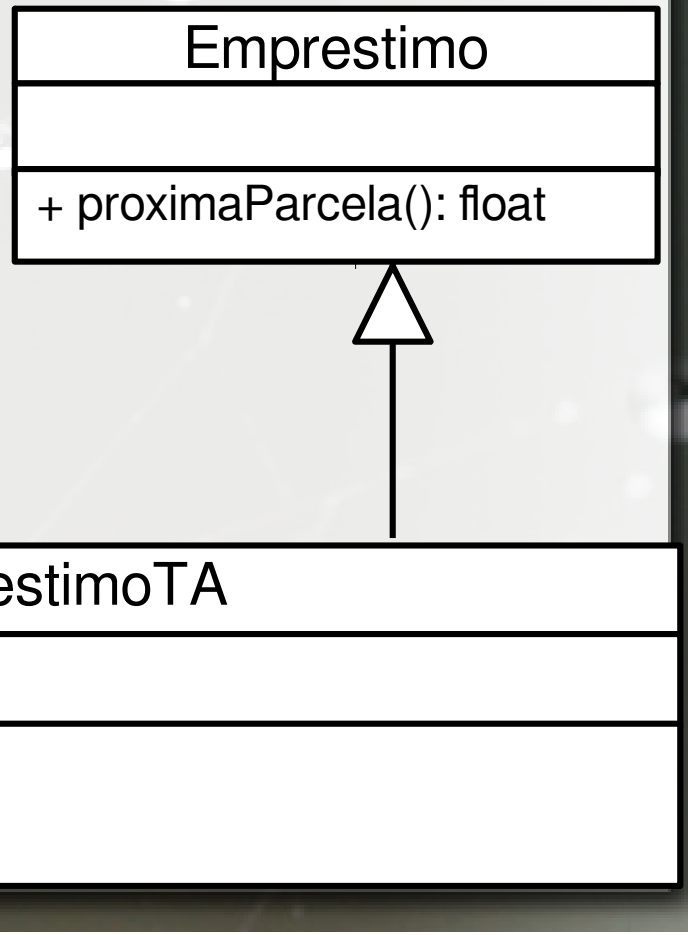
Programação Orientada a Objetos

Amarração

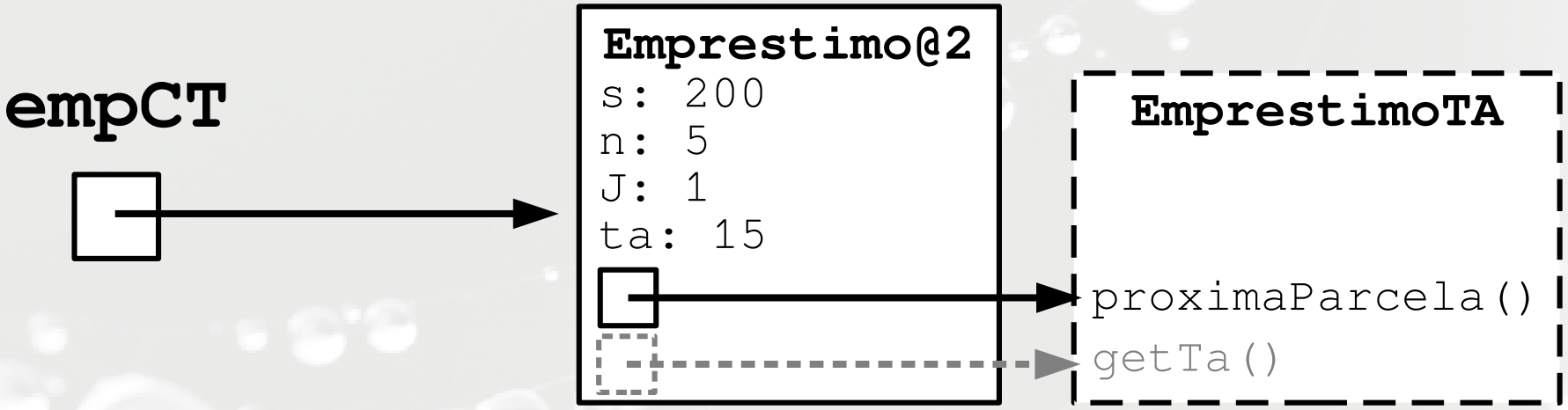
André Santanchè
Laboratory of Information Systems – LIS
Instituto de Computação – UNICAMP
Maio 2020

Retornando ao Empréstimo

```
public class Emprestimo {

    ...

    public float proximaParcela() { … }
}


public class EmprestimoTA extends Emprestimo {
    ...

    public float proximaParcela() { … }
}
```

