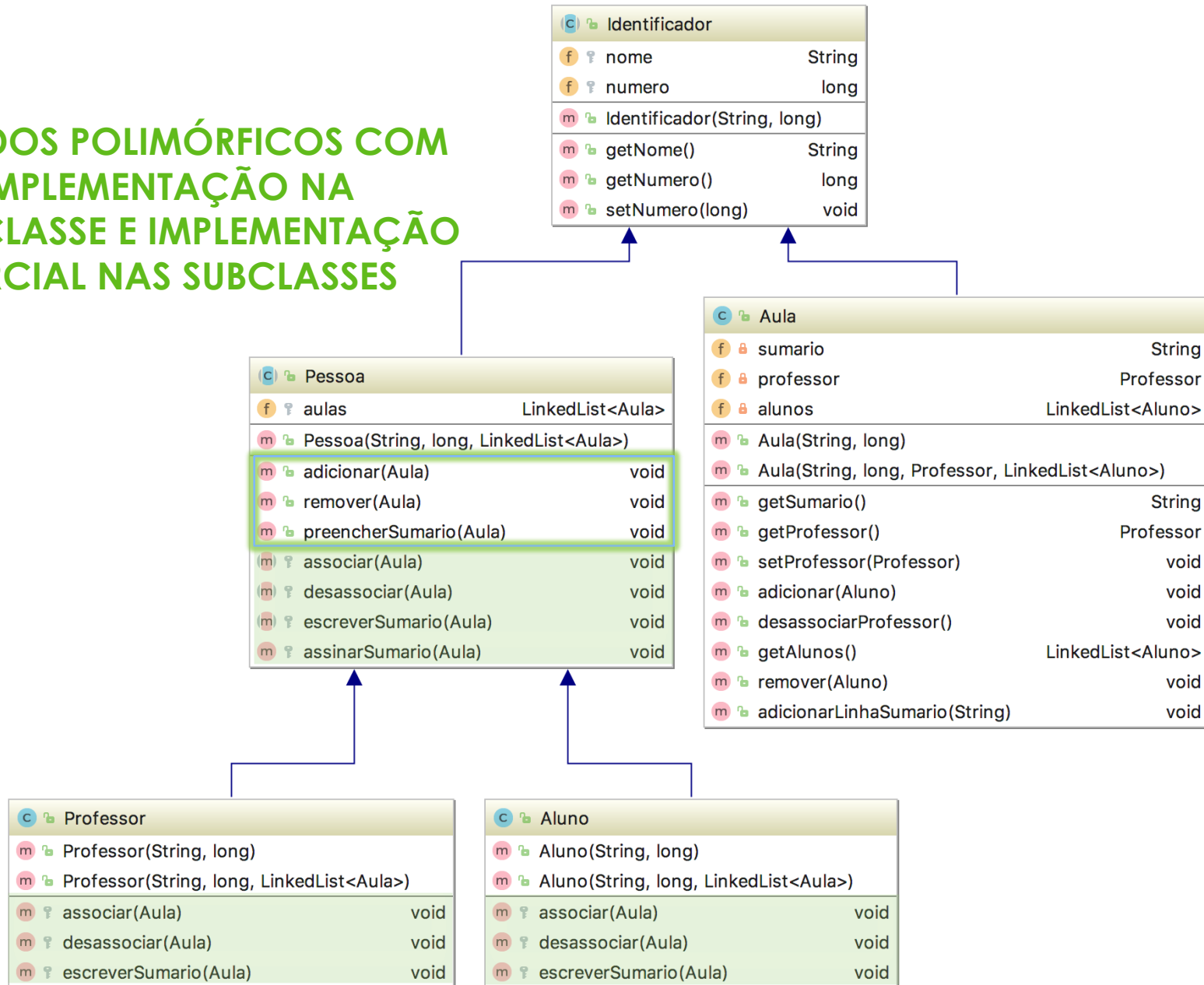
**MÉTODOS POLIMÓRFICOS  
DEFINIDOS NA SUPERCLASSE  
MAS COM IMPLEMENTAÇÃO  
NAS SUBCLASSES**



# 4.1. CONCEITO

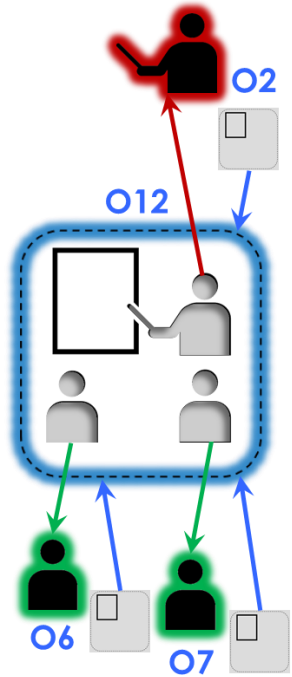
7

## MÉTODOS POLIMÓRFICOS COM IMPLEMENTAÇÃO NA SUPERCLASSE E IMPLEMENTAÇÃO PARCIAL NAS SUBCLASSES



## 4.2. EXEMPLO

8



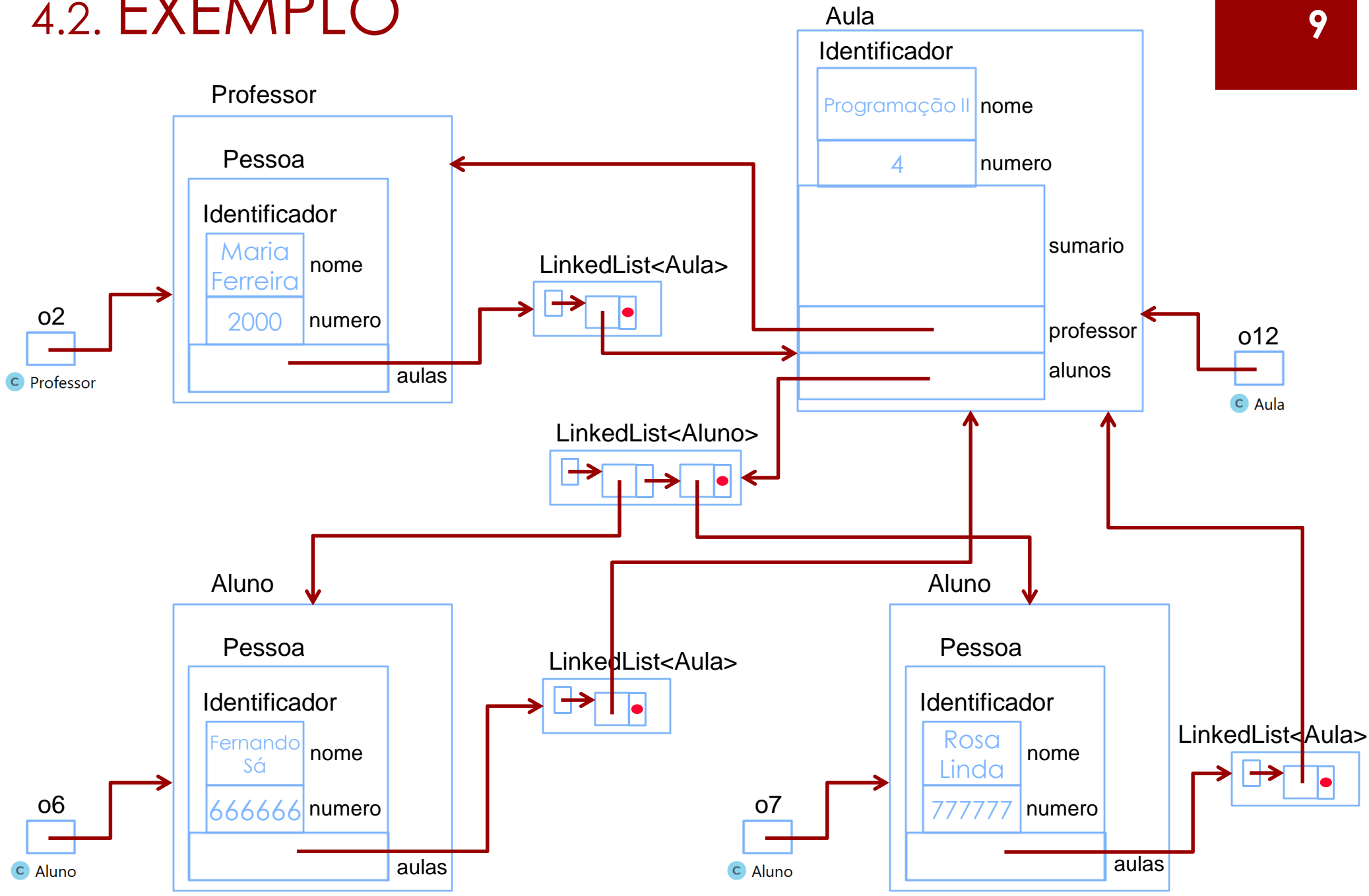
c Main

```
public Main() {  
    ...  
    LinkedList<Aluno> alunos2 = new LinkedList<>();  
    alunos2.add(o6);  
    alunos2.add(o7);  
    Aula o12 = new Aula("Programação II", 4, o2, alunos2);  
}
```

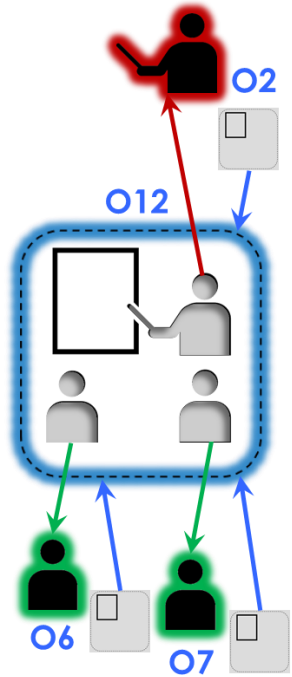


## 4.2. EXEMPLO

9



## 4.2. EXEMPLO



Main

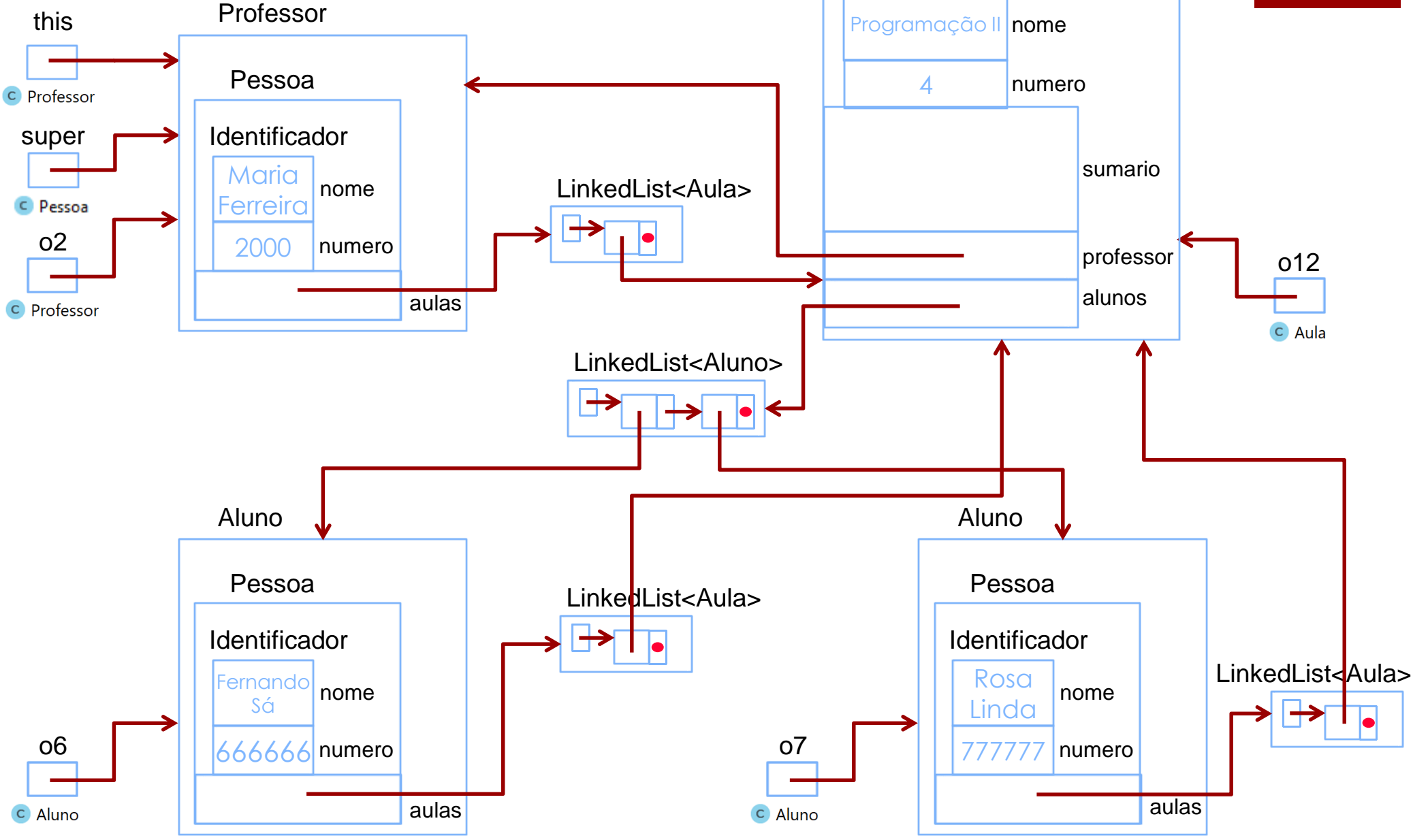
```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

10

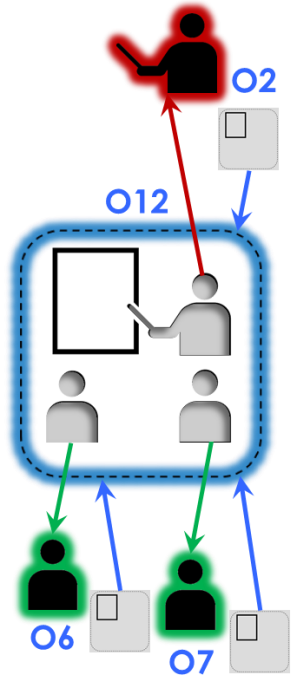
COMO DEVE, UM PROFESSOR,  
PREENCHER O SUMÁRIO  
DE UMA AULA?

## 4.2. EXEMPLO

11



## 4.2. EXEMPLO



```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

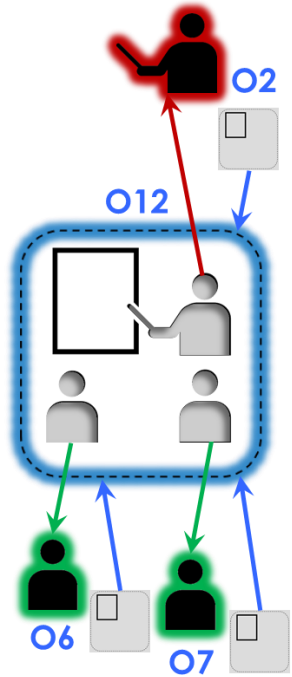
Professor

??????

