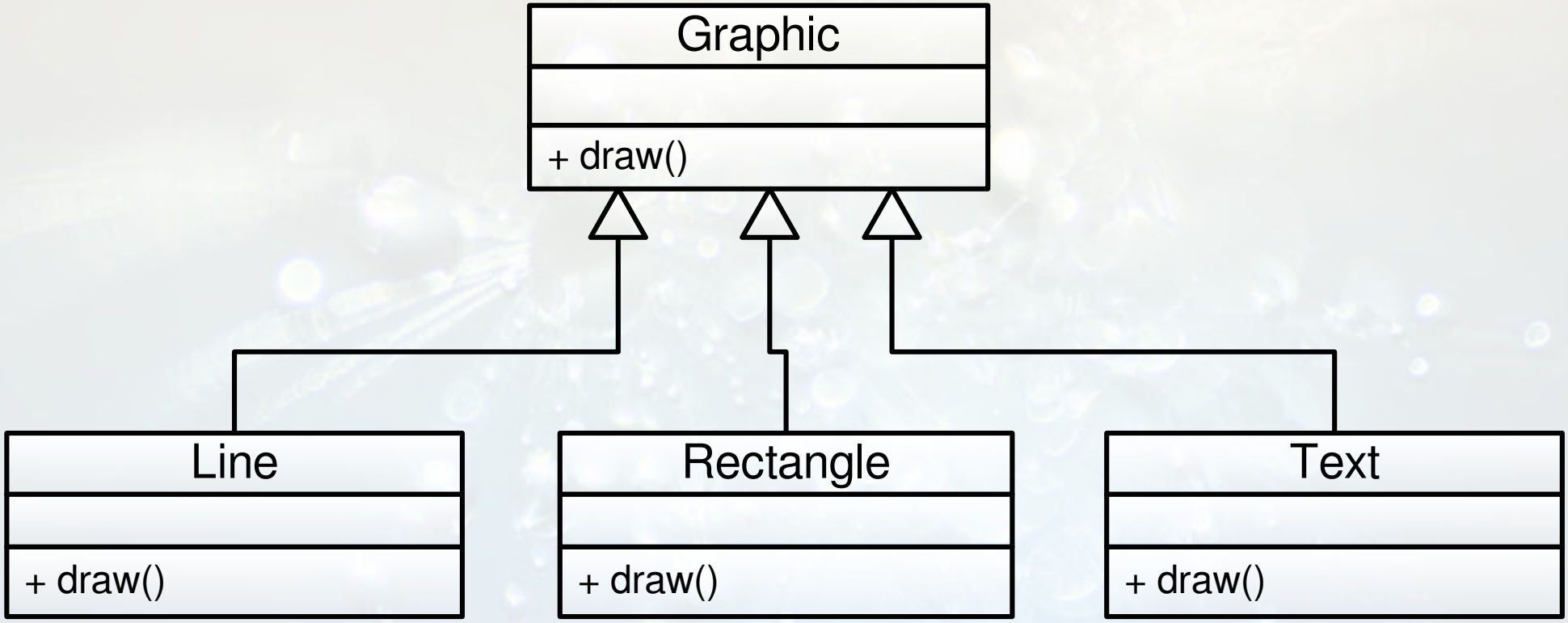
Programação Orientada a Objetos

Classe Abstrata

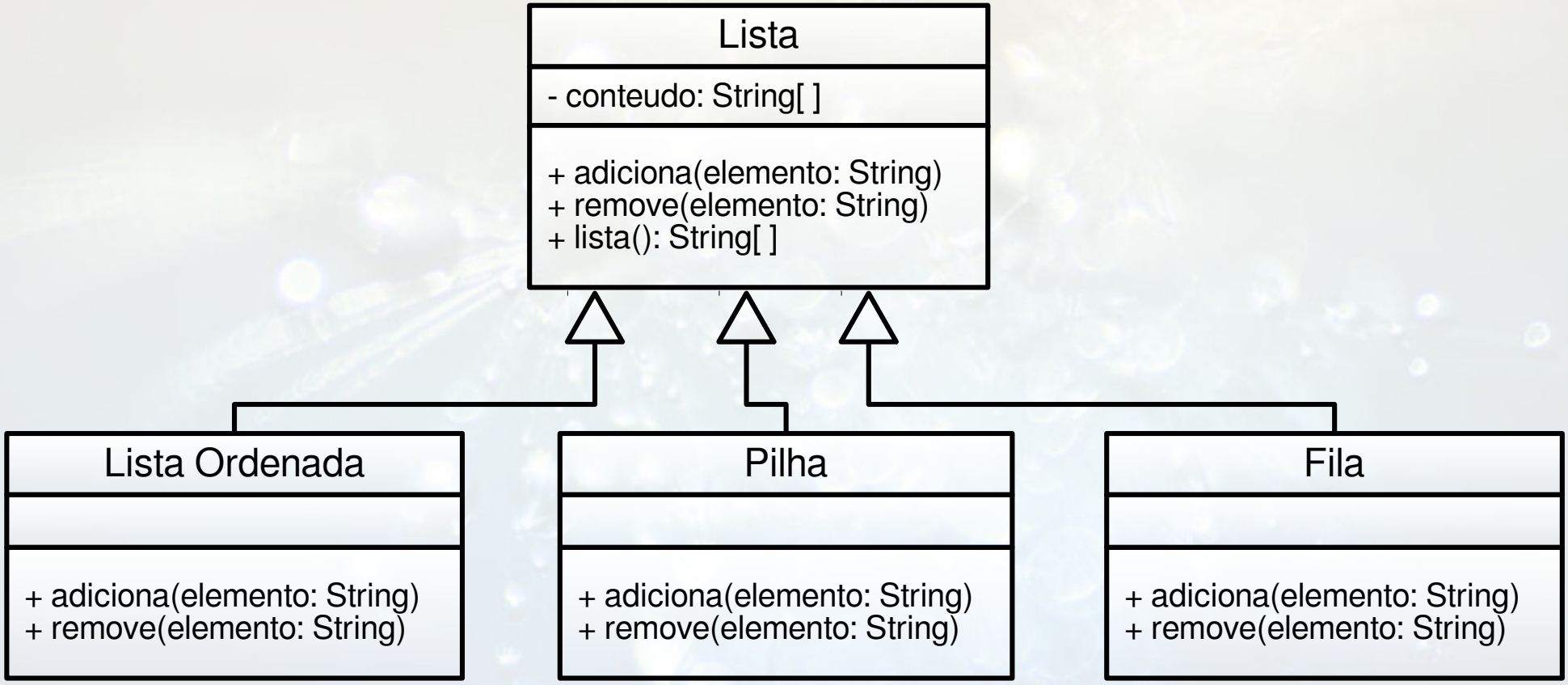
André Santanchè
Laboratory of Information Systems – LIS
Instituto de Computação – UNICAMP
Maio 2020

Picture by Neal Fowler [https://flic.kr/p/oCNoe6]

Generalizando uma Família Gráfica



Generalizando uma Família de Listas



Classe que Generaliza Família