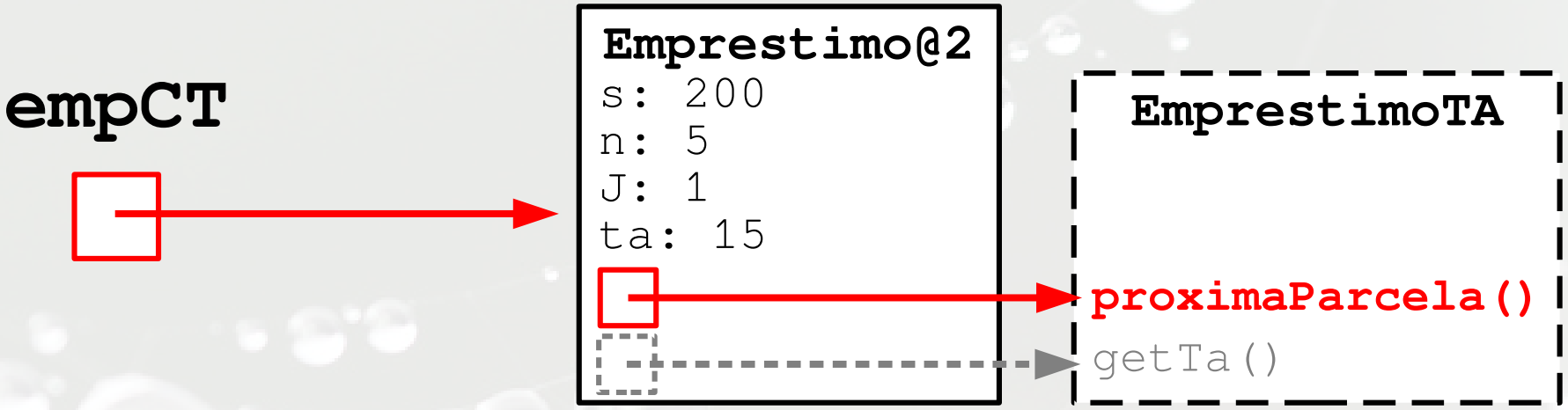


Exemplo Anterior




```
Emprestimo empCT = new EmprestimoTA(200, 5, 1, 15);
```

Exemplo Anterior




```
float pct = empCT.proximaParcela();
```

Instanciação Condicionada

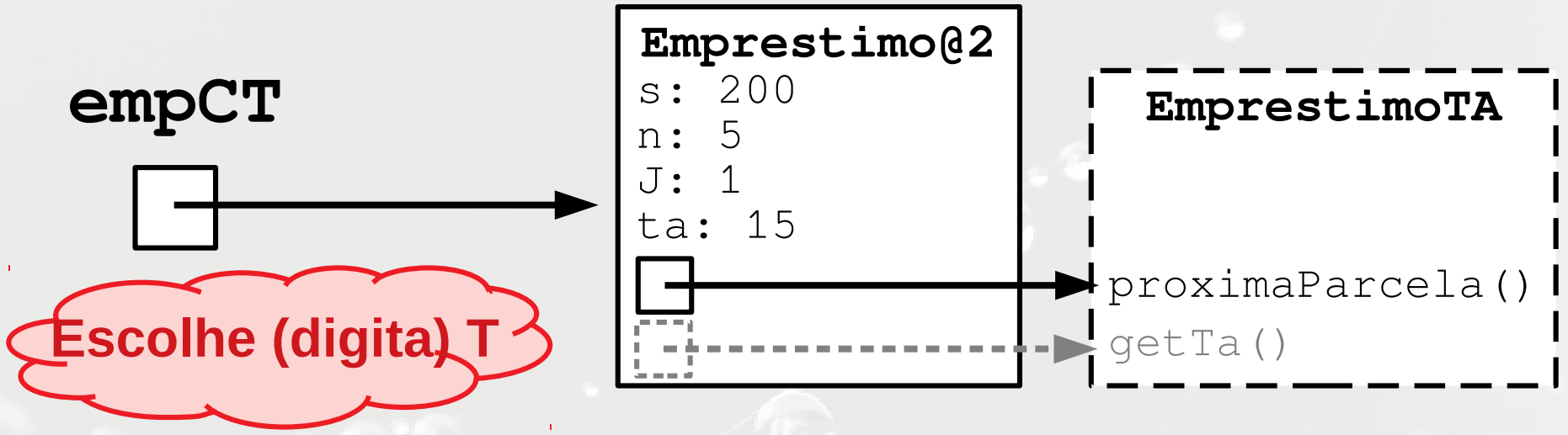
```
Emprestimo emp;

System.out.print("Digite T para emprestimo com taxa e N sem: ");
Scanner teclado = new Scanner(System.in);
String tipo = teclado.nextLine();
teclado.close();

if (tipo.equalsIgnoreCase("T"))
   emp = new EmprestimoTA(200, 5, 1, 15);
else
   emp = new Emprestimo(200, 5, 1);
```

```
int i = 1;
float pct = emp.proximaParcela();
while (pct > 0) {
   if (pct > 0)
      System.out.println(... pct);
   pct = emp.proximaParcela();
   i++;
}
```

Tempo de Execução


```
Emprestimo emp;
```

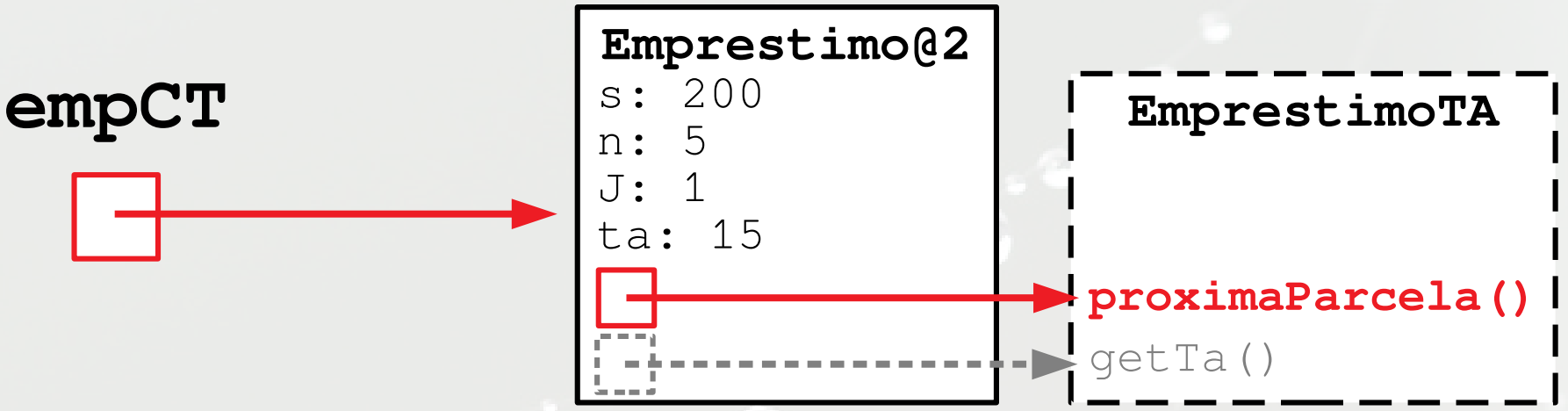
Tempo de Execução




```
Emprestimo emp;
```

```
… emp = new EmprestimoTA(200, 5, 1, 15);
```

Tempo de Execução

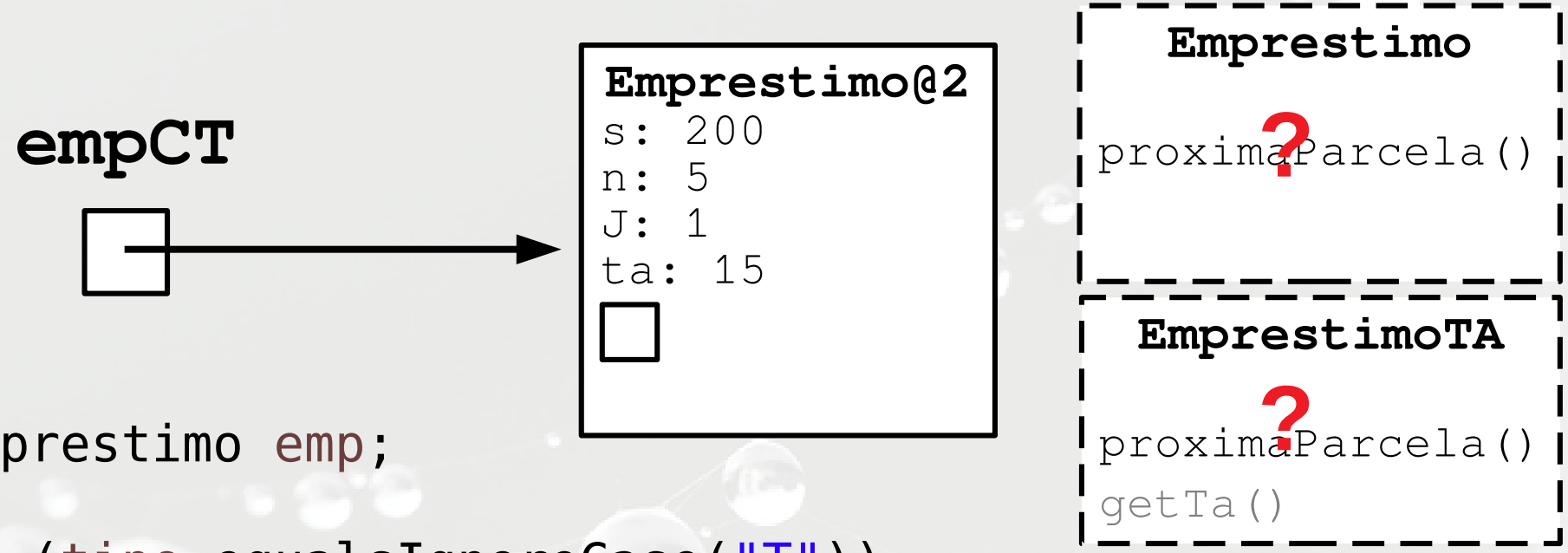



```
Emprestimo emp;

   … emp = new EmprestimoTA(200, 5, 1, 15);

float pct = emp.proximaParcela();
```