12

**MAS A CLASSE Professor**  
**NÃO TEM MÉTODO**  
**preencherSumario(Aula)**

## 4.2. EXEMPLO



Main

```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

13

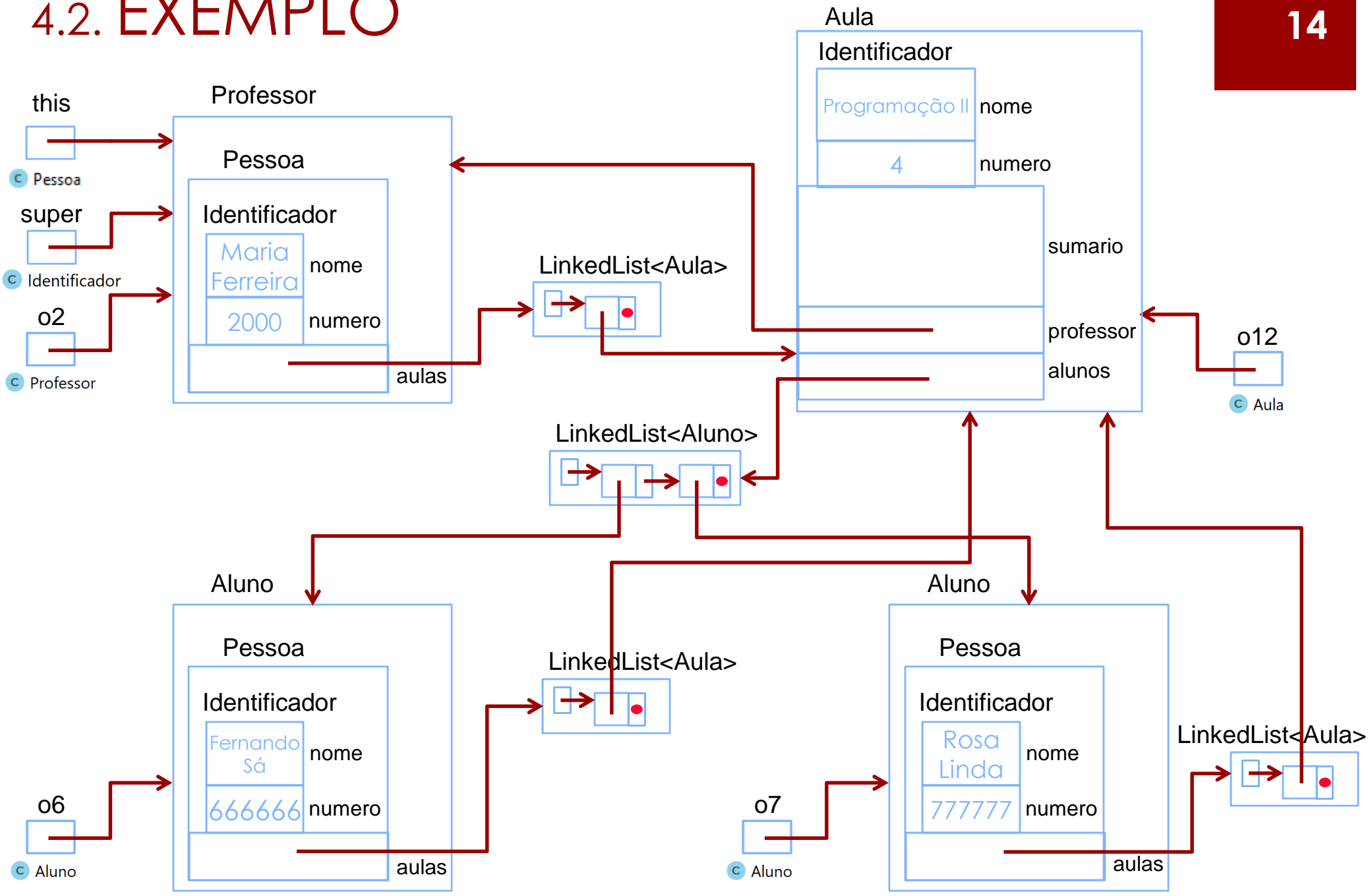
Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

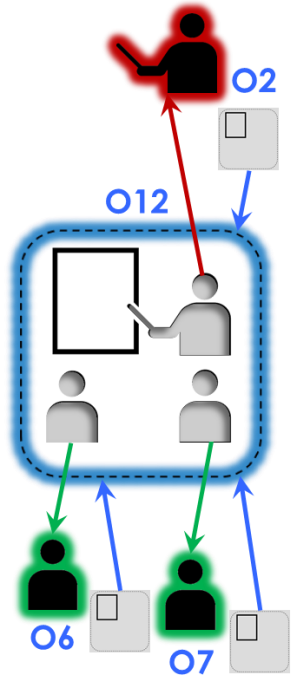
**PROCURA O MÉTODO**  
**preencherSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

## 4.2. EXEMPLO

14



## 4.2. EXEMPLO



Main

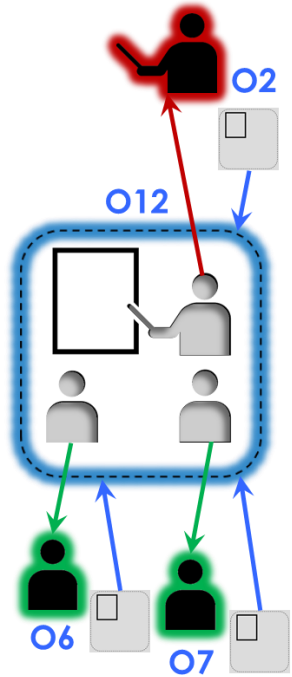
```
public Main() {  
    o2.preencherSumario(o12);  
}
```

15

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

## 4.2. EXEMPLO



Main

```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

16

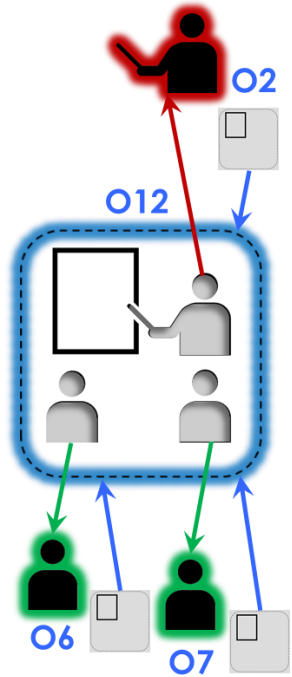
Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

COMO É QUE UMA  
PESSOA ESCREVE  
O SUMÁRIO DE  
UMA AULA?



## 4.2. EXEMPLO



Main

```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

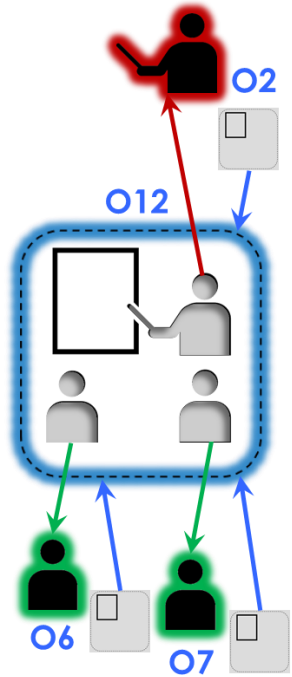
17

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

○ this É UMA  
REFERÊNCIA PARA  
UM PROFESSOR

## 4.2. EXEMPLO



Main

```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

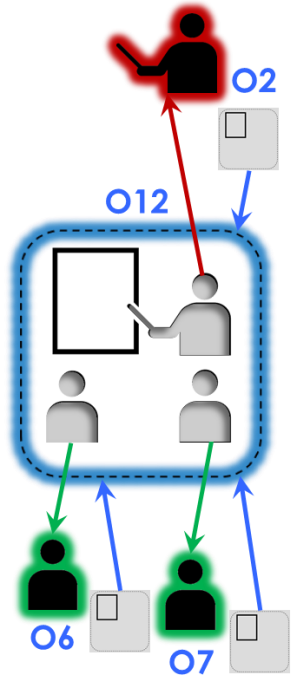
18

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

LOGO DEVE COMEÇAR A  
PROCURAR O MÉTODO  
escreverSumario(Aula)  
NA CLASSE Professor

## 4.2. EXEMPLO



19

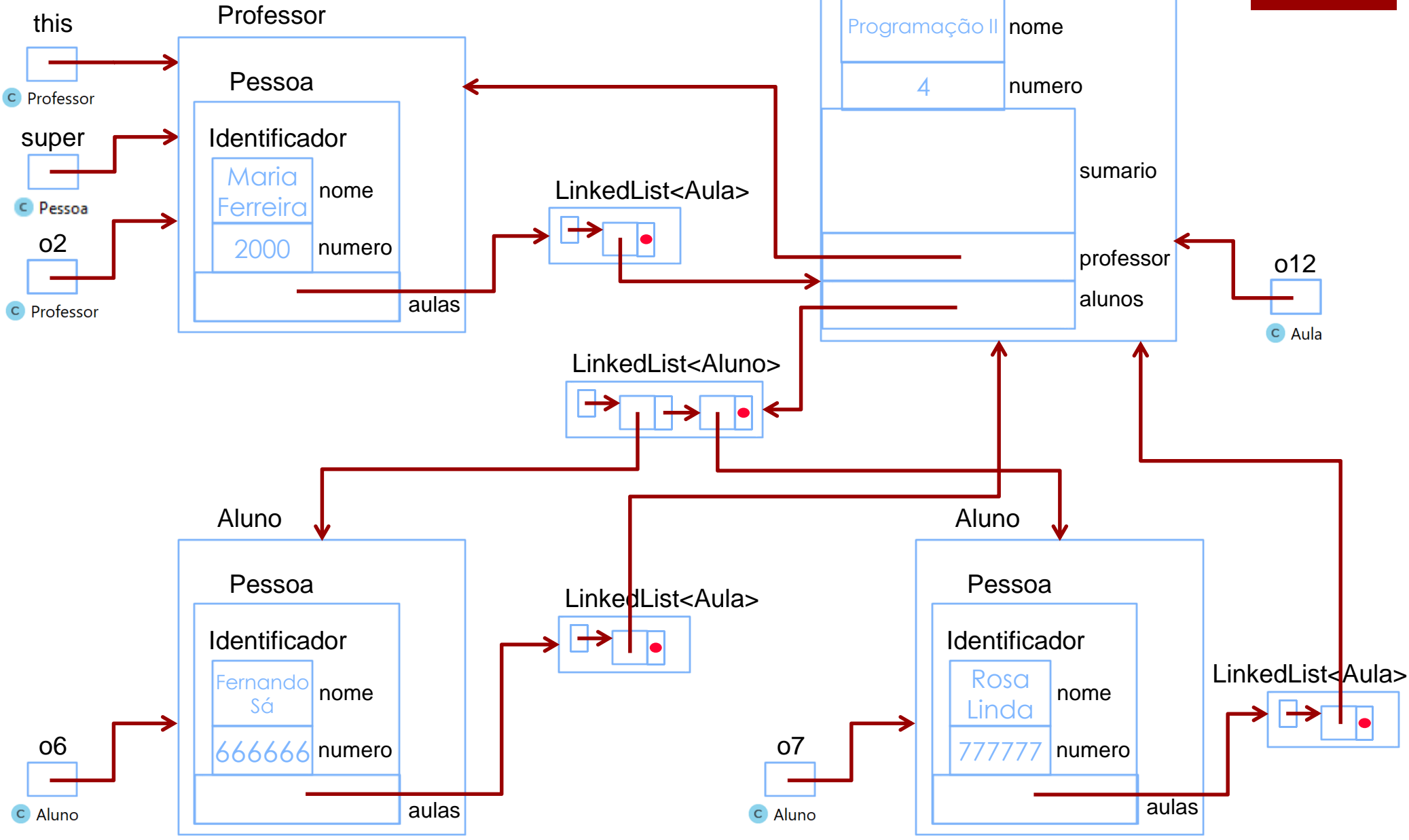
```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

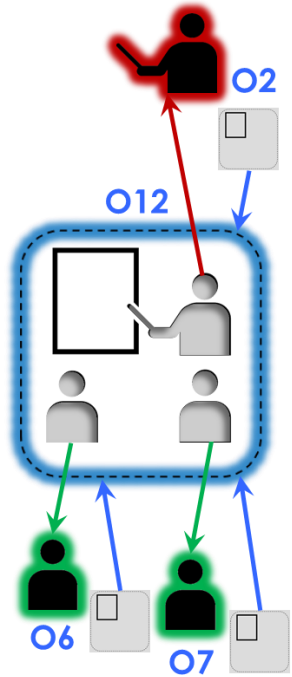
```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

## 4.2. EXEMPLO

20



## 4.2. EXEMPLO



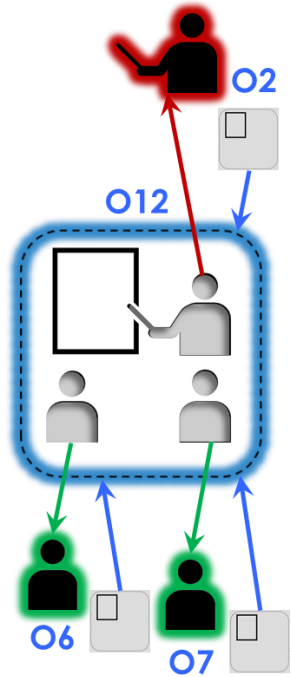
21

```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

## 4.2. EXEMPLO



22

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

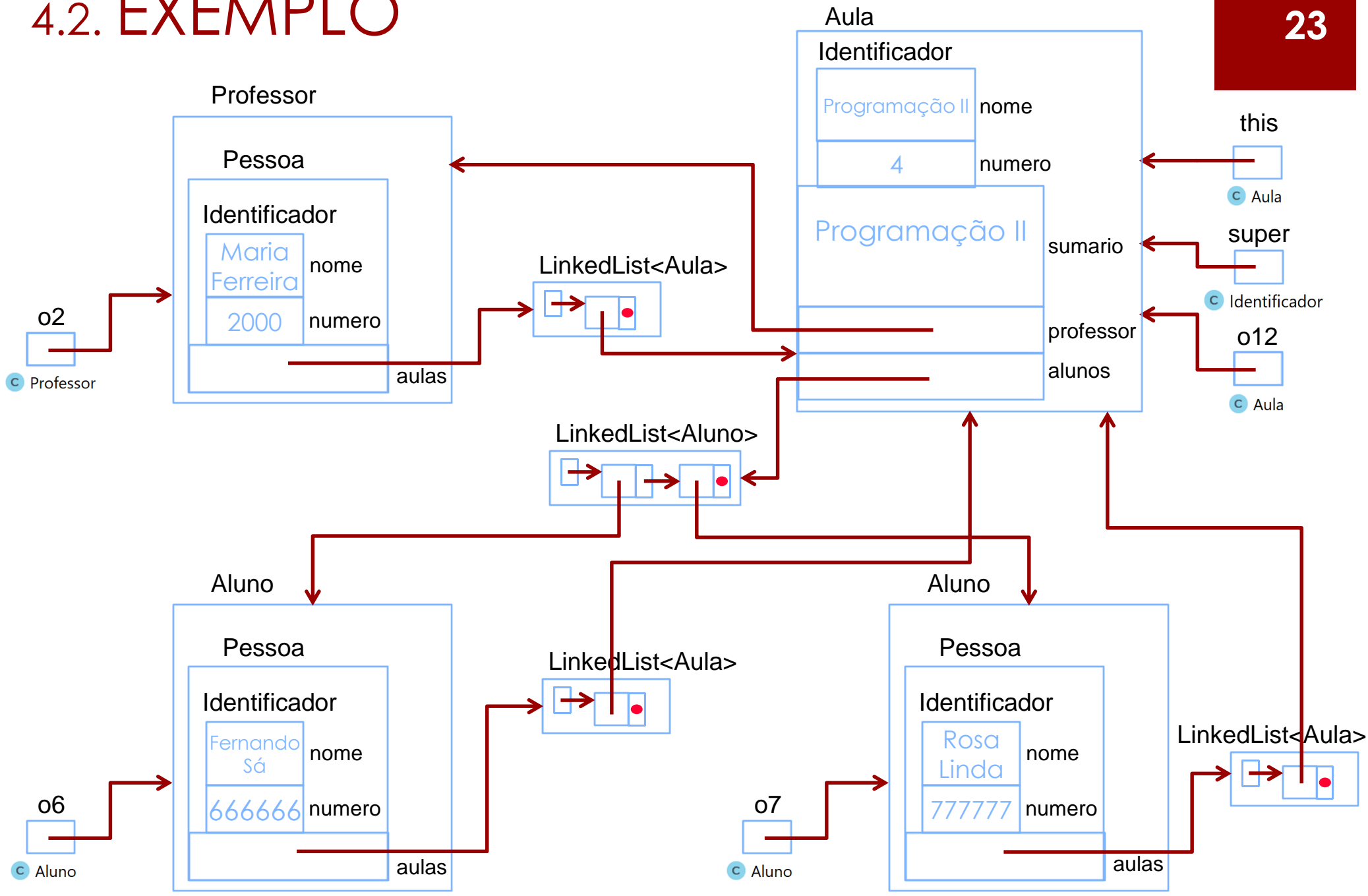
```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

Aula

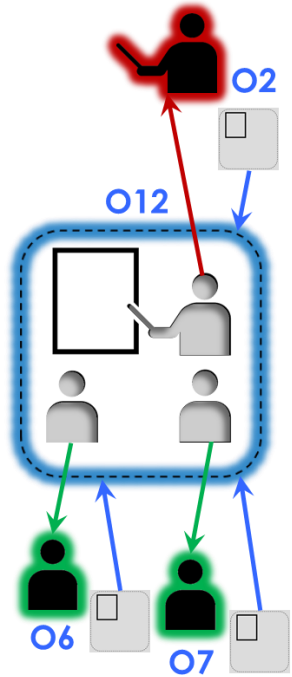
```
public void adicionarLinhaSumario(String linha) {  
    StringBuilder sb = new StringBuilder(sumario);  
    sb.append(linha).append("\n");  
    sumario = sb.toString();  
}
```