- Reúne métodos que podem ser reusados por herdeiros
- Define métodos comuns a todos
  - a ser sobrescritos por herdeiros
- Superclasse
  - usualmente é uma definição genérica
    - não deveria ser instanciada
  - Não deveria precisar implementar métodos sobrescritos

Retomando o Polígono

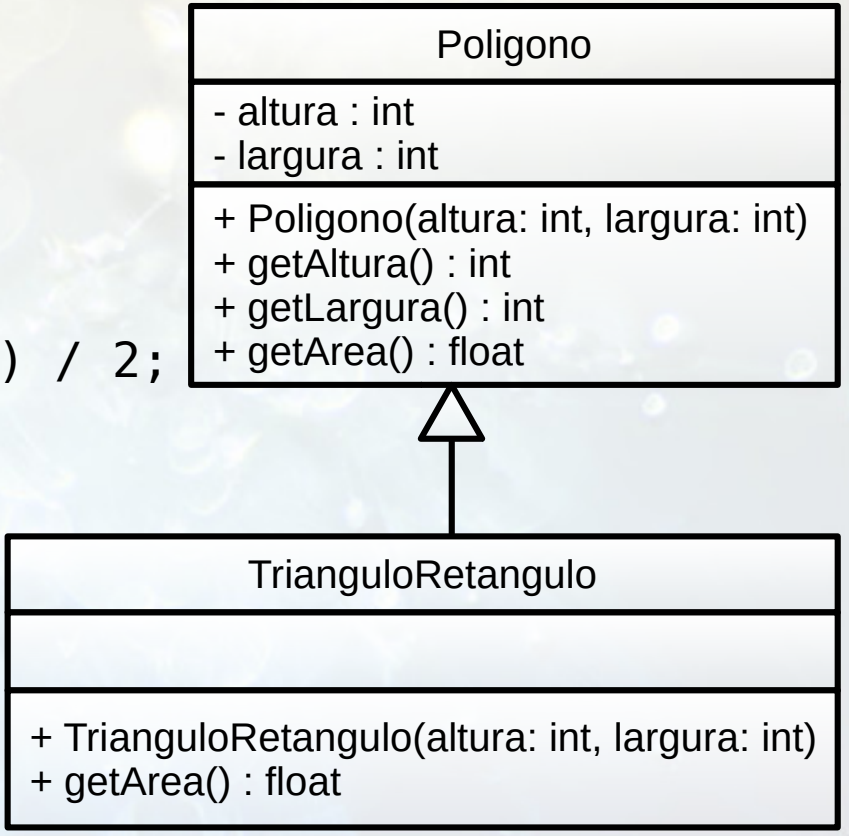
Polígono Java

```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;
    }
}
```