Tempo de Compilação

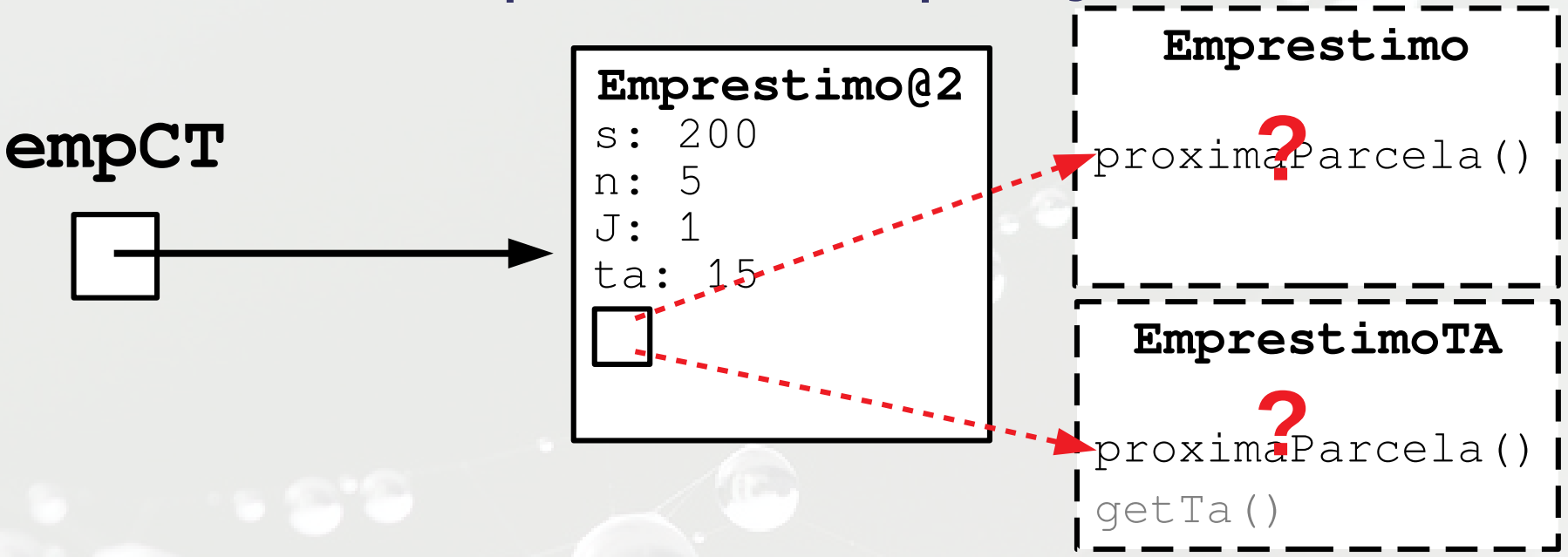

```
Emprestimo emp;
```

Tempo de Compilação




```
Emprestimo emp;

if (tipo.equalsIgnoreCase("T"))
    emp = new EmprestimoTA(200, 5, 1, 15);
else
    emp = new Emprestimo(200, 5, 1);
```

Tempo de Compilação




```
float pct = emp.proximaParcela();
```

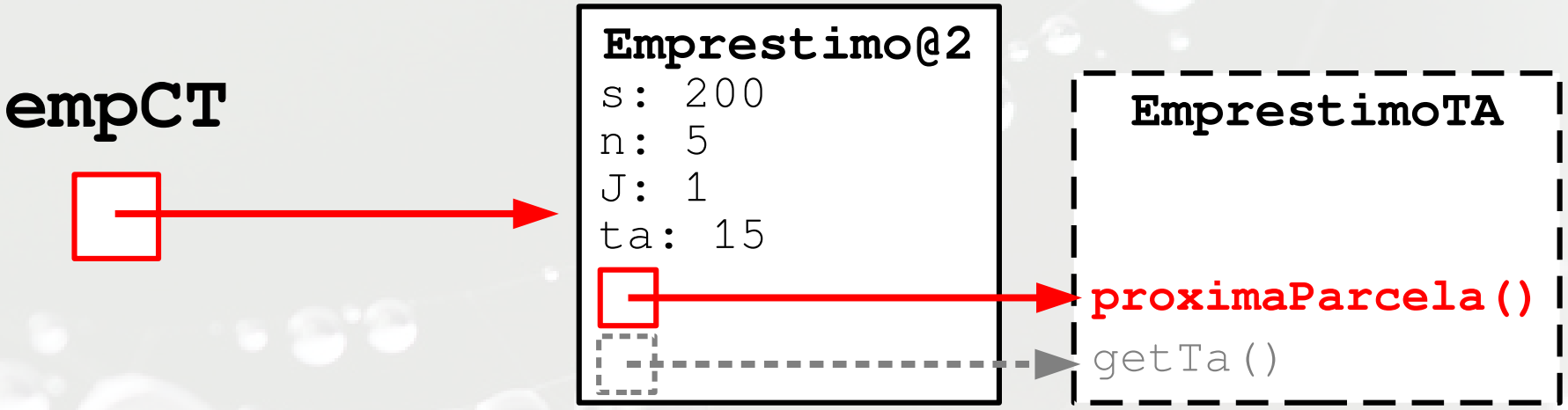
Amarração

- Amarração: ligação da chamada de um método ao método
- A decisão de quando ligar depende da amarração:
  - Estática
  - Dinâmica

Amarração Estática x Dinâmica

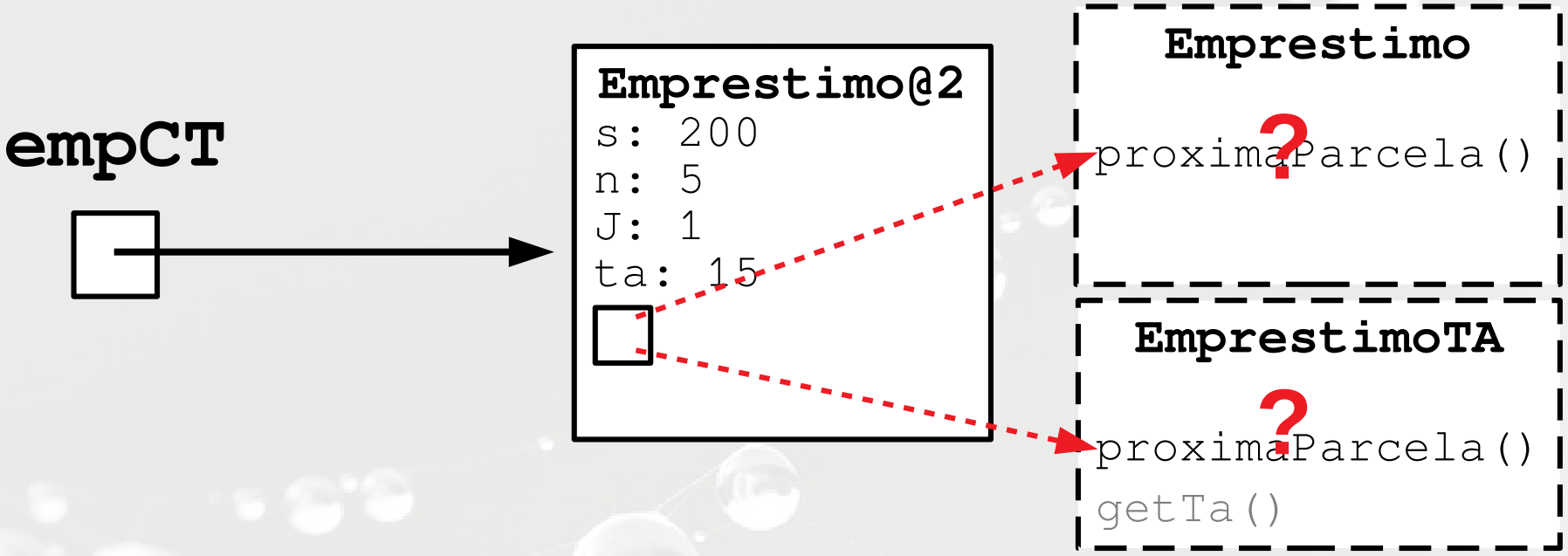
- **Amarração estática (static bindind)**: realiza a ligação no momento da compilação
- **Amarração dinâmica ou tardia (dynamic or late binding**): realiza a ligação no momento da execução (ligação tardia)