## 4.2. EXEMPLO

23



## 4.2. EXEMPLO



24

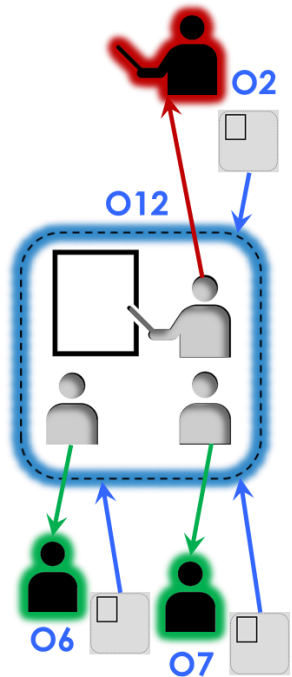
```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```



## 4.2. EXEMPLO



25

```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

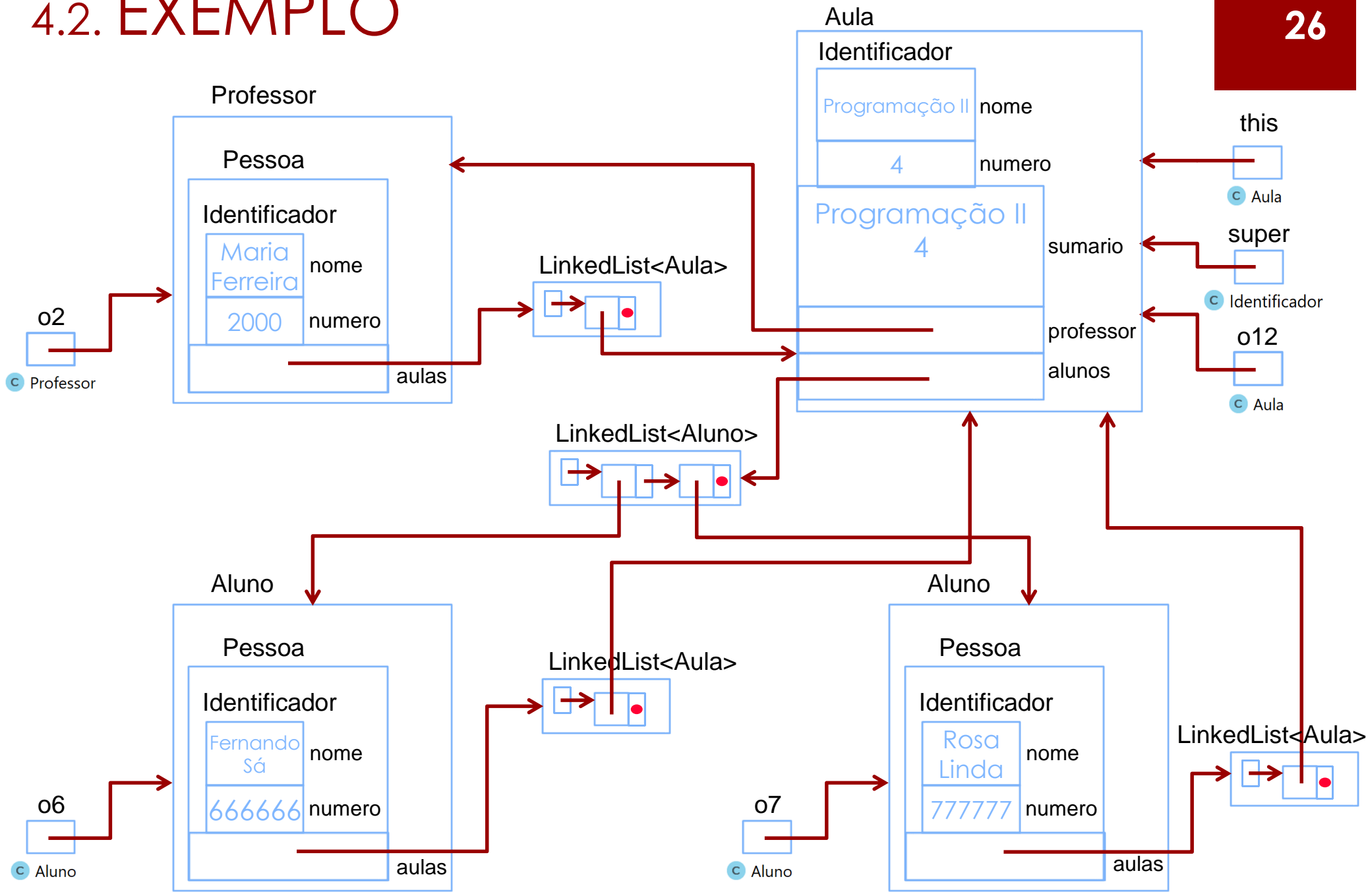
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

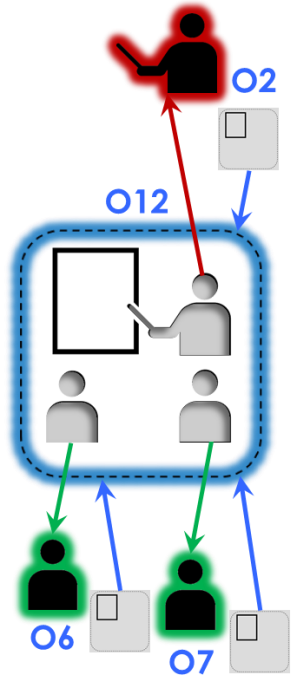
```
Aula  
public void adicionarLinhaSumario(String linha) {  
    StringBuilder sb = new StringBuilder(sumario);  
    sb.append(linha).append("\n");  
    sumario = sb.toString();  
}
```

## 4.2. EXEMPLO

26



## 4.2. EXEMPLO



Main

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

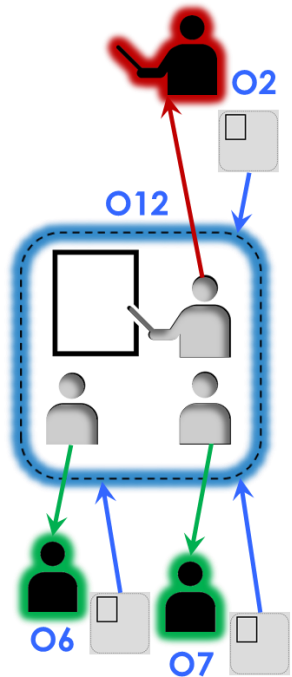
Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

## 4.2. EXEMPLO



28

```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

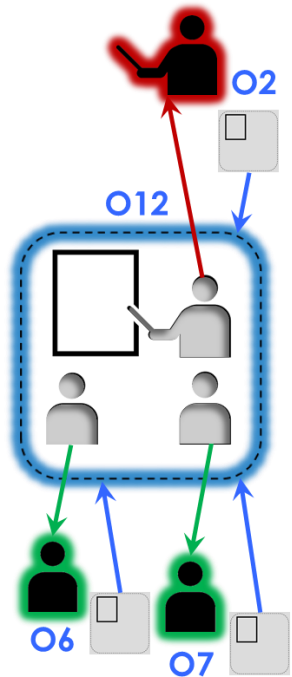
```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

Professor

??????

**MAS A CLASSE Professor**  
**NÃO TEM MÉTODO**  
**assinarSumario(Aula)**

## 4.2. EXEMPLO



29

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

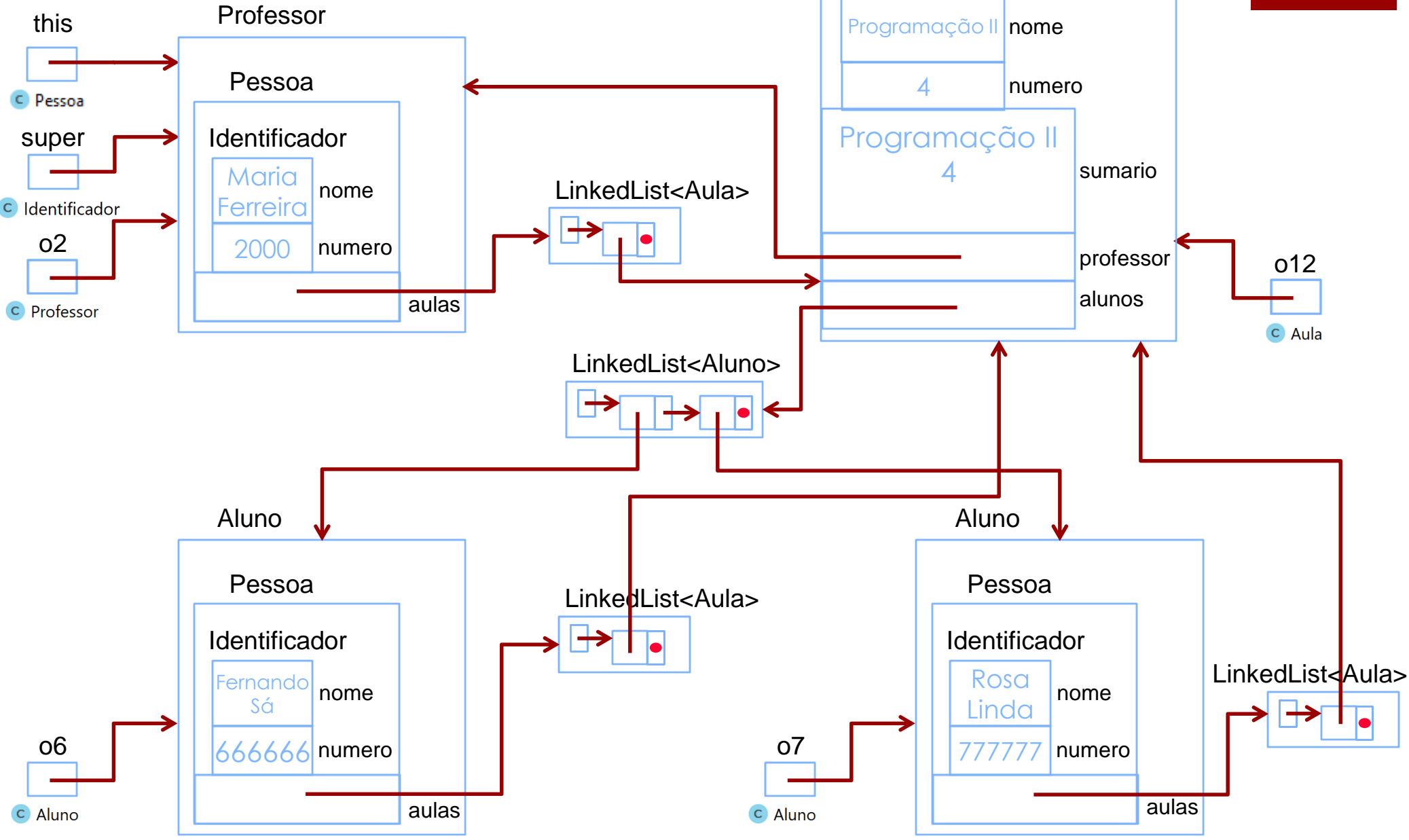
Pessoa

```
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

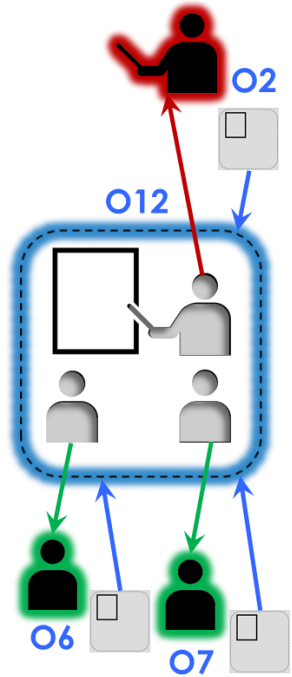
**PROCURA O MÉTODO**  
**assinarSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

## 4.2. EXEMPLO

30



## 4.2. EXEMPLO



31

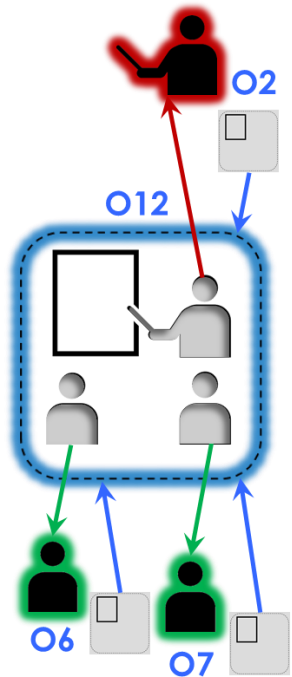
```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

```
Pessoa  
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

## 4.2. EXEMPLO



32

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

Pessoa

```
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

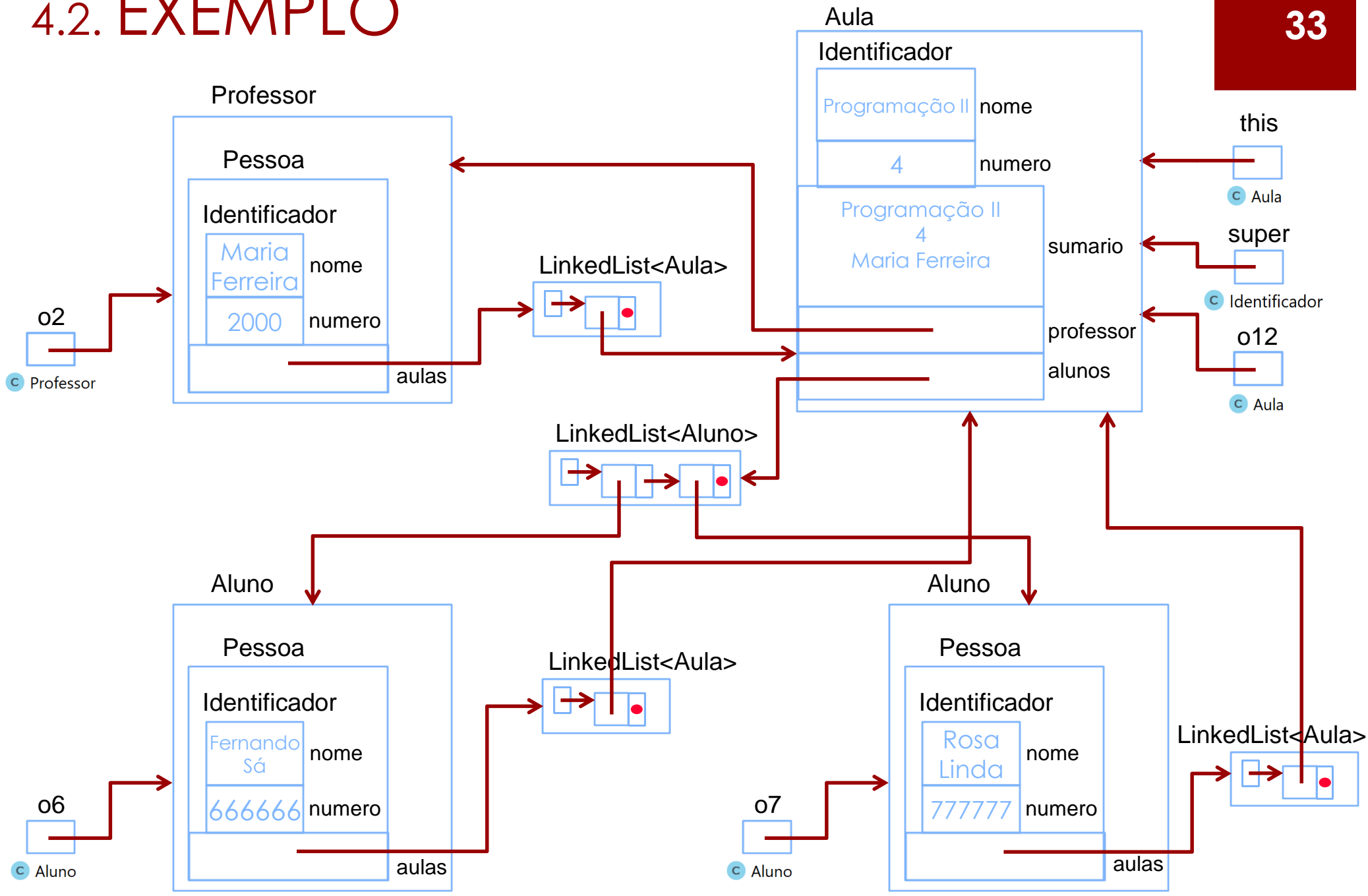
Aula

```
public void adicionarLinhaSumario(String linha) {  
    StringBuilder sb = new StringBuilder(sumario);  
    sb.append(linha).append("\n");  
    sumario = sb.toString();  
}
```

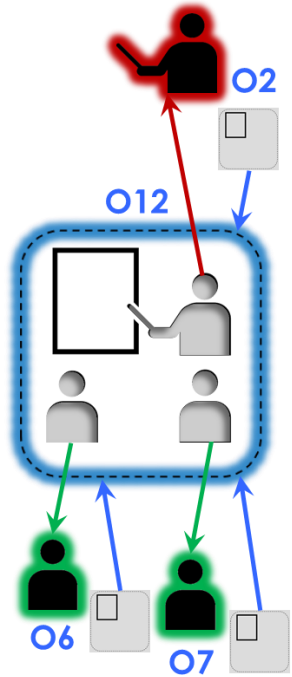


## 4.2. EXEMPLO

33



## 4.2. EXEMPLO



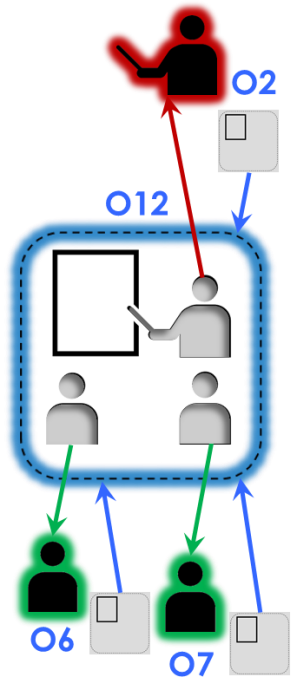
34

```
Main
public Main() {
    o2.preencherSumario(o12);
}
```

```
Pessoa
public void preencherSumario(Aula aula) {
    if (!aulas.contains(aula)) {
        return;
    }
    escreverSumario(aula);
}
```

```
Professor
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula); 1
    }
}
```

## 4.2. EXEMPLO



35

```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ①  
    }  
}
```

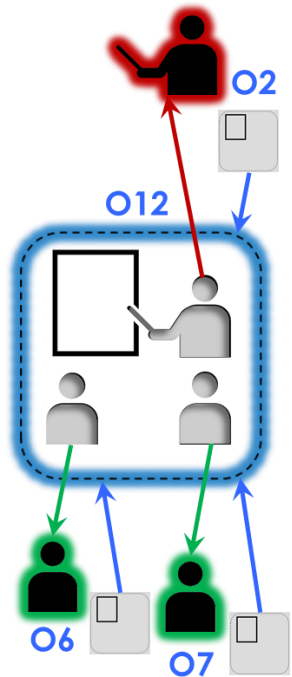
Aluno

??????