Poligono
- altura : int - largura : int
+ Poligono(altura: int, largura: int) + getAltura() : int + getLargura() : int + getArea() : float

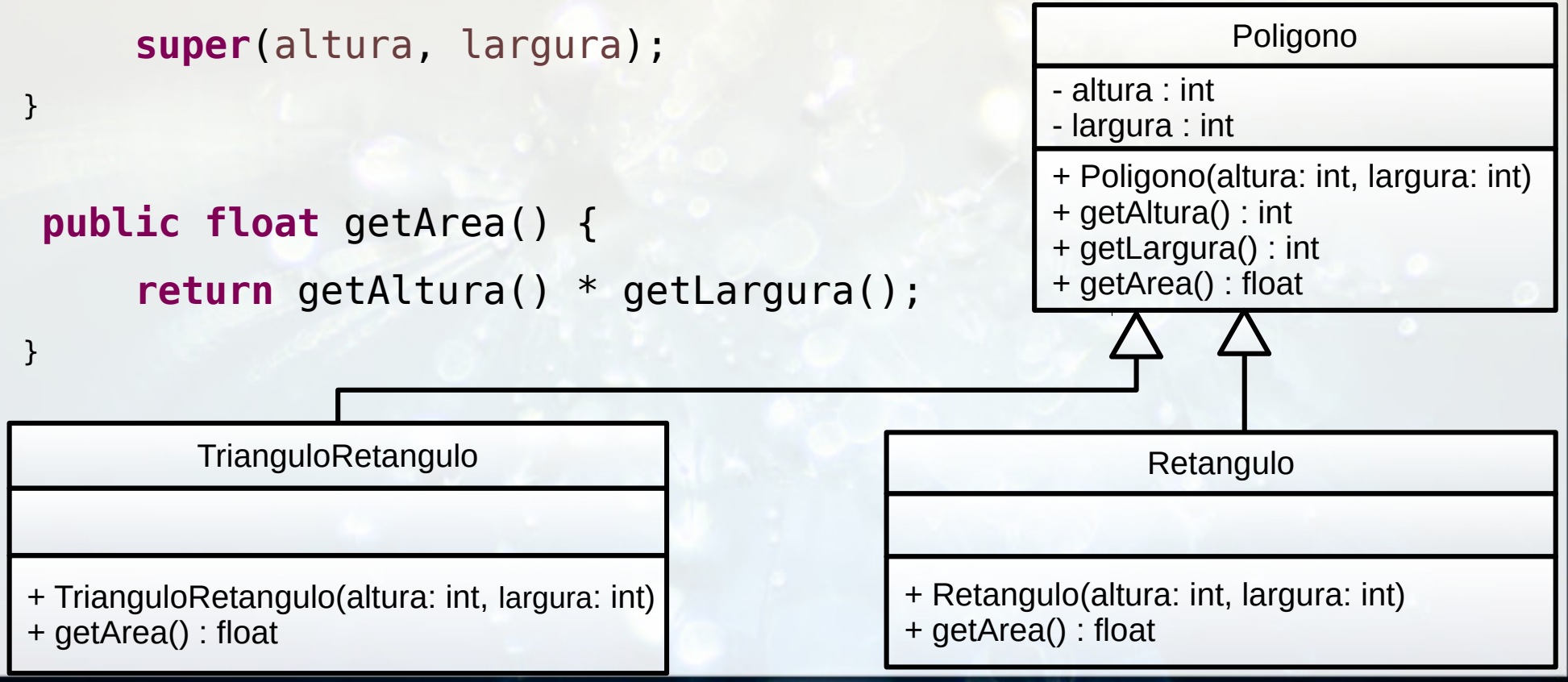
Triângulo Retângulo Java

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

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



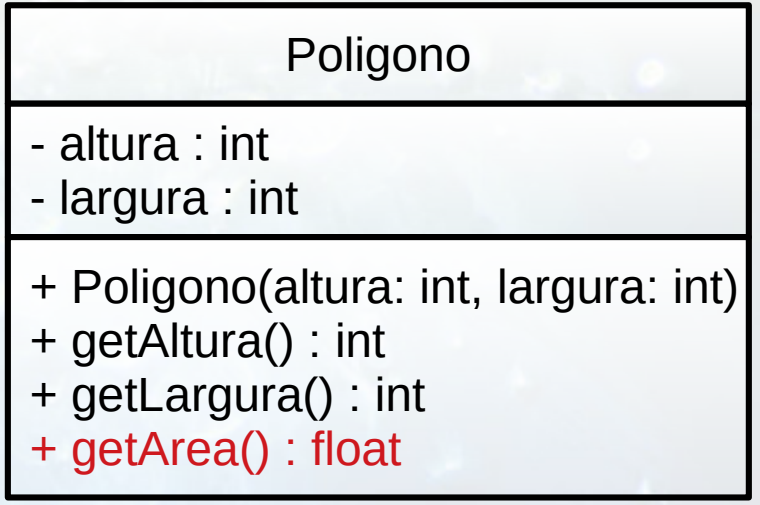
Retângulo Java

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

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



Superclasse Polígono

- Método getArea() é um “stub”
  - não deveria ser implementado nem chamado
- Trata-se de uma classe generalizadora
  - não deveria ser instanciada



Classe Abstrata

Classe Abstrata

- Não pode ser instanciada