Amarração Dinâmica



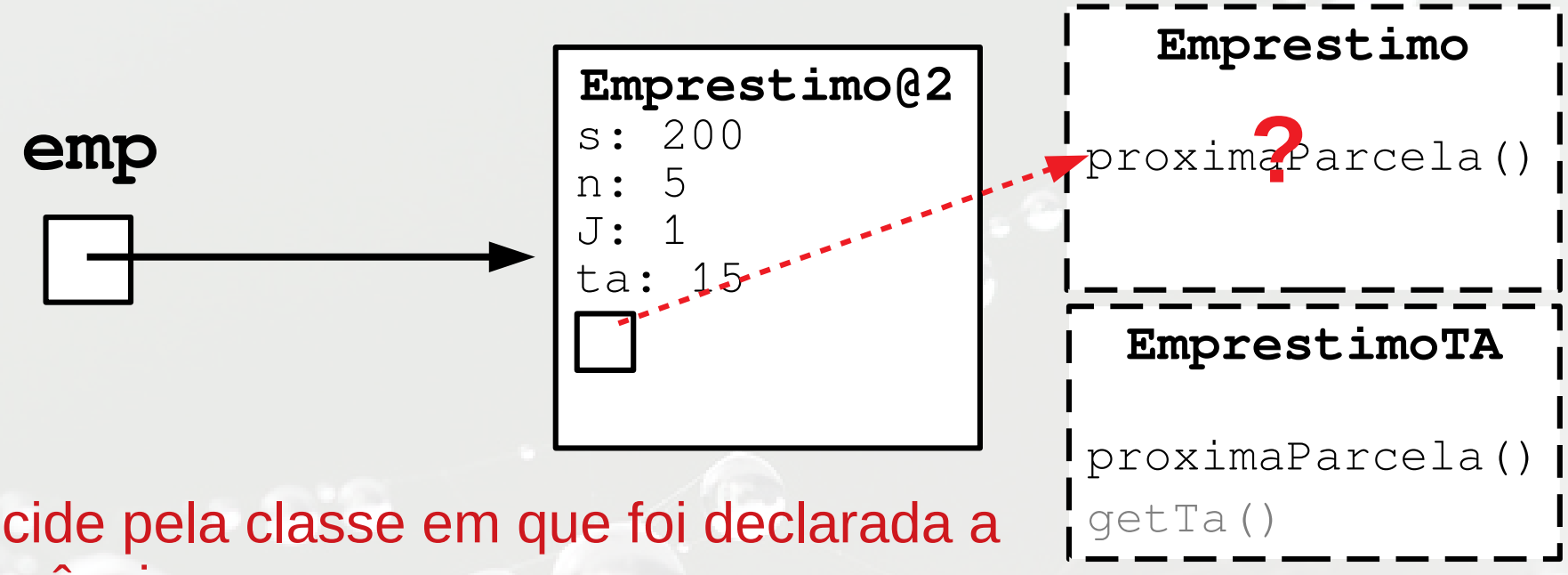

```
float pct = empCT.proximaParcela();
```

Amarração Estática




```
Emprestimo emp;
...
float pct = emp.proximaParcela();
```

Amarração Estática



Decide pela classe em que foi declarada a referência.

```
Emprestimo emp;
...
float pct = emp.proximaParcela();
```

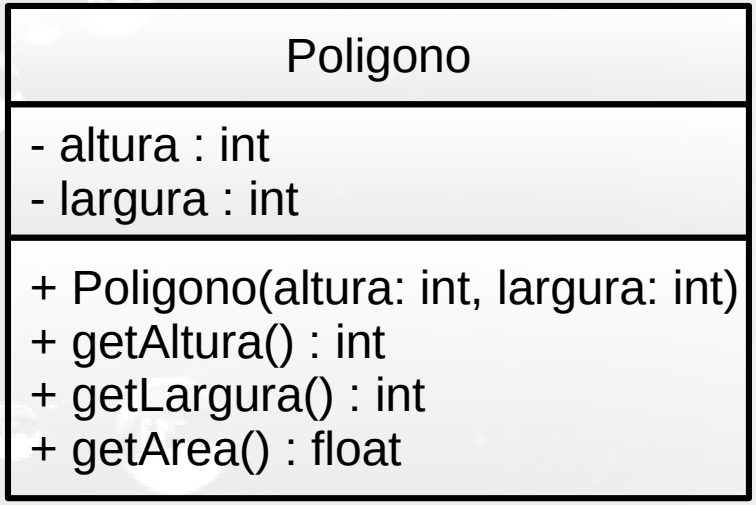
Amarração Estática x Dinâmica

- **Amarração dinâmica**
  - reflete a intenção do usuário
  - adotado pelas liguagens OO modernas
    - Java, Python, JavaScript...
- **Amarração estática**
  - execução mais rápida
    - evita escolha em tempo de execução
  - opção que pode ser escolhida em C++ para desempenho

Amarração Estática x Dinâmica em C++
Polígono

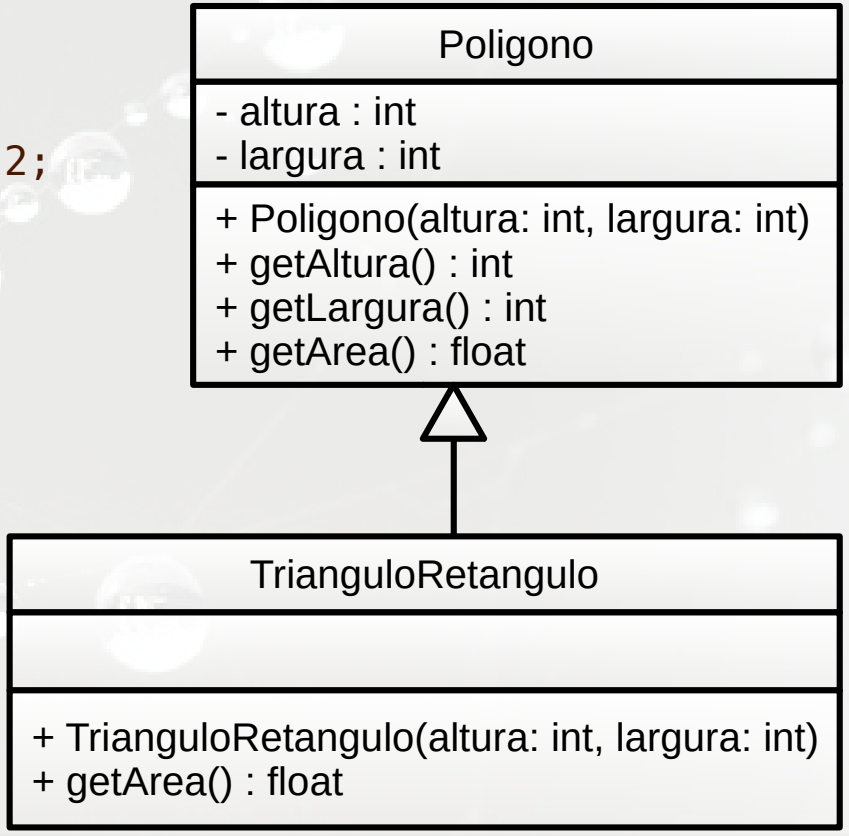
Polígono C++

```
class Poligono {
    private:
       int altura;
       int largura;
    public:
       Poligono(int altura, int largura) {
           this->altura = altura;
           this->largura = largura;
       }

       int getAltura() {
           return altura;
       }

       int getLargura() {
           return largura;
       }

       float getArea() {
           return 0;
       }
};
```