**MAS A CLASSE Aluno  
NÃO TEM MÉTODO  
preencherSumario(Aula)**

## 4.2. EXEMPLO

36



```
Main
public Main() {
    o2.preencherSumario(o12);
}
```

```
Pessoa
public void preencherSumario(Aula aula) {
    if (!aulas.contains(aula)) {
        return;
    }
    escreverSumario(aula);
}
```

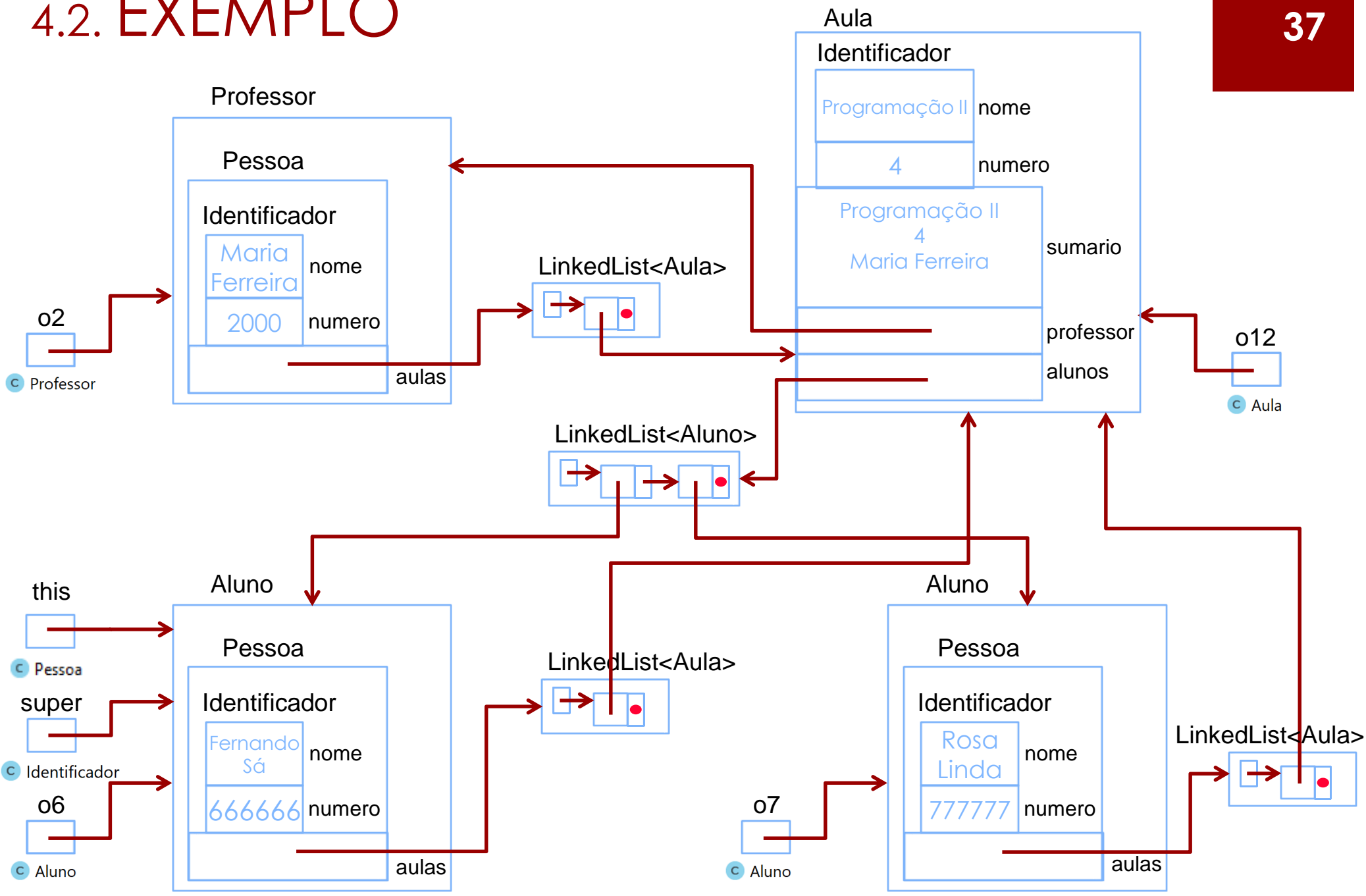
```
Professor
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula); ①
    }
}
```

```
Pessoa
public void preencherSumario(Aula aula) {
    if (!aulas.contains(aula)) {
        return;
    }
    escreverSumario(aula);
}
```

**PROCURA O MÉTODO**  
**preencherSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

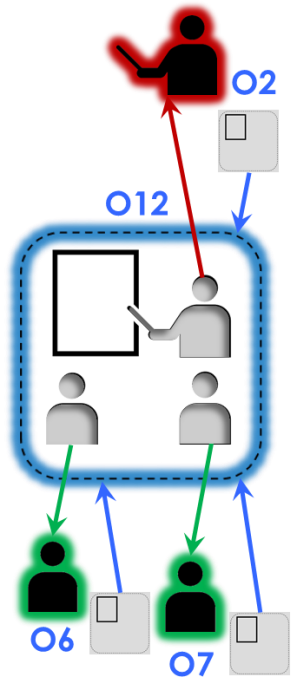
## 4.2. EXEMPLO

37



## 4.2. EXEMPLO

38



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

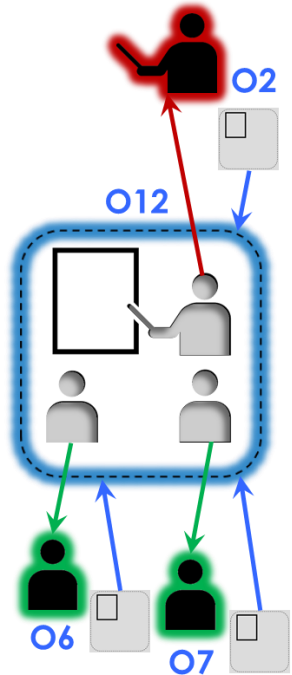
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); 1  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

## 4.2. EXEMPLO

39



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

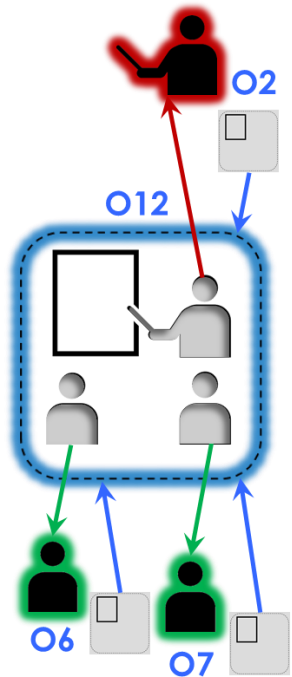
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ①  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

COMO É QUE UMA  
PESSOA ESCREVE  
O SUMÁRIO DE  
UMA AULA?

## 4.2. EXEMPLO



© Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

© Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ①  
    }  
}
```

© Main

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

© Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

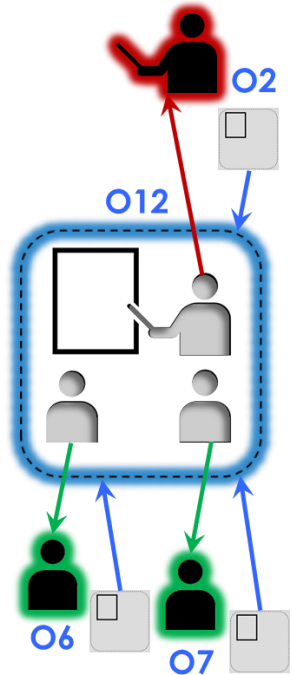
40

○ this É UMA  
REFERÊNCIA PARA  
UM ALUNO



## 4.2. EXEMPLO

41



```
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

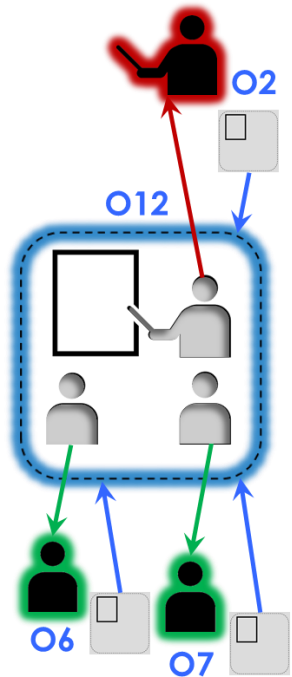
```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ①  
    }  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

LOGO DEVE COMEÇAR A  
PROCURAR O MÉTODO  
escreverSumario(Aula)  
NA CLASSE Aluno

## 4.2. EXEMPLO

42



```
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

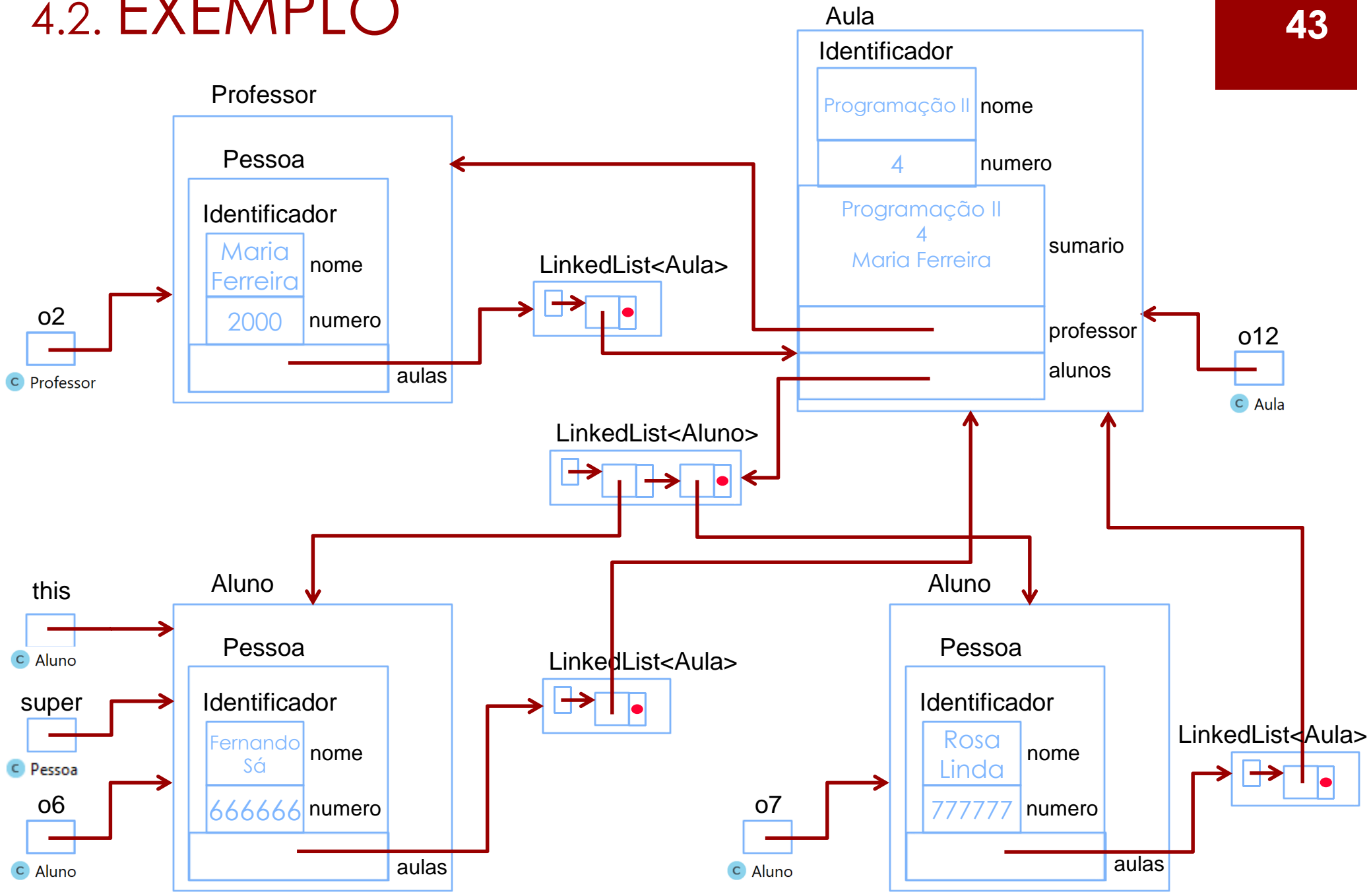
```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

## 4.2. EXEMPLO

43



44

```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {
    if (!aulas.contains(aula)) {
        return;
    }
    escreverSumario(aula);
}
```

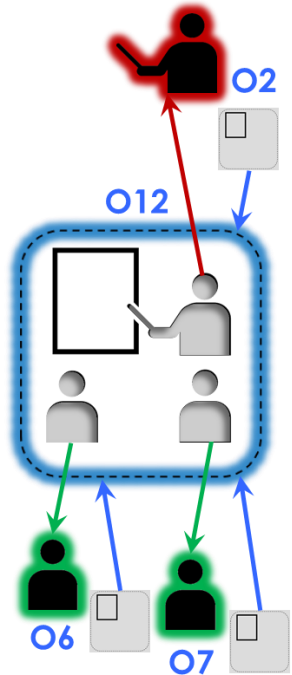
```
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula); ①
    }
}
```

```
public void preencherSumario(Aula aula) {
    if (!aulas.contains(aula)) {
        return;
    }
    escreverSumario(aula);
}
```

```
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

## 4.2. EXEMPLO

45



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ①  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Aluno  
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

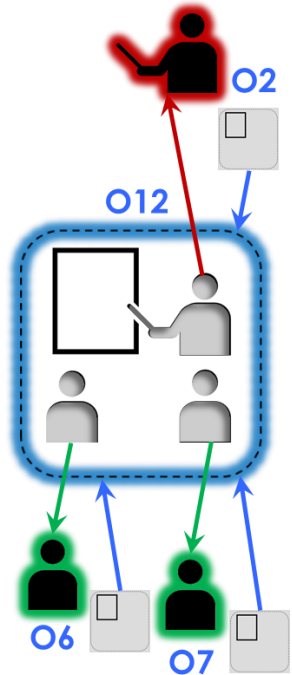
Aluno

??????