- Pode declarar Métodos Abstratos
  - métodos apenas com a assinatura
  - mas sem implementação

Polígono Abstrato

```
public abstract 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 abstract float getArea();
}
```

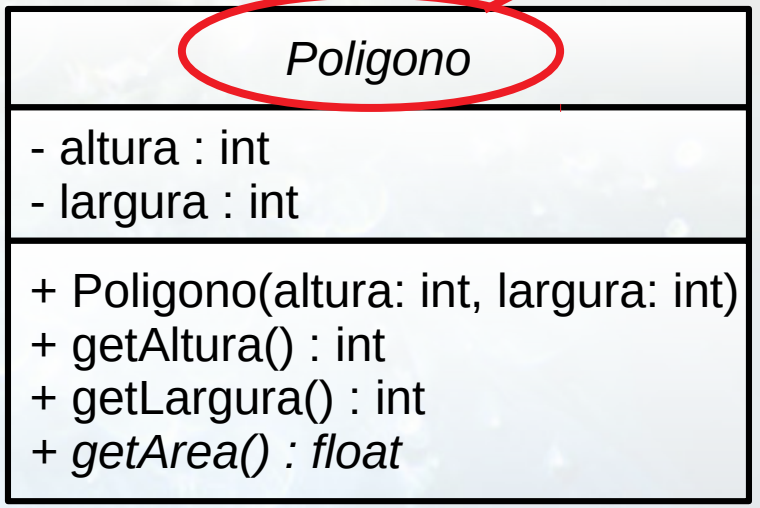
Poligono
- altura : int - largura : int
+ Poligono(altura: int, largura: int) + getAltura() : int + getLargura() : int + *getArea() : float*

Polígono Abstrato

```
public abstract 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 abstract float getArea();
}
```

classe abstrata

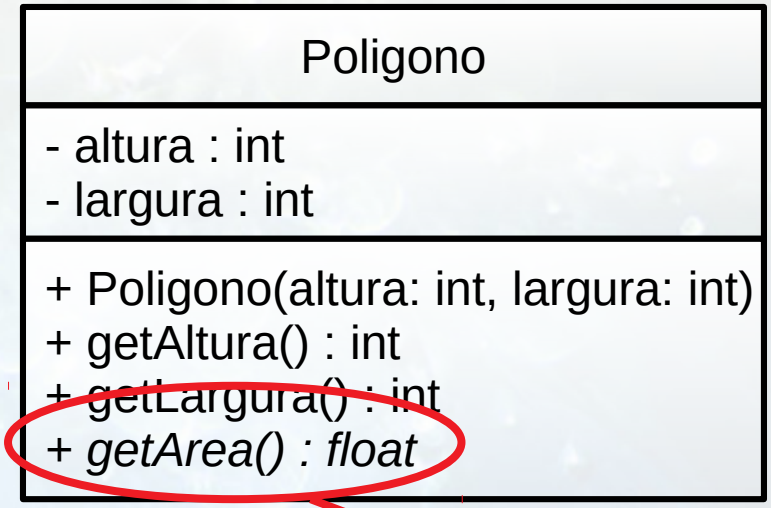
itálico



Polígono Abstrato

```
public abstract 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 abstract float getArea();
}
```

método abstrato