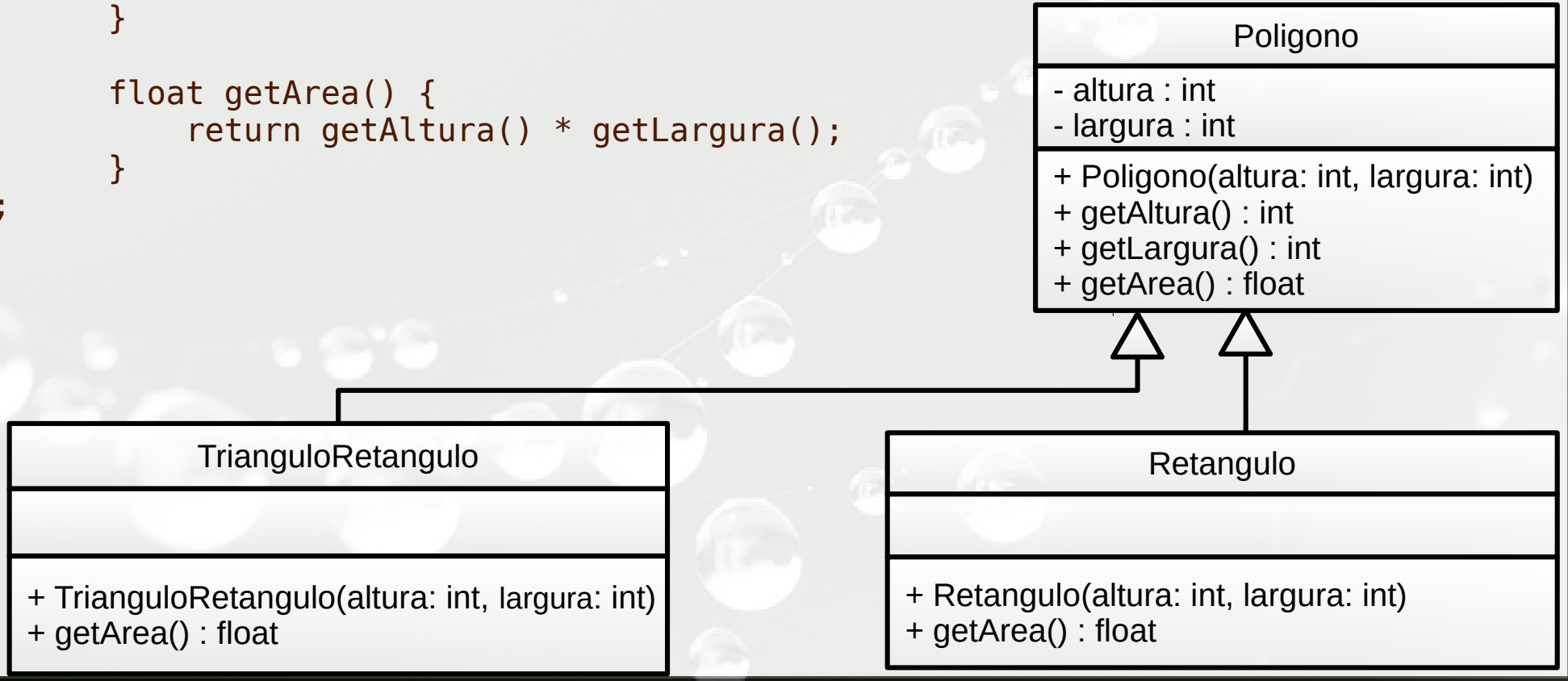
Triângulo Retângulo C++

```
class TrianguloRetangulo : public Poligono {
    public:
       TrianguloRetangulo(int altura, int largura) : Poligono(altura, largura){
       }

       float getArea() {
           return getAltura() * getLargura() / 2;
       }
};
```



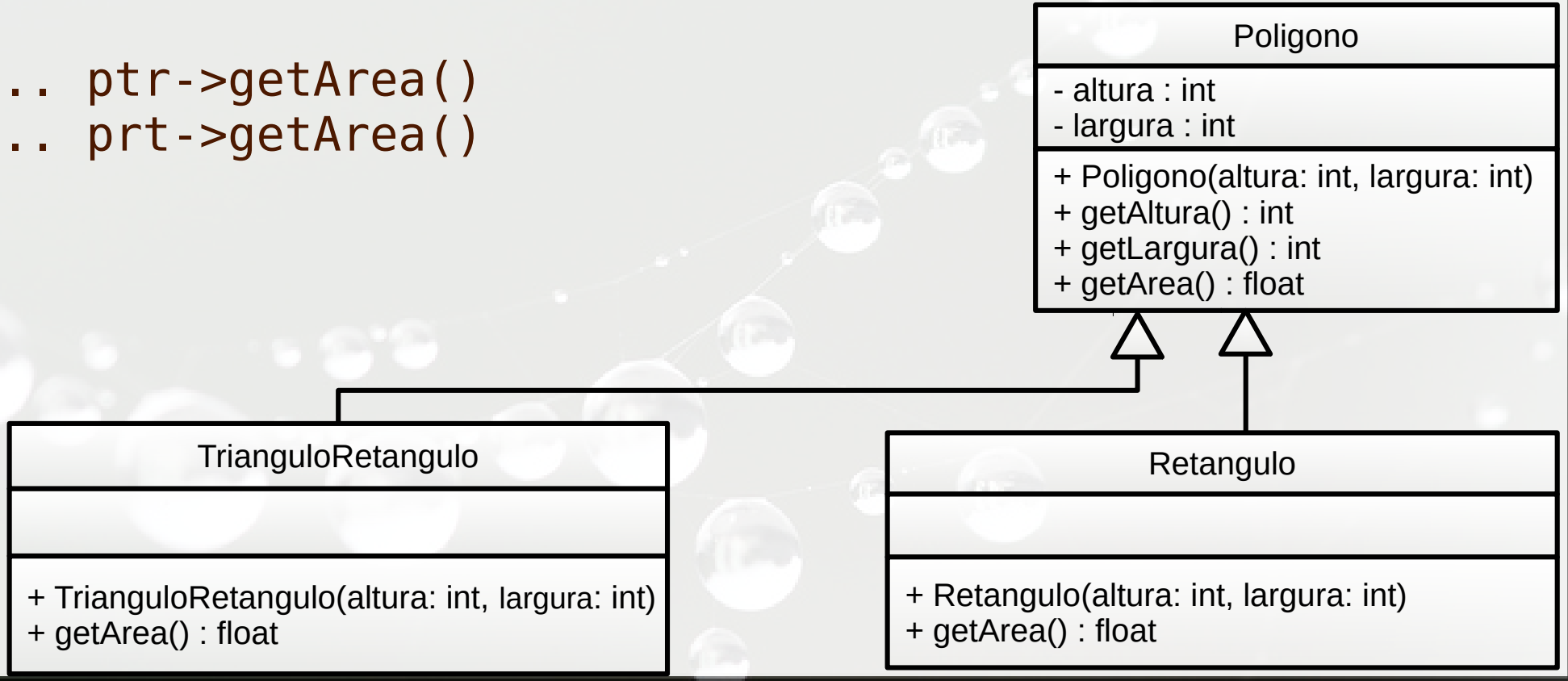
Retângulo C++

```
class Retangulo : public Poligono {
    public:
       Retangulo(int altura, int largura) : Poligono(altura, largura) {
       }

       float getArea() {
           return getAltura() * getLargura();
       }
};
```