**MAS A CLASSE Aluno  
NÃO TEM MÉTODO  
assinarSumario(Aula)**

## 4.2. EXEMPLO

46



```
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

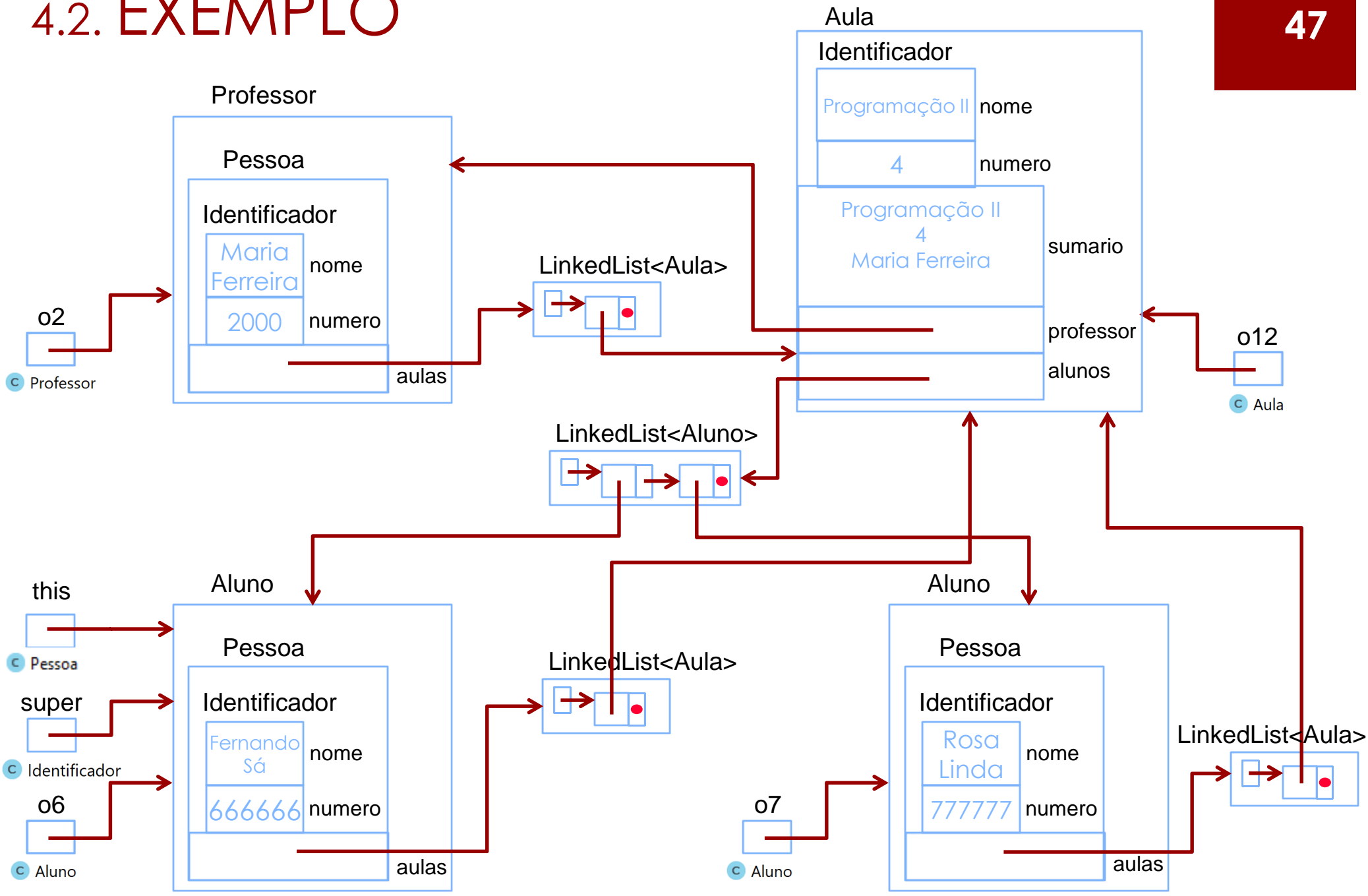
```
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

```
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

**PROCURA O MÉTODO**  
**assinarSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

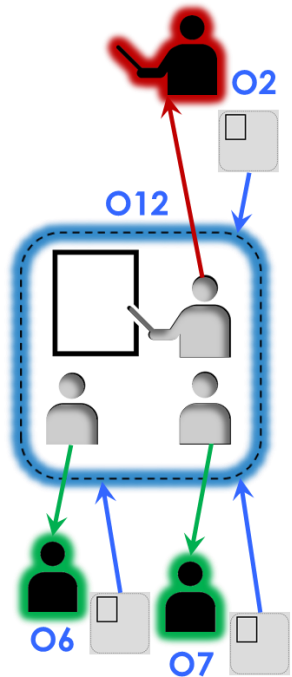
## 4.2. EXEMPLO

47



## 4.2. EXEMPLO

48



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); 1  
    }  
}
```

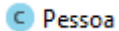
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Aluno  
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

```
Pessoa  
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```



## 49



**C** Aluno

**C** Pessoa

```
protected void assinarSumario(Aula aula) {
    aula.adicionarLinhaSumario(nome);
}
```

Ⓒ Pessoa

**C** Professor

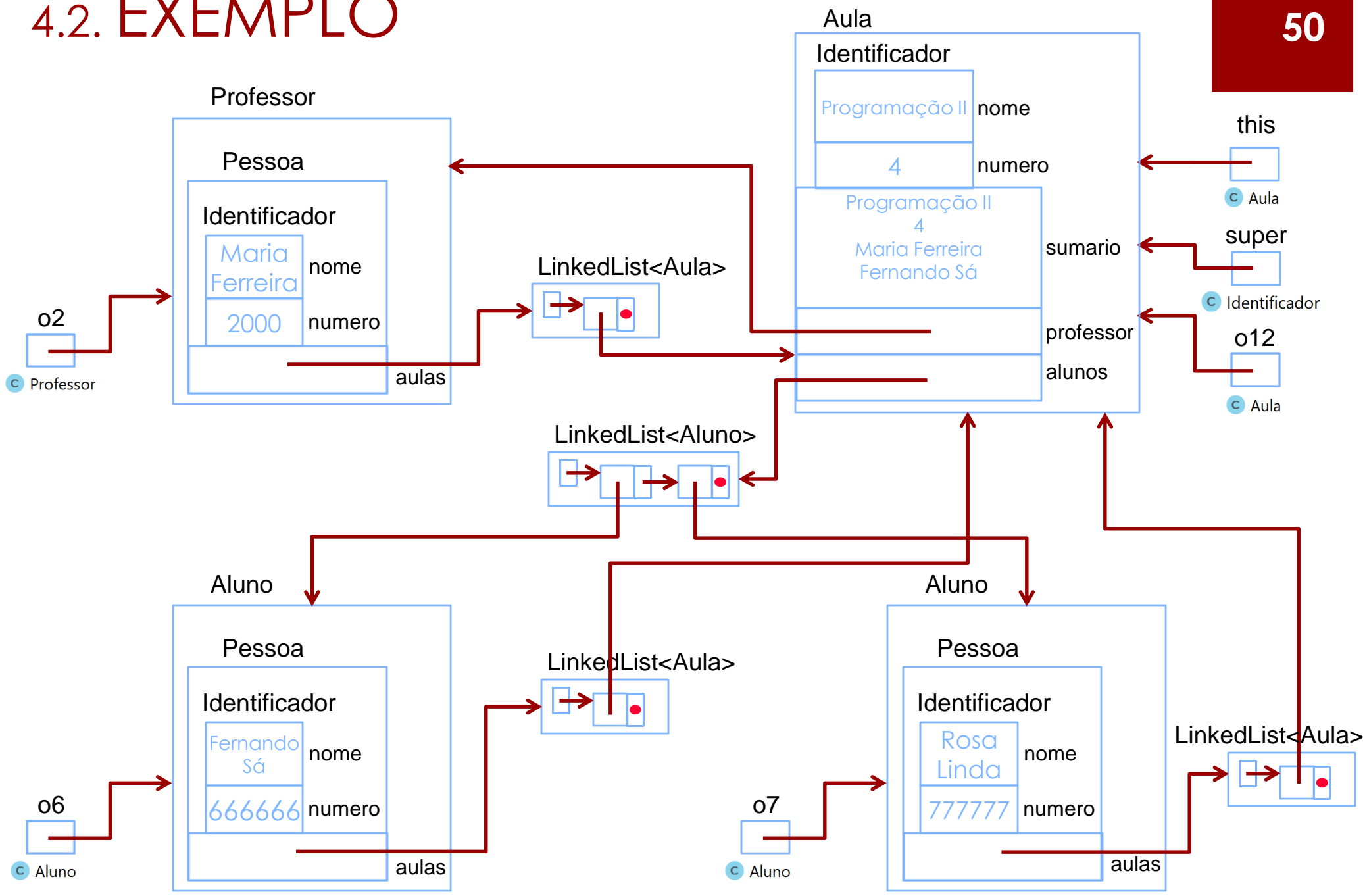
```
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula); 1
    }
}
```

C Aula

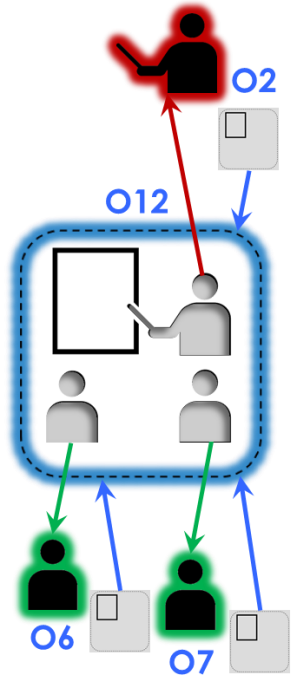
```
public void adicionarLinhaSumario(String linha) {  
    StringBuilder sb = new StringBuilder(sumario);  
    sb.append(linha).append("\n");  
    sumario = sb.toString();  
}
```

## 4.2. EXEMPLO

50



## 4.2. EXEMPLO



Main

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

51

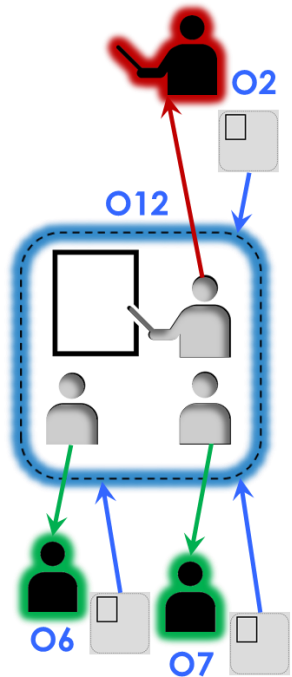
Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

## 4.2. EXEMPLO



Main

```
public Main() {  
    o2.preencherSumario(o12);  
}
```

52

Pessoa

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

Professor

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

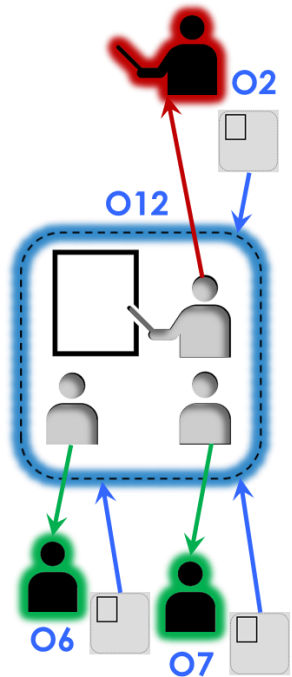
Aluno

??????

**MAS A CLASSE Aluno  
NÃO TEM MÉTODO  
preencherSumario(Aula)**

## 4.2. EXEMPLO

53



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

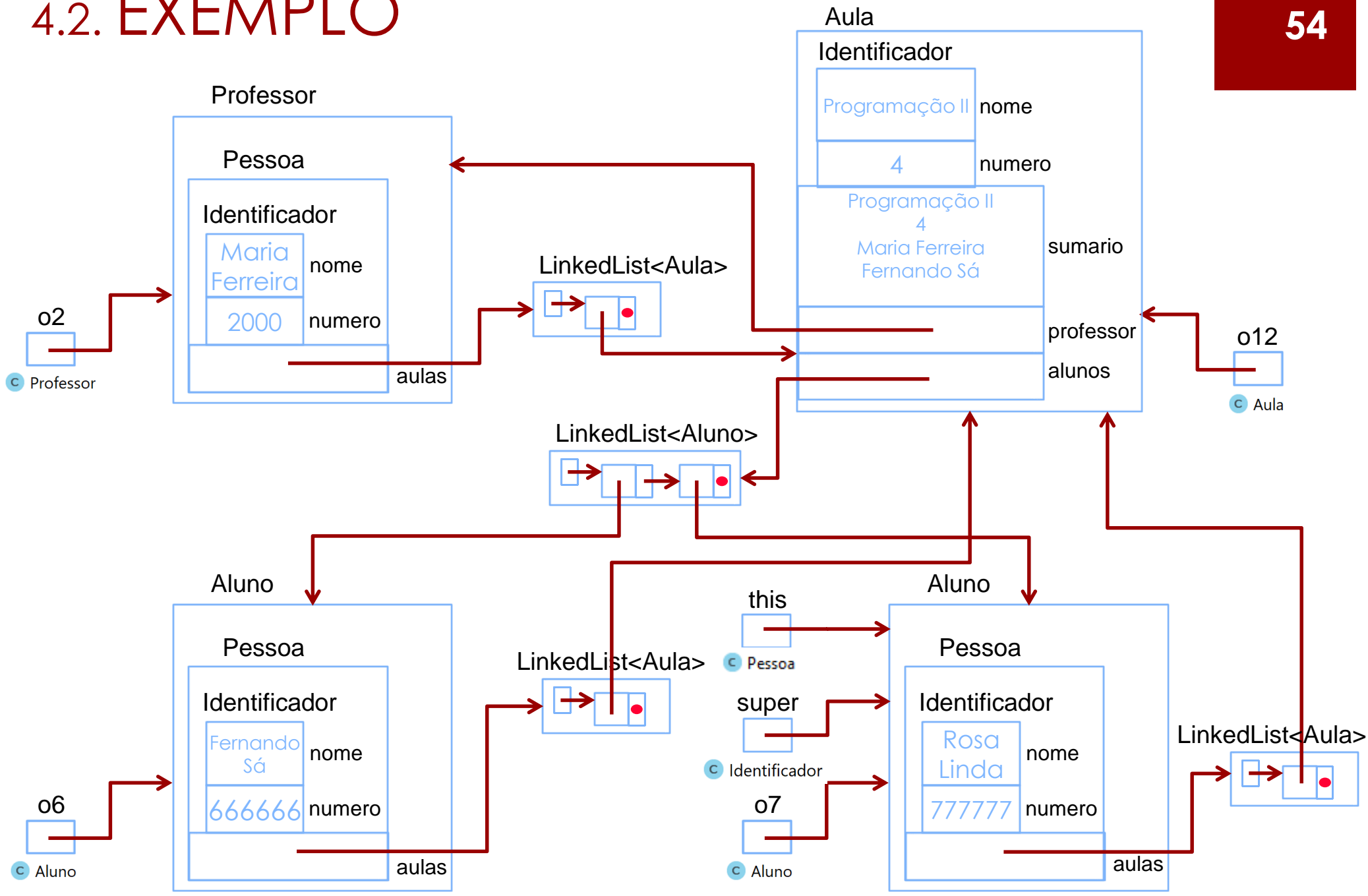
```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

**PROCURA O MÉTODO**  
**preencherSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

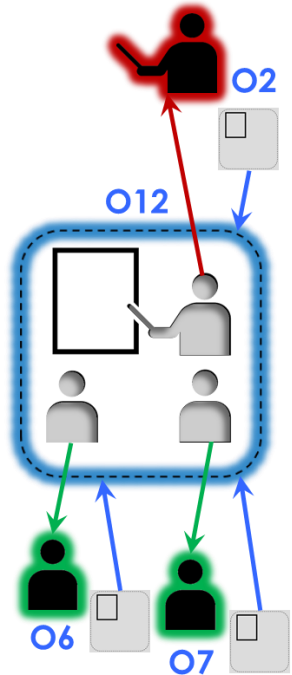
## 4.2. EXEMPLO

54



## 4.2. EXEMPLO

55



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

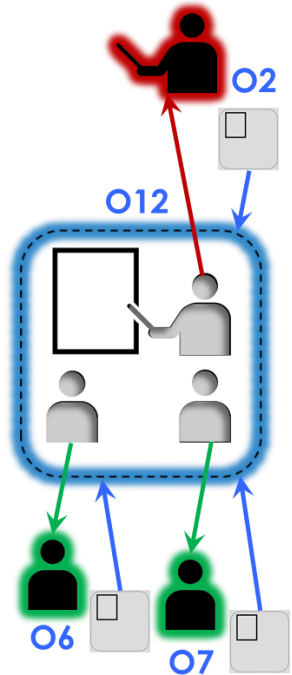
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); 2  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