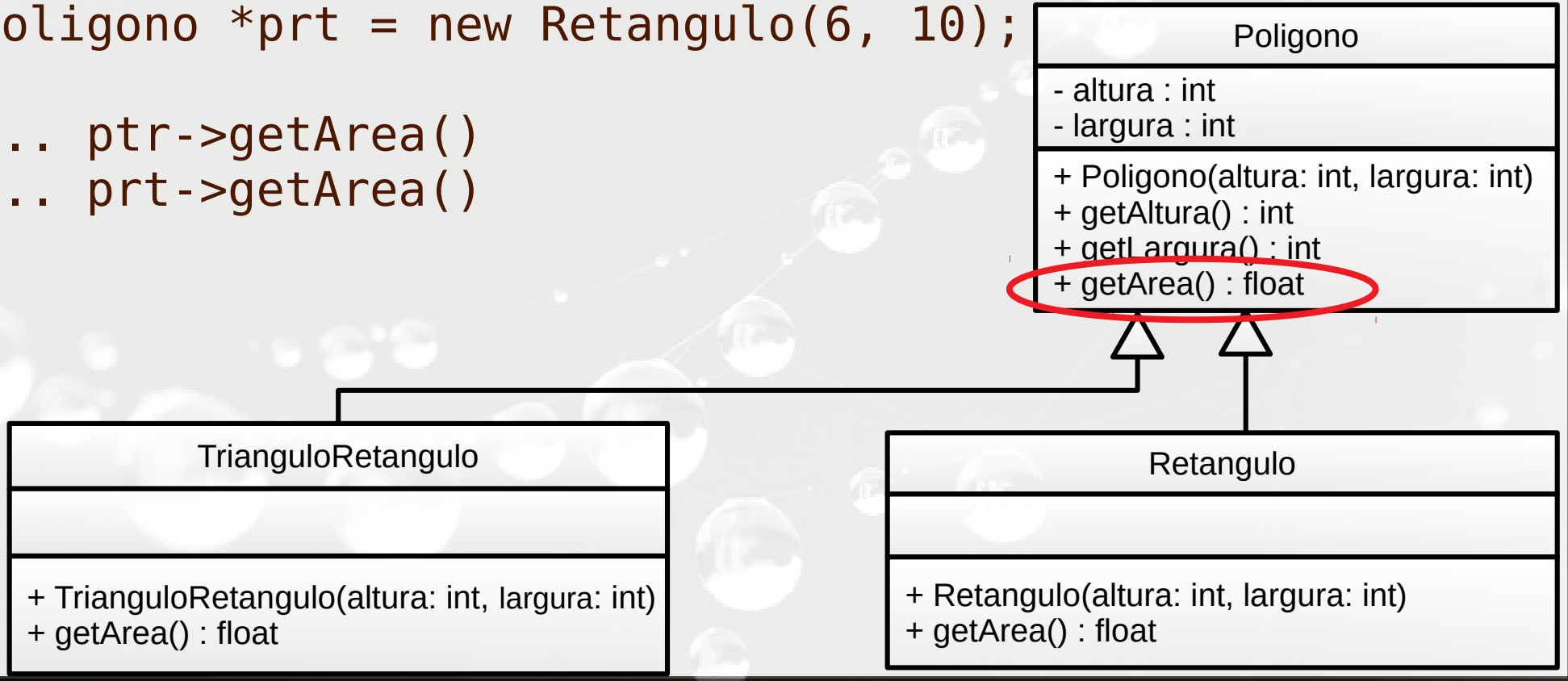
Polígono C++

```
Poligono *ptr = new TrianguloRetangulo(6, 10);
Poligono *prt = new Retangulo(6, 10);

... ptr->getArea()
... prt->getArea()
```



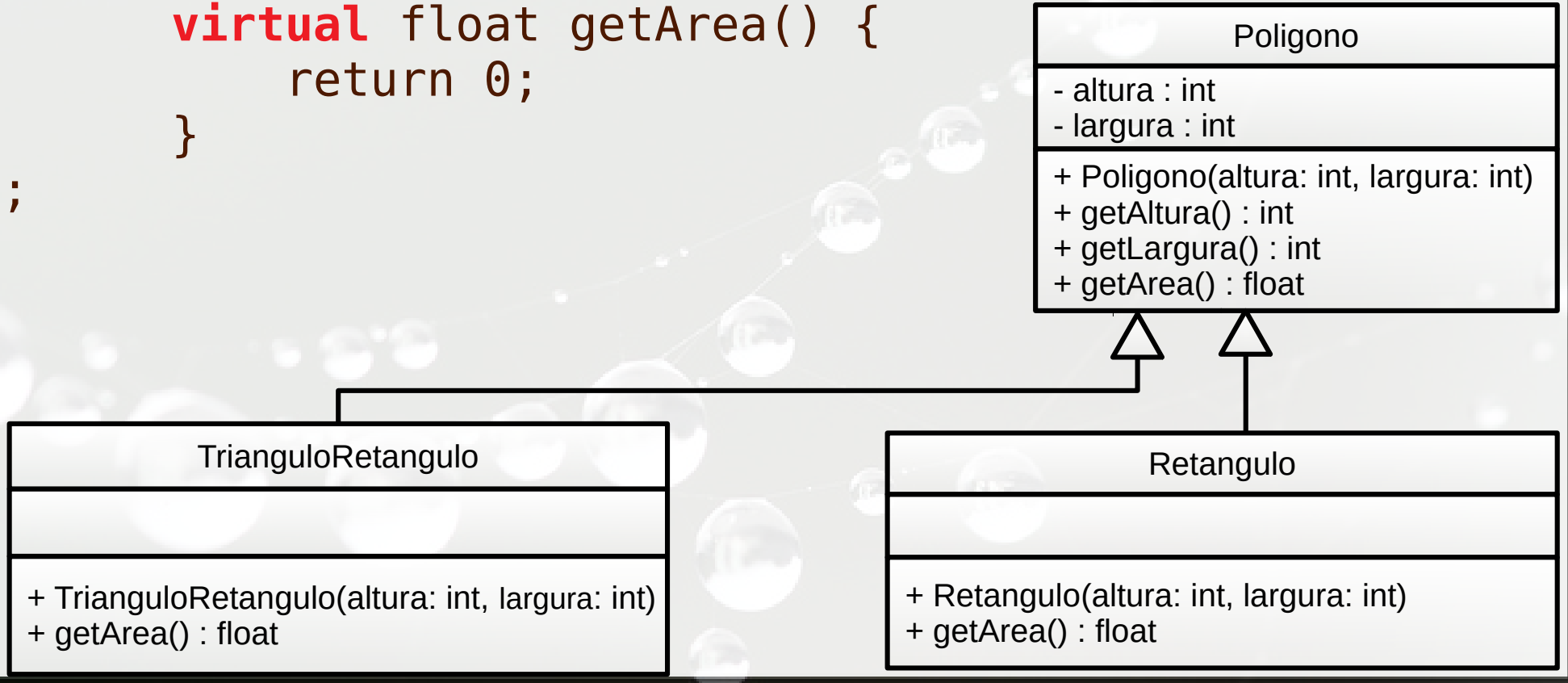
Polígono C++
Amarração Estática

```
Poligono *ptr = new TrianguloRetangulo(6, 10);
Poligono *prt = new Retangulo(6, 10);

... ptr->getArea()
... prt->getArea()
```



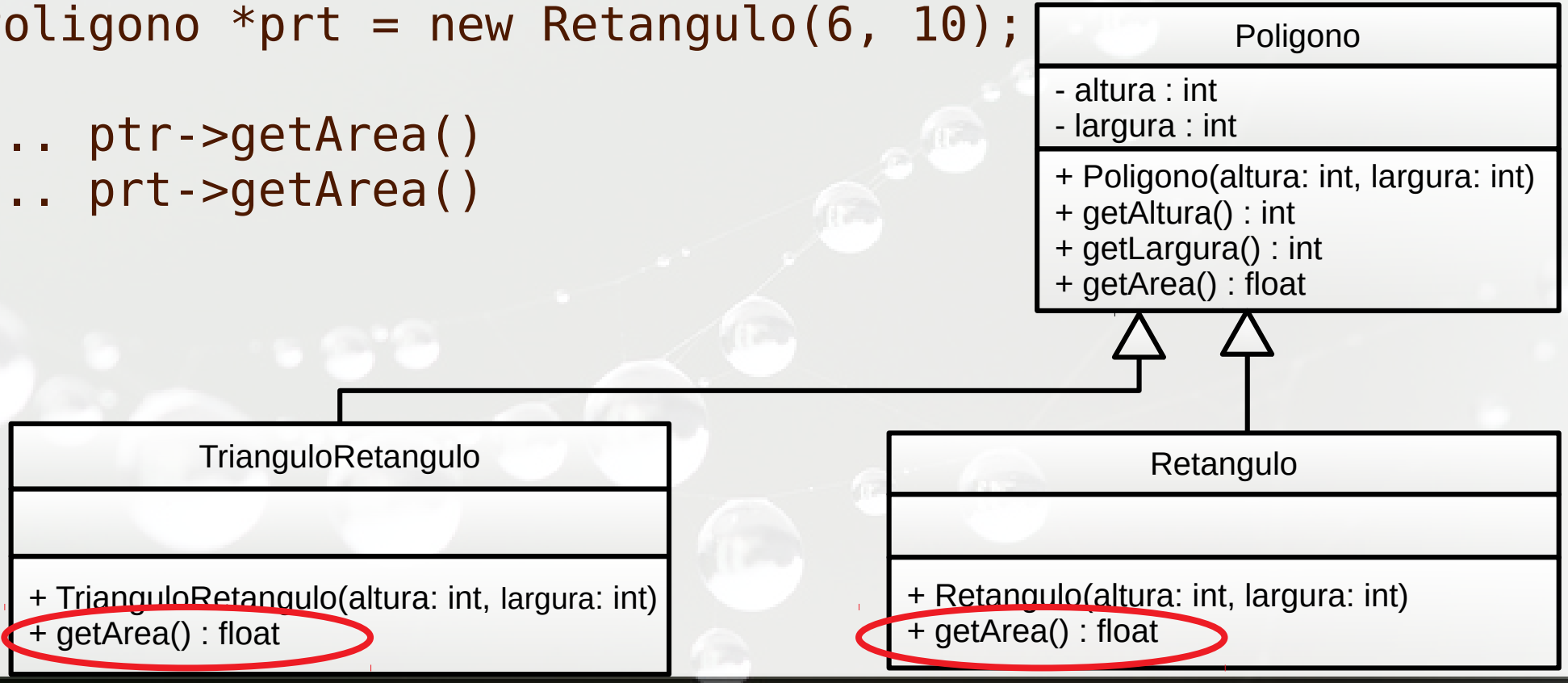
Polígono C++

```
class PoligonoV {
        ...
        virtual float getArea() {
            return 0;
        }
};
```



Polígono C++
Amarração Dinâmica

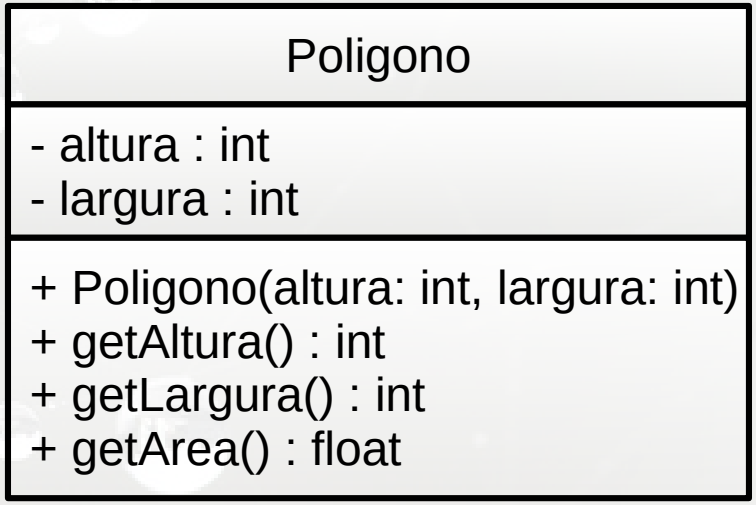
```
Poligono *ptr = new TrianguloRetangulo(6, 10);
Poligono *prt = new Retangulo(6, 10);

... ptr->getArea()
... prt->getArea()
```



Amarração Estática x Dinâmica em Java
Polígono

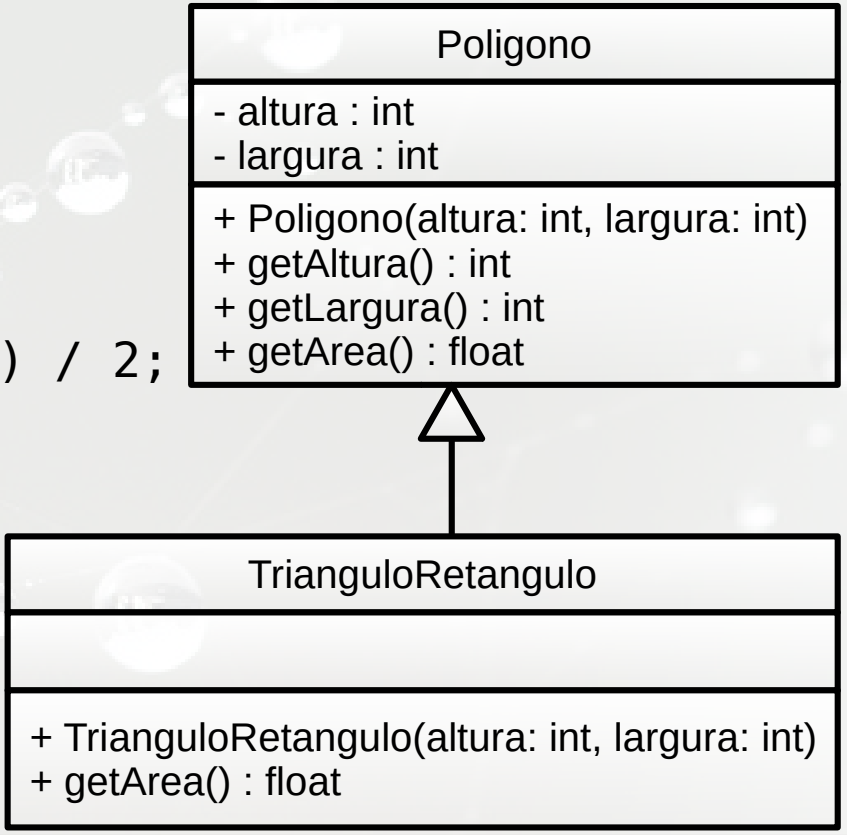
Polígono Java

```
public class Poligono {
    private int altura;
    private int largura;


    public Poligono(int altura, int largura) {
        this.altura = altura;
        this.largura = largura;
    }

    public int getAltura() {
        return altura;
    }

    public int getLargura() {
        return largura;
    }

    public float getArea() {
        return 0;
    }
}
```