## 4.2. EXEMPLO

56



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

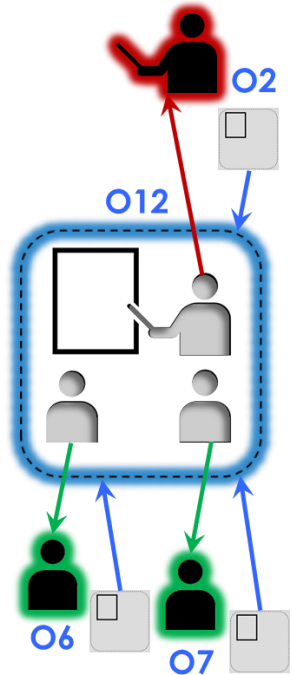
```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

COMO É QUE UMA  
PESSOA ESCREVE  
O SUMÁRIO DE  
UMA AULA?



## 4.2. EXEMPLO

57



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

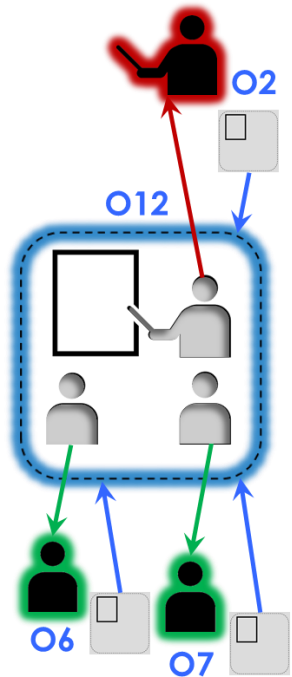
```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

○ this É UMA  
REFERÊNCIA PARA  
UM ALUNO

## 4.2. EXEMPLO

58



```
public Main() {  
    o2.preencherSumario(o12);  
}
```

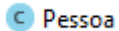
```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

LOGO DEVE COMEÇAR A  
PROCURAR O MÉTODO  
escreverSumario(Aula)  
NA CLASSE Aluno

## 59



**C** Aluno

```
protected void escreverSumario(Aula aula) {
    assinarSumario(aula);
}
```

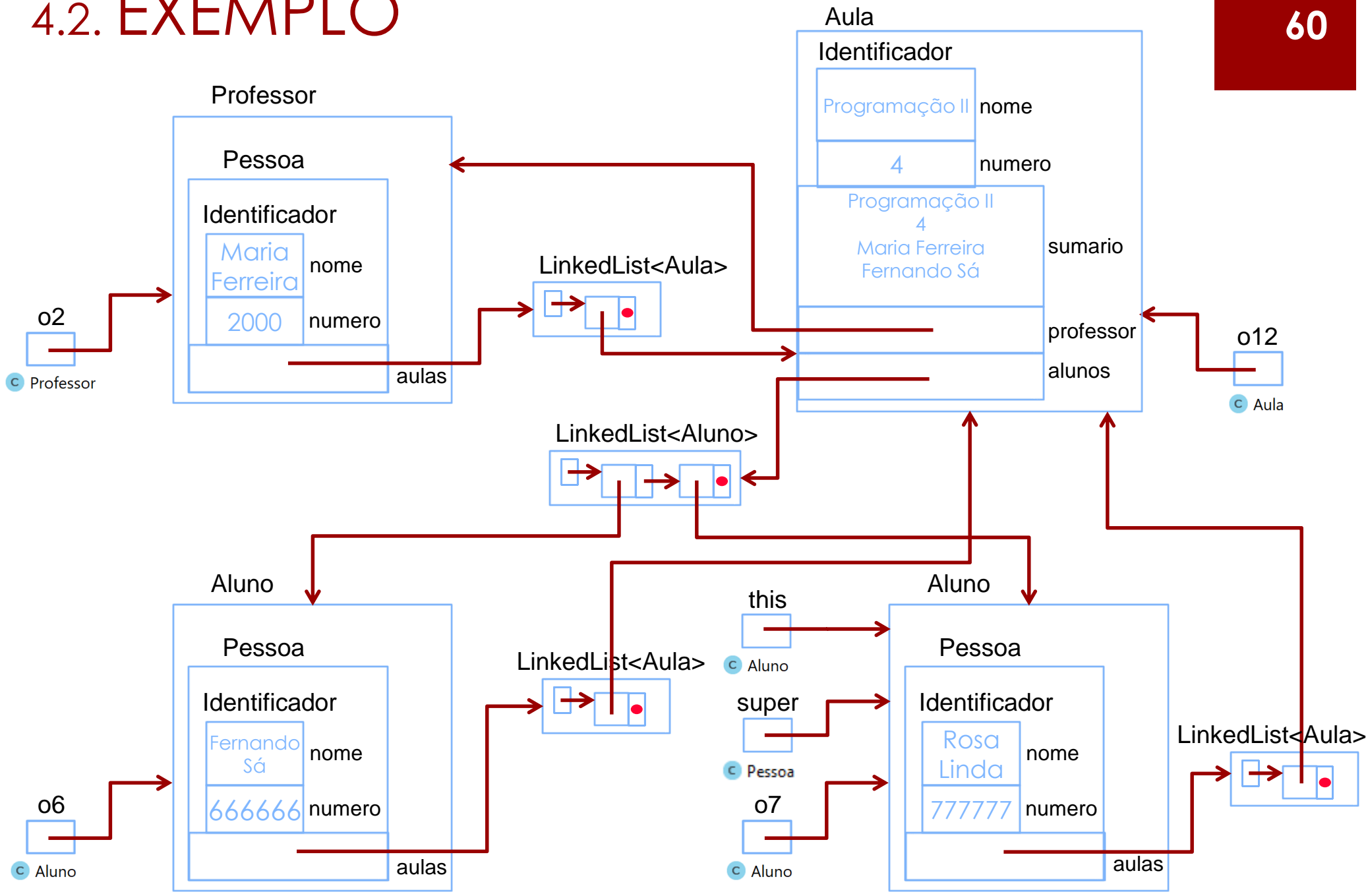
```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

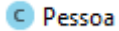
```
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula);
    }
}
```

## 4.2. EXEMPLO

60



## 61



**C** Aluno

```
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

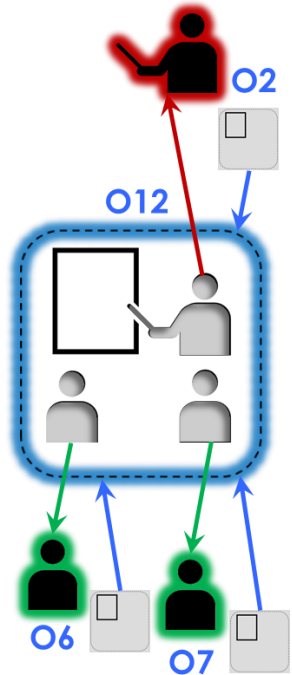
```
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

```
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
protected void escreverSumario(Aula aula) {
    aula.adicionarLinhaSumario(aula.getNome());
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));
    assinarSumario(aula);
    for (Aluno aluno : aula.getAlunos()) {
        aluno.preencherSumario(aula);
    }
}
```

## 4.2. EXEMPLO

62



```
Main  
public Main() {  
    ...  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula);  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Aluno  
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

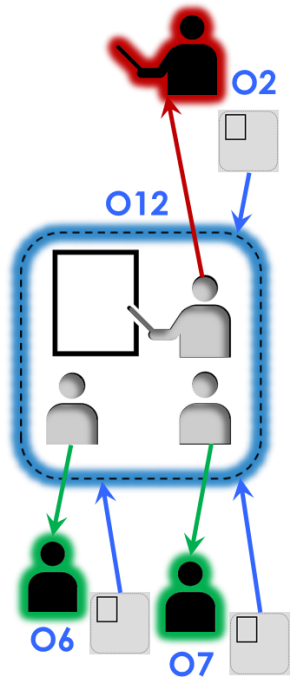
Aluno

??????

**MAS A CLASSE Aluno  
NÃO TEM MÉTODO  
assinarSumario(Aula)**

## 4.2. EXEMPLO

63



```
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
c Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
c Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

```
c Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

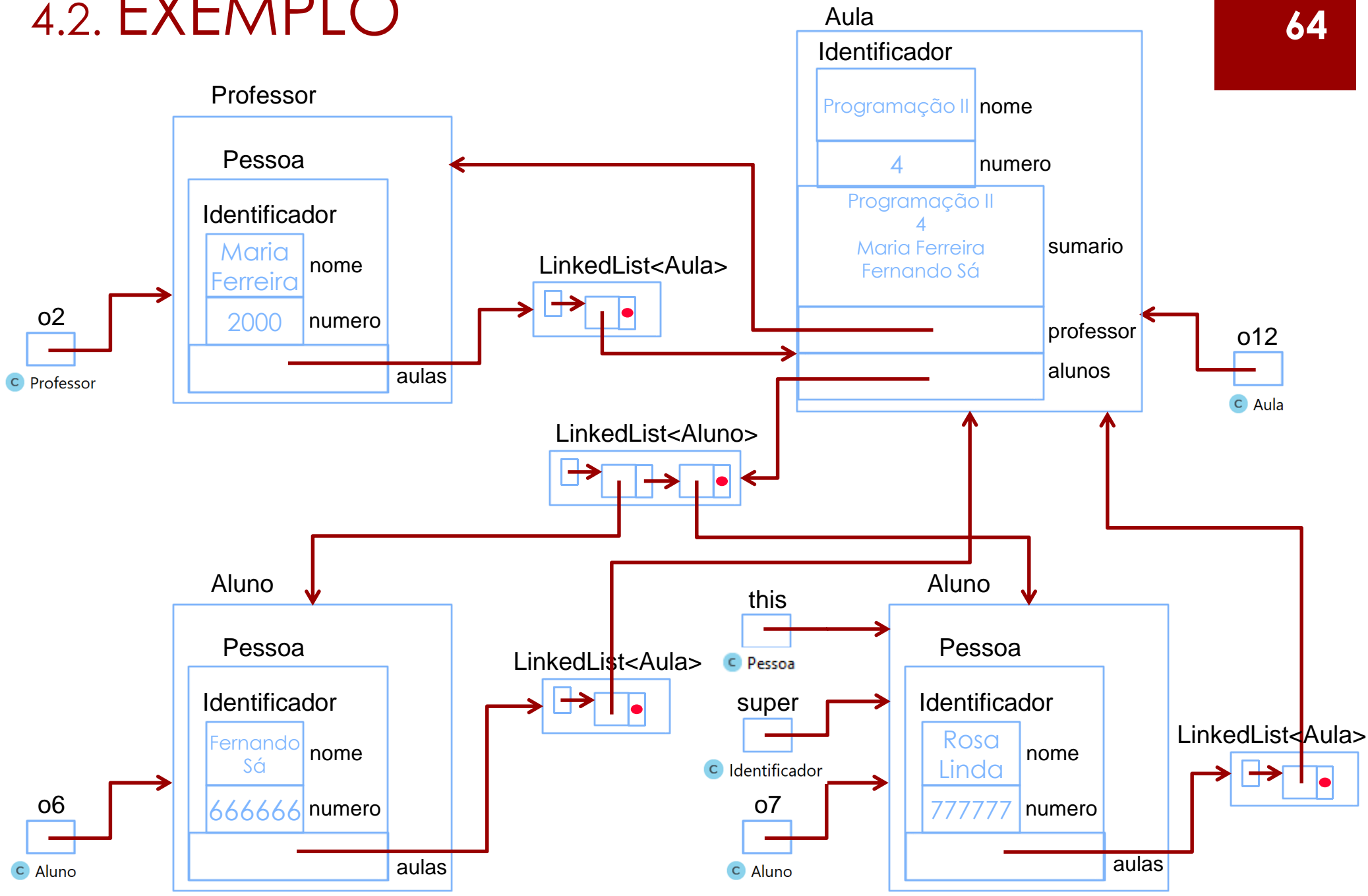
```
c Aluno  
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

```
c Pessoa  
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

**PROCURA O MÉTODO**  
**assinarSumario(Aula)**  
**NA SUPERCLASSE Pessoa**

## 4.2. EXEMPLO

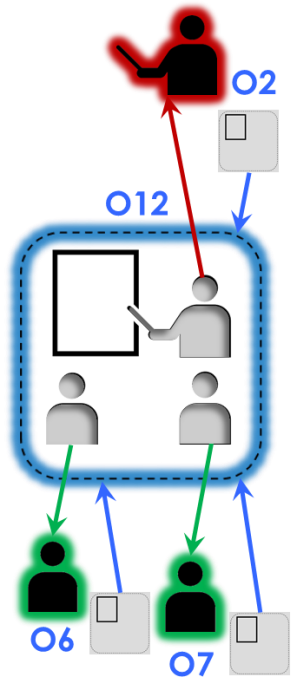
64





## 4.2. EXEMPLO

65



```
Main  
public Main() {  
    o2.preencherSumario(o12);  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

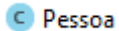
```
Professor  
protected void escreverSumario(Aula aula) {  
    aula.adicionarLinhaSumario(aula.getNome());  
    aula.adicionarLinhaSumario(String.valueOf(aula.getNumero()));  
    assinarSumario(aula);  
    for (Aluno aluno : aula.getAlunos()) {  
        aluno.preencherSumario(aula); ②  
    }  
}
```

```
Pessoa  
public void preencherSumario(Aula aula) {  
    if (!aulas.contains(aula)) {  
        return;  
    }  
    escreverSumario(aula);  
}
```

```
Aluno  
protected void escreverSumario(Aula aula) {  
    assinarSumario(aula);  
}
```

```
Pessoa  
protected void assinarSumario(Aula aula) {  
    aula.adicionarLinhaSumario(nome);  
}
```

## 66



**C** Aluno

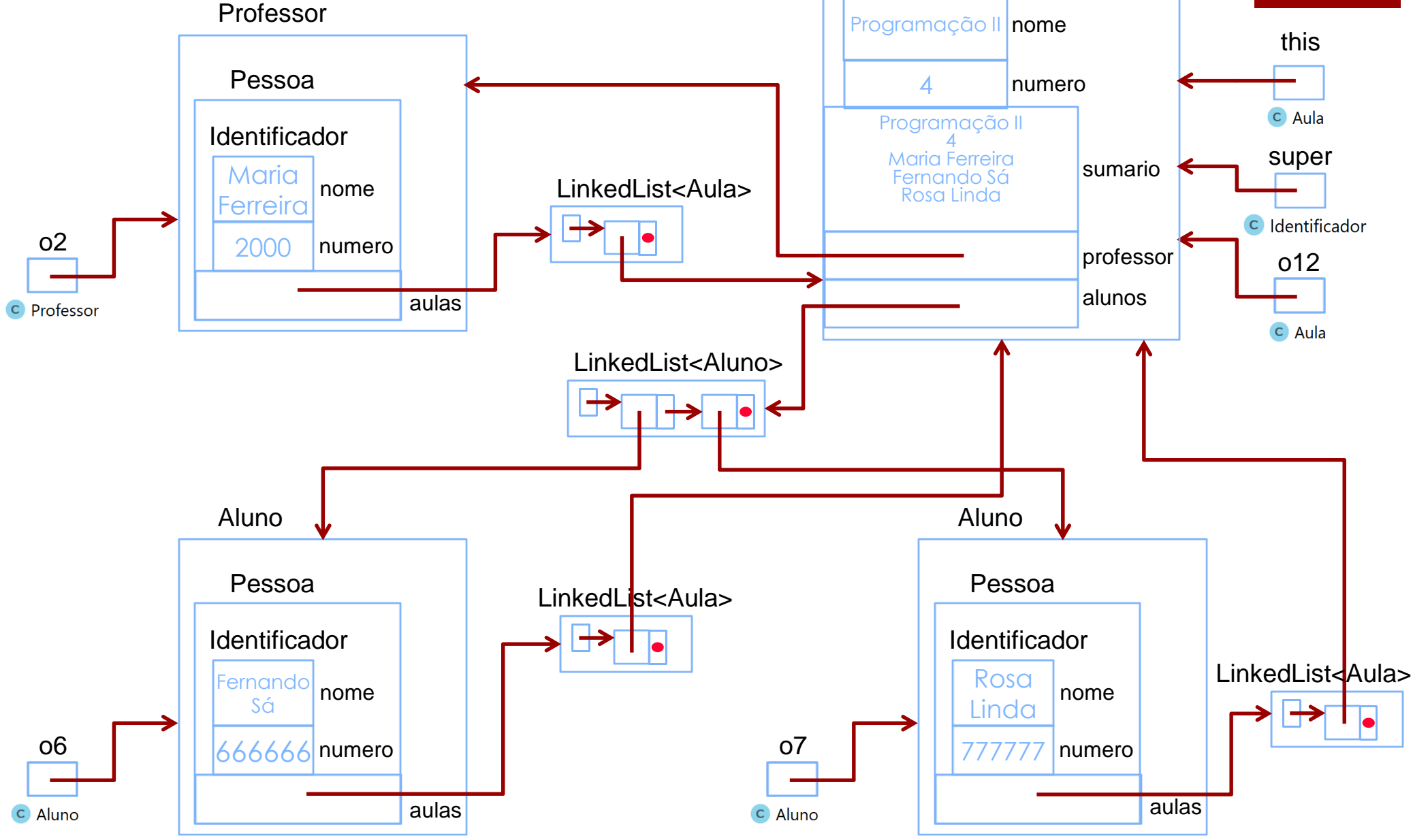
**C** Pessoa

 Aula

```
public void adicionarLinhaSumario(String linha) {
    StringBuilder sb = new StringBuilder(sumario);
    sb.append(linha).append("\n");
    sumario = sb.toString();
}
```

## 4.2. EXEMPLO

67



## 4.3. CASOS POSSÍVEIS

- Método(s) implementado(s) apenas na superclasse
- Método(s) reimplementado(s) em subclasse(s)
- Método(s) abstrato(s) na superclasse implementado(s) em subclasse(s)

## 4.4. VARIÁVEIS POLIMÓRFICAS

69

**UMA VARIÁVEL POLIMÓRFICA É  
DEFINIDA POR UM DADO TIPO E  
REFERENCIA OBJETOS DESSE TIPO OU  
DE QUALQUER SUBTIPO**

```
public Main() {  
    ...  
    Identificador id;  
    id = o12;  
    id = o2;  
    id = o6;  
  
    Pessoa p;  
    p = o2;  
    p = o6;  
}
```

**UPCAST**

## 4.4. VARIÁVEIS POLIMÓRFICAS

70

```
public Main() {
```

```
...
```

```
Identificador id;
```

```
id = o12;
```

```
id = o2;
```

```
id = o6;
```

```
id.
```

m	getNome()	String
---	-----------	--------

m	getNumero()	long
---	-------------	------

m	setNumero(long numero)	void
---	------------------------	------

Ctrl+Down and Ctrl+Up will move caret down and up in  $\pi$ .

```
}
```

**A VARIÁVEL  
POLIMÓRFICA `id` APENAS  
PERMITE O ACESSO AOS  
MÉTODOS PÚBLICOS DA  
CLASSE `Identificador`**

## 4.4. VARIÁVEIS POLIMÓRFICAS

71

```
public Main() {  
    ...  
    Identificador id;  
    id = o12;  
    id = o2;  
    id = o6;
```

```
    Pessoa p;  
    p = o2;  
    p = o7;  
    p.|
```

m	adicionar(Aula aula)	void
m	preencherSumario(Aula aula)	void
m	remover(Aula aula)	void
m	getNome()	String
m	getNumero()	long
m	setNumero(long numero)	void

Press Ctrl+Period to choose the selected (or first) suggestion π

```
}
```

**A VARIÁVEL  
POLIMÓRFICA **p** APENAS  
PERMITE O ACESSO AOS  
MÉTODOS PÚBLICOS DA  
CLASSE **Pessoa****

## 4.4. VARIÁVEIS POLIMÓRFICAS

72

```
public Main() {
```

```
...
```

```
Identificador id;
```

```
id = o12.;
```

```
id = m ▢ getProfessor() Professor
```

```
id = m ▢ adicionar(Aluno aluno) void
```

```
Pess m ▢ adicionarLinhaSumario(String linha) void
```

```
p = m ▢ desassociarProfessor() void
```

```
p = m ▢ getAlunos() LinkedList<Aluno>
```

```
m ▢ getSumario() String
```

```
m ▢ remover(Aluno aluno) void
```

```
m ▢ setProfessor(Professor professor) void
```

```
m ▢ getNome() String
```

```
m ▢ getNumero() long
```

```
m ▢ setNumero(long numero) void
```

```
Press Ctrl+Period to choose the selected (or first) suggestion and insert a dot after.. π:
```

```
}
```

**A VARIÁVEL o12  
PERMITE O ACESSO  
A TODOS OS  
MÉTODOS  
PÚBLICOS DA  
CLASSE Aula**



## 4.4. VARIÁVEIS POLIMÓRFICAS

73

```
public Main() {  
    ...  
    Identificador id;  
    id = o12;  
    id = o2.;  
    id  
    m adicionar(Aula aula) void  
    m getNome() String  
    p = m getNumero() long  
    p = m preencherSumario(Aula aula) void  
    m remover(Aula aula) void  
    m setNumero(long numero) void  
    Press Ctrl+Period to choose the selected (or first) suggestion π  
}
```

**A VARIÁVEL `o2` PERMITE O  
ACESSO A TODOS OS  
MÉTODOS PÚBLICOS DA  
CLASSE `Professor`**

## 4.4. VARIÁVEIS POLIMÓRFICAS

74

```
public Main() {
```

```
...
```

```
Identificador id;
```

```
id = o12;
```

```
id = o2;
```

```
id = o6.;
```

m	adicionar(Aula aula)	void
m	getNome()	String
m	getNumero()	long
m	preencherSumario(Aula aula)	void
m	remover(Aula aula)	void
m	setNumero(long numero)	void

Ctrl+Down and Ctrl+Up will move caret down and up in the .π:

```
}
```

**A VARIÁVEL `o6` PERMITE O  
ACESSO A TODOS OS  
MÉTODOS PÚBLICOS DA  
CLASSE `Aluno`**

VAMOS AGORA VER  
UM EXEMPLO

## 4.4. VARIÁVEIS POLIMÓRFICAS

76

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

77

```
public Main() {  
    ***  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```

**CRIAÇÃO DE UMA LISTA DE  
IDENTIFICADORES ONDE SÃO  
ADICIONADOS AULAS,  
PROFESSORES E ALUNOS**

## 4.4. VARIÁVEIS POLIMÓRFICAS

78

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```

ESCREVE O NOME DE CADA  
IDENTIFICADOR DA LISTA

QUE NOME DEVE ESCREVER?

## 4.4. VARIÁVEIS POLIMÓRFICAS

79

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```

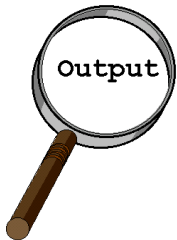
**1º VERIFICA SE O MÉTODO `getNome` EXISTE NA CLASSE `Identificador`**

**2º COMEÇA A PROCURA DO MÉTODO `getNome` NA CLASSE DO OBJETO REFERENCIADO POR `identificador`**

## 4.4. VARIÁVEIS POLIMÓRFICAS

80

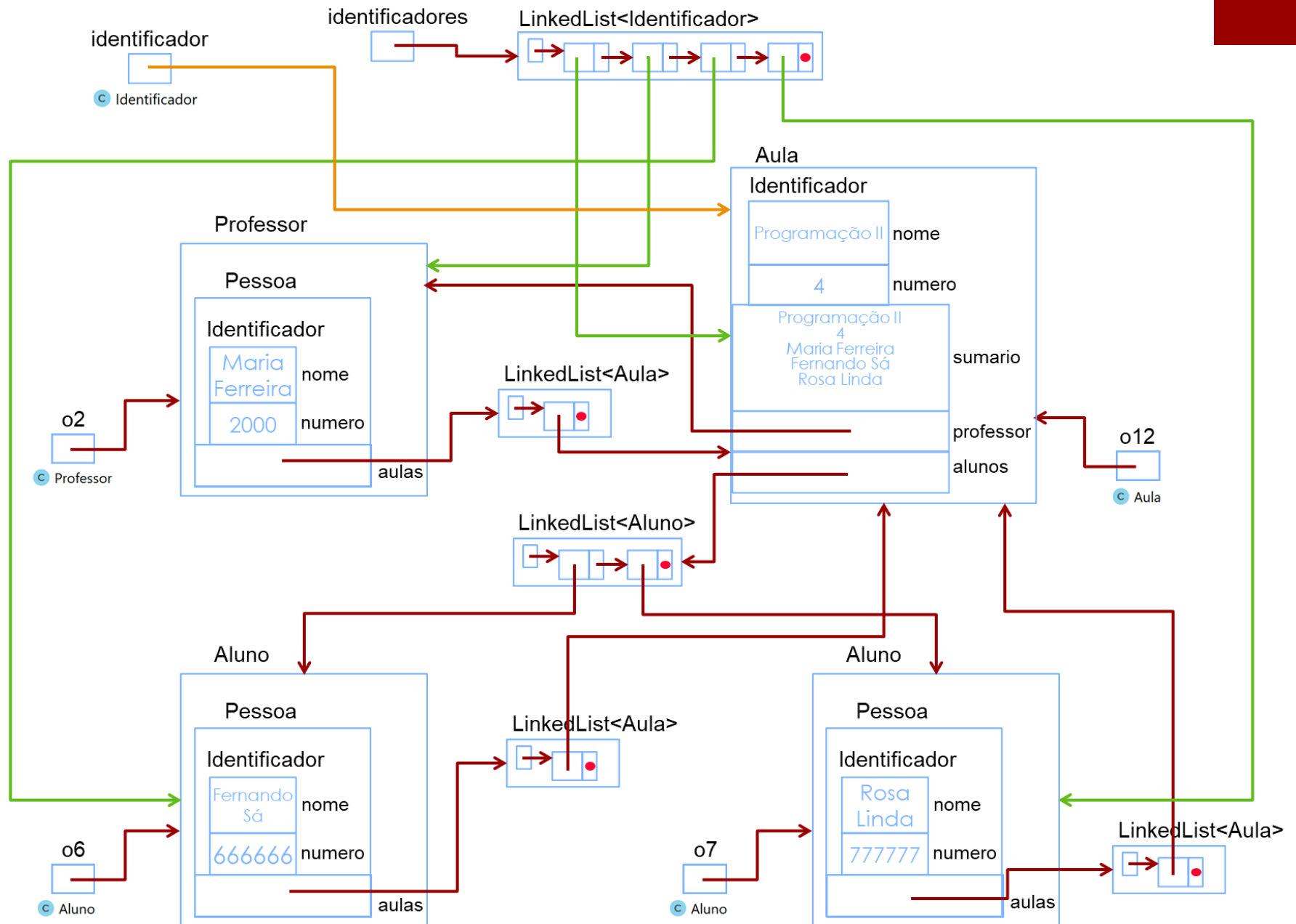
```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 1  
    }  
}
```





# 4.4. VARIÁVEIS POLIMÓRFICAS

81

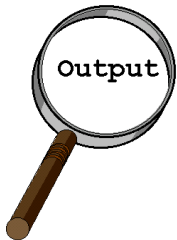


## 4.4. VARIÁVEIS POLIMÓRFICAS

82

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 1  
    }  
}
```

Programação II

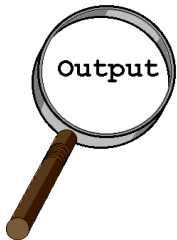


## 4.4. VARIÁVEIS POLIMÓRFICAS

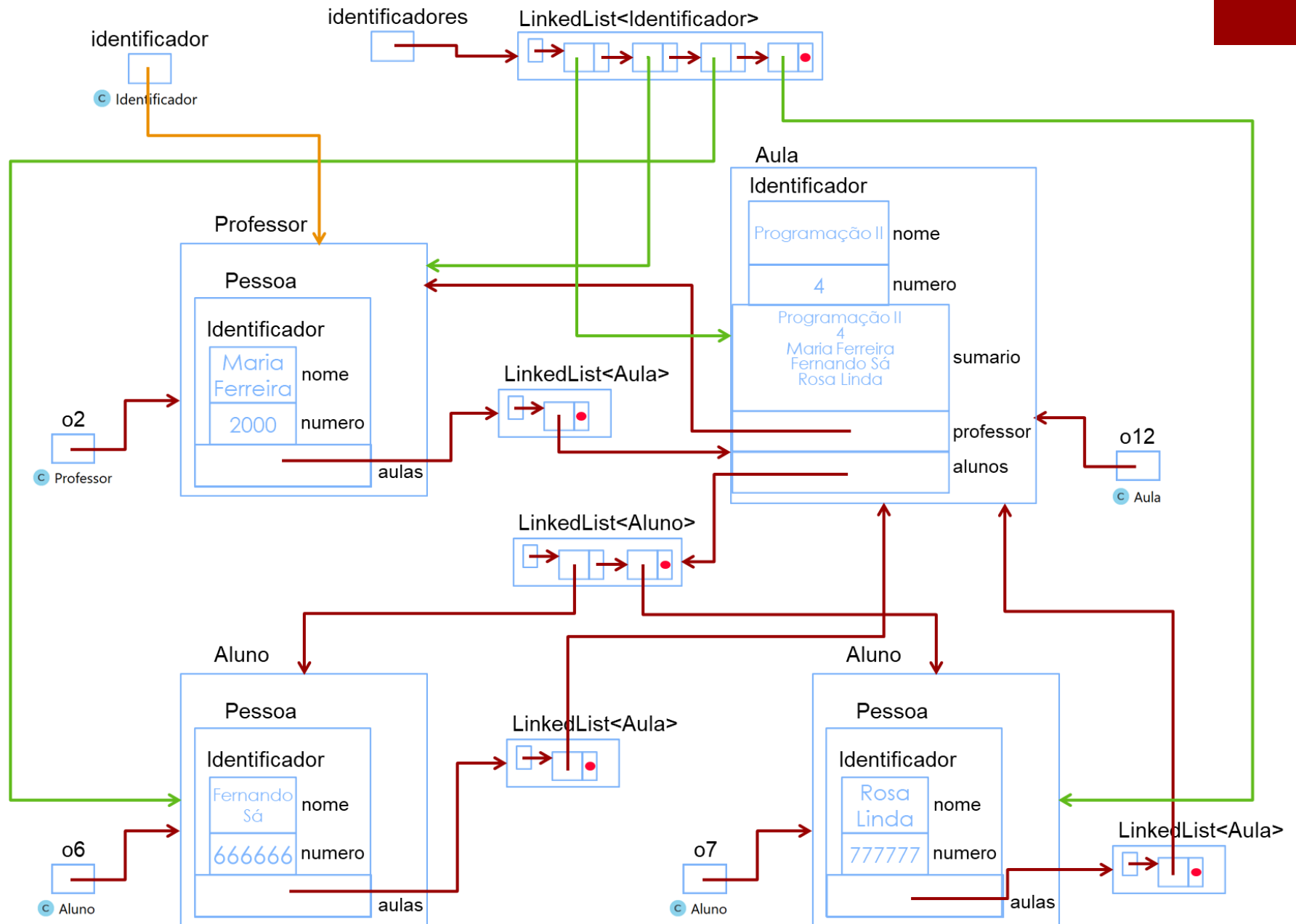
83

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 2  
    }  
}
```

Programação II



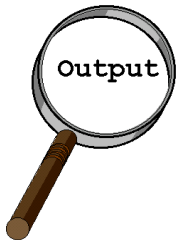
## 84



## 4.4. VARIÁVEIS POLIMÓRFICAS

85

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 2  
    }  
}
```

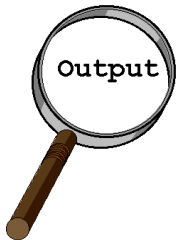


Programação II  
Maria Ferreira

## 4.4. VARIÁVEIS POLIMÓRFICAS

86

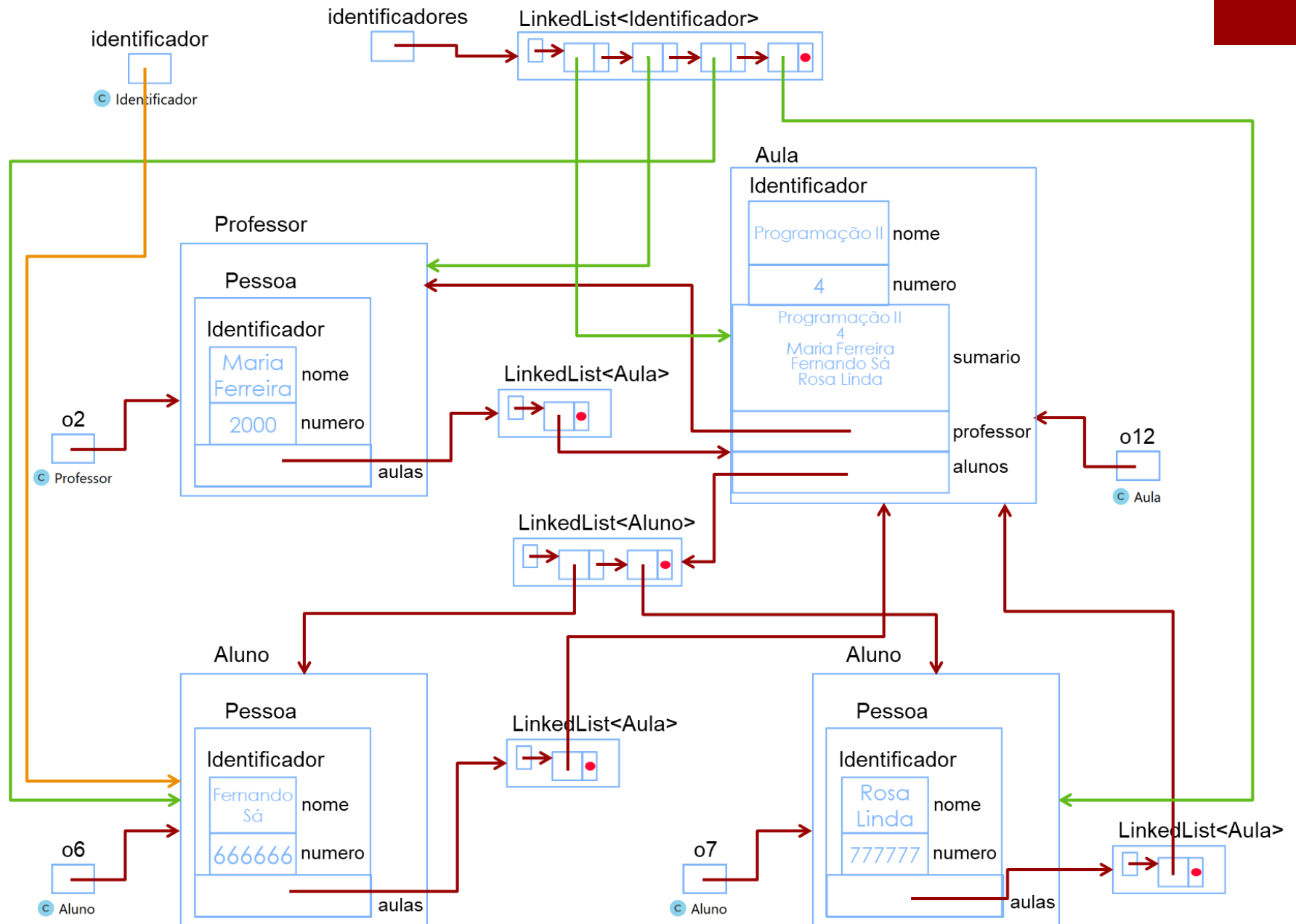
```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 3  
    }  
}
```



Programação II  
Maria Ferreira

# 4.4. VARIÁVEIS POLIMÓRFICAS

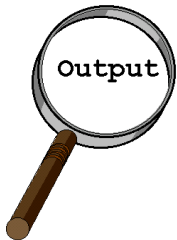
87



## 4.4. VARIÁVEIS POLIMÓRFICAS

88

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome()); 3  
    }  
}
```



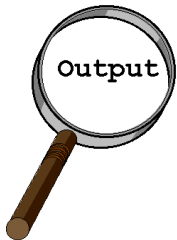
Programação II  
Maria Ferreira  
Fernando Sá



## 4.4. VARIÁVEIS POLIMÓRFICAS

89

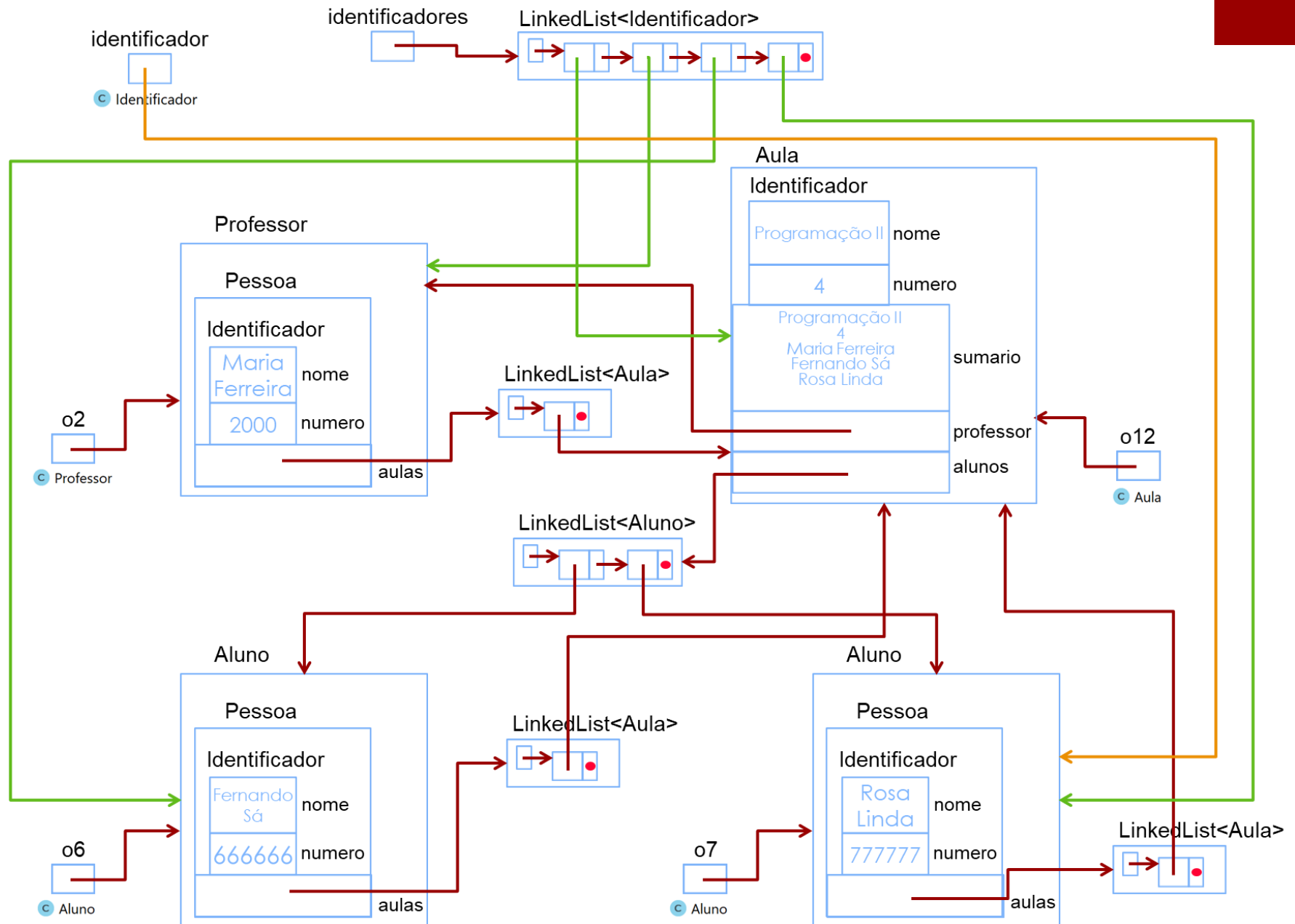
```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```



Programação II  
Maria Ferreira  
Fernando Sá

# 4.4. VARIÁVEIS POLIMÓRFICAS

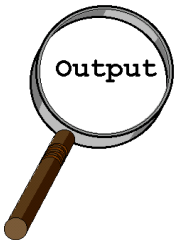
90



## 4.4. VARIÁVEIS POLIMÓRFICAS

91

```
public Main() {  
    ...  
    LinkedList<Identificador> identificadores = new LinkedList<>();  
    identificadores.add(o12);  
    identificadores.add(o2);  
    identificadores.add(o6);  
    identificadores.add(o7);  
  
    for (Identificador identificador : identificadores) {  
        System.out.println(identificador.getNome());  
    }  
}
```



Programação II  
Maria Ferreira  
Fernando Sá  
Rosa Linda

## 4.4. VARIÁVEIS POLIMÓRFICAS

```
public Main() {  
    ...  
    for (Identificador identificador : identificadores) {  
        identificador.adicionar(o11);  
    }  
}
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

93

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        identificador.adicionar(o11);  
    }  
}
```

**COMO ADICIONAR UMA AULA  
A UM IDENTIFICADOR DO TIPO  
Aula?**

## 4.4. VARIÁVEIS POLIMÓRFICAS

94

```
public Main() {
```

```
    for (Identificador identificador : identificadores) {  
        identificador.adicionar(o11);  
    }
```

Cannot resolve method 'adicionar(cap4\_polimorfismo.modelo.Aula)'

```
}
```

**VERIFICA QUE O MÉTODO adicionar NÃO EXISTE NA CLASSE Identificador**

**NESTE CASO OCORRE UM ERRO DE COMPILAÇÃO**

### PROBLEMA:

**ADICIONAR UMA AULA APENAS ÀS  
PESSOAS (PROFESSORES E ALUNOS) DE  
UMA LISTA DE IDENTIFICADORES**

## 4.4. VARIÁVEIS POLIMÓRFICAS

96

### SOLUÇÃO 1:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```



## 4.4. VARIÁVEIS POLIMÓRFICAS

97

### SOLUÇÃO 1:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

**CONSIDERA CADA** identificador **COMO** Pessoa

**ADICIONANDO A AULA A CADA UMA DESSAS PESSOAS**

## 4.4. VARIÁVEIS POLIMÓRFICAS

98

### SOLUÇÃO 1:

```
public Main() {  
    ...  
    for (Identificador identificador : identificadores) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

**DOWNCAST**

## 4.4. VARIÁVEIS POLIMÓRFICAS

99

### SOLUÇÃO 1:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

**COMO CONSIDERAR UMA  
AULA identificador COMO  
Pessoa?**

## 4.4. VARIÁVEIS POLIMÓRFICAS

100

### SOLUÇÃO 1:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

**QUANDO O identificador REFERENCIA UMA PESSOA É EFETUADO O DOWNCAST DE identificador PARA A CLASSE Pessoa**

**CASO O identificador REFERENCIE UMA AULA OCORRE UM ERRO DE EXECUÇÃO AO TENTAR EFETUAR O DOWNCAST DE identificador PARA A CLASSE Pessoa**

```
java.lang.ClassCastException: cap4_polimorfismo.modelo.Aula cannot be cast to cap4_polimorfismo.modelo.Pessoa  
at cap4_polimorfismo.vista.Main.<init>(Main.java:97)|  
at cap4_polimorfismo.vista.Main.main(Main.java:103)
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

101

### SOLUÇÃO 2:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        if (identificador instanceof Pessoa) {  
            ((Pessoa) identificador).adicionar(o11);  
        }  
    }  
}
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

102

### SOLUÇÃO 2:

```
public Main() {  
    ***  
    for (Identificador identificador : identificadores) {  
        if (identificador instanceof Pessoa) {  
            ((Pessoa) identificador).adicionar(o11);  
        }  
    }  
}
```

**APENAS CONSIDERA COMO Pessoa CADA identificador  
QUE SEJA INSTÂNCIA DA CLASSE Pessoa**

**ADICIONANDO A CADA UMA DESSAS PESSOAS A AULA**

## 4.4. VARIÁVEIS POLIMÓRFICAS

103

### SOLUÇÃO 3:

```
public Main() {  
    ***  
    LinkedList<Identificador> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Identificador identificador : pessoas) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

104

### SOLUÇÃO 3:

```
public Main() {  
    ***  
    LinkedList<Identificador> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Identificador identificador : pessoas) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
}
```

**CONSIDERAR UMA LISTA DE IDENTIFICADORES  
APENAS COM PESSOAS**



## 4.4. VARIÁVEIS POLIMÓRFICAS

105

### SOLUÇÃO 3:

```
public Main() {  
    ...  
    LinkedList<Identificador> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Identificador identificador : pessoas) {  
        ((Pessoa) identificador).adicionar(o11);  
    }  
  
    pessoas.add(o9);  
}
```

**É POSSÍVEL ADICIONAR UMA AULA A UMA LISTA DE IDENTIFICADORES**

## 4.4. VARIÁVEIS POLIMÓRFICAS

106

### SOLUÇÃO 4:

```
public Main() {
```

```
    LinkedList<Pessoa> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Pessoa pessoa : pessoas) {  
        pessoa.adicionar(o11);  
    }
```

```
}
```

## 4.4. VARIÁVEIS POLIMÓRFICAS

107

### SOLUÇÃO 4:

```
public Main() {  
    ***  
    LinkedList<Pessoa> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Pessoa pessoa : pessoas) {  
        pessoa.adicionar(o11);  
    }  
}
```

**CONSIDERAR APENAS UMA LISTA DE PESSOAS**

## 4.4. VARIÁVEIS POLIMÓRFICAS

108

### SOLUÇÃO 4:

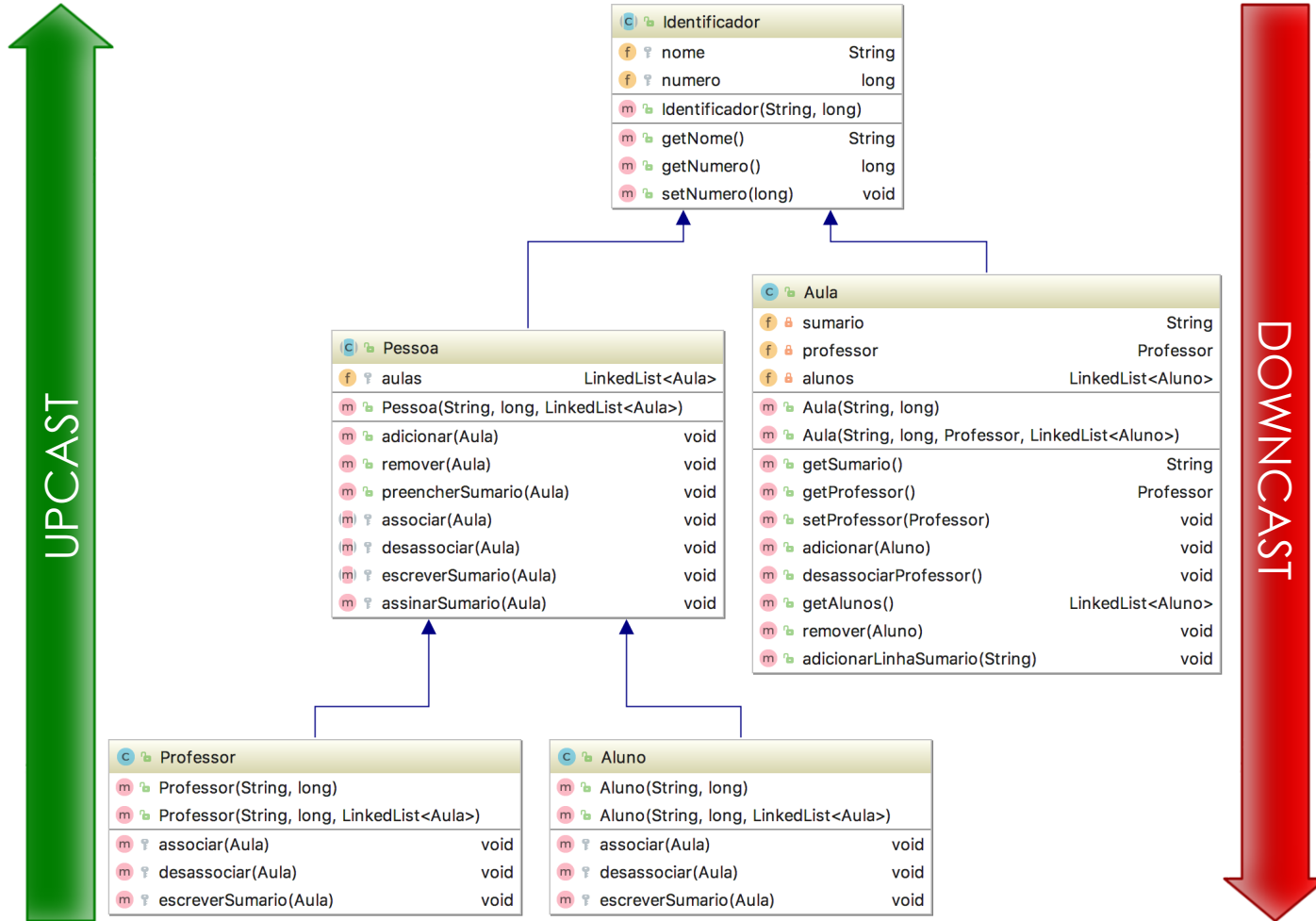
```
public Main() {  
    ...  
    LinkedList<Pessoa> pessoas = new LinkedList<>();  
    pessoas.add(o2);  
    pessoas.add(o6);  
    pessoas.add(o7);  
  
    for (Pessoa pessoa : pessoas) {  
        pessoa.adicionar(o11);  
    }  
  
    pessoas.add(o9);  
  
    add (cap4_polimorfismo.modelo.Pessoa) in LinkedList cannot be applied  
    to (cap4_polimorfismo.modelo.Aula)  
}
```

**NÃO É POSSÍVEL ADICIONAR UMA AULA A  
UMA LISTA DE PESSOAS**

# 4.4. VARIÁVEIS POLIMÓRFICAS

109

AUTOMÁTICO  
(APLICAÇÃO DO  
POLIMORFISMO)  
  
SEMPRE SEGURO



EXPLÍCITO  
  
SEMPRE  
VERIFICADO