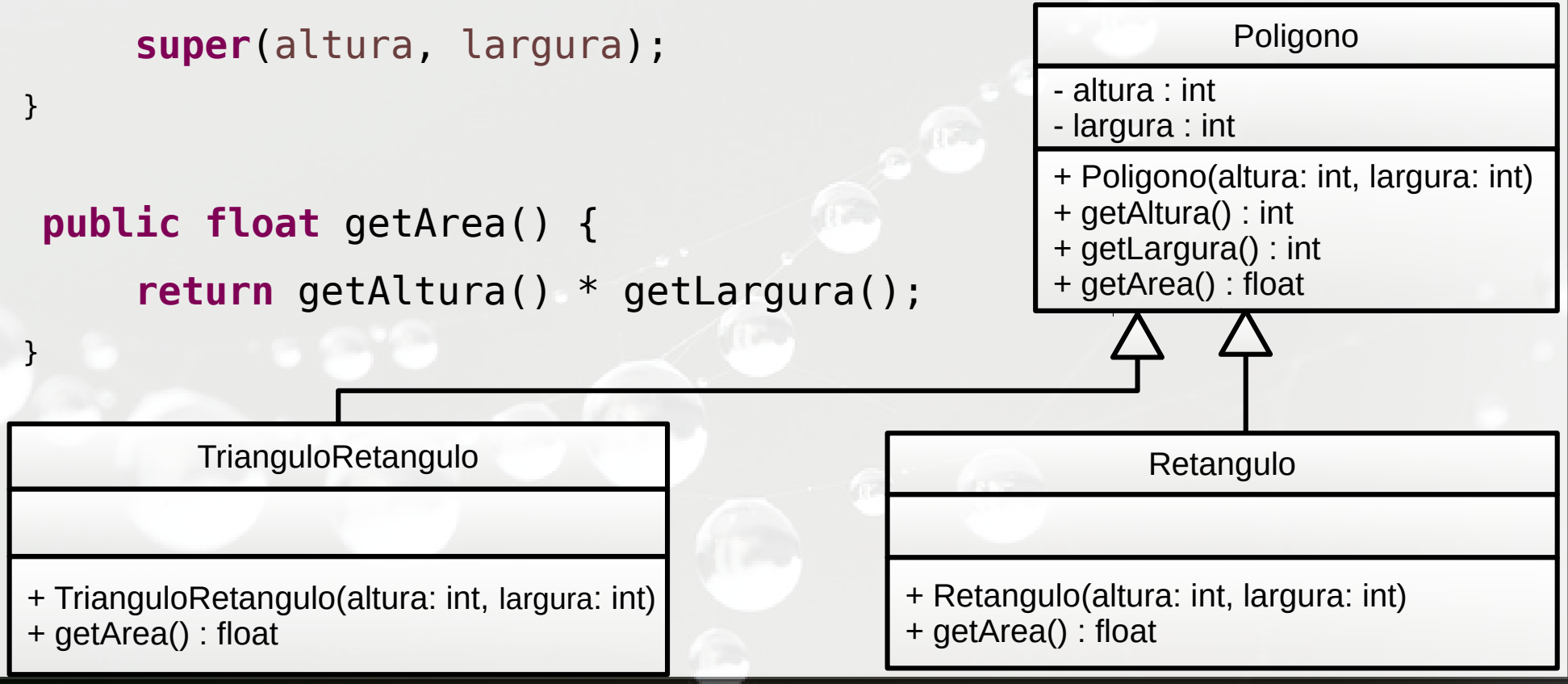
Triângulo Retângulo Java

```
public class TrianguloRetangulo extends Poligono {
    public TrianguloRetangulo(int altura, int largura) {
        super(altura, largura);
    }

    public float getArea() {
        return getAltura() * getLargura() / 2;
    }
}
```



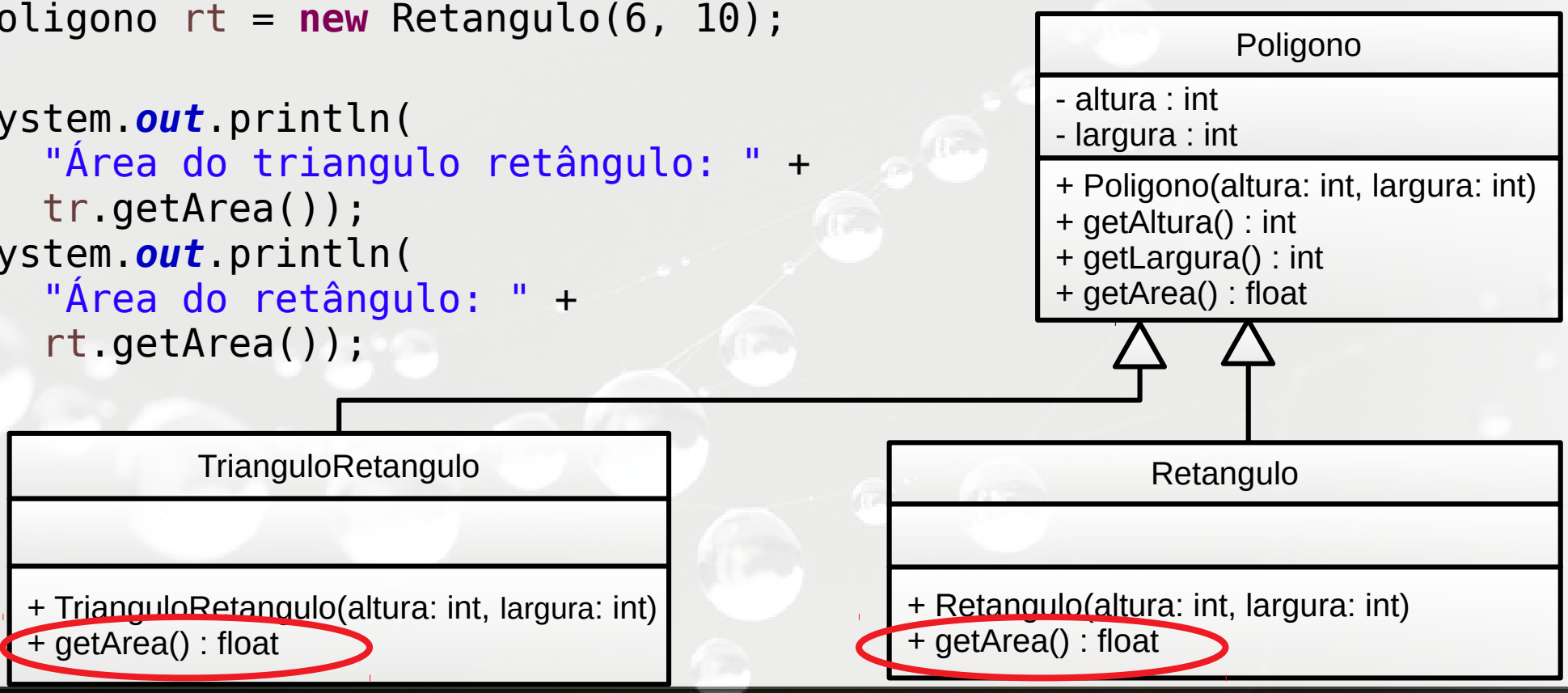
Retângulo Java

```
public class Retangulo extends Poligono {
    public Retangulo(int altura, int largura) {
        super(altura, largura);
    }

    public float getArea() {
        return getAltura() * getLargura();
    }
}
```



Polígono Java
Amarração Dinâmica

```
Poligono tr = new TrianguloRetangulo(6, 10);
Poligono rt = new Retangulo(6, 10);

System.out.println(
   "Área do triangulo retângulo: " +
   tr.getArea());
System.out.println(
   "Área do retângulo: " +
   rt.getArea());
```



André Santanchè
http://www.ic.unicamp.br/~santanche

Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.
- Mais detalhes sobre a referida licença Creative Commons veja no link: http://creativecommons.org/licenses/by-nc-sa/3.0/
- Agradecimento a James Ratcliffe [http://www.flickr.com/photos/jamie/1762955591/] por sua fotografia “A spider web after a misty morning” usada na capa e nos fundos, disponível em [http://www.flickr.com/photos/jamie/1762955591/]

  vide licença específica da fotografia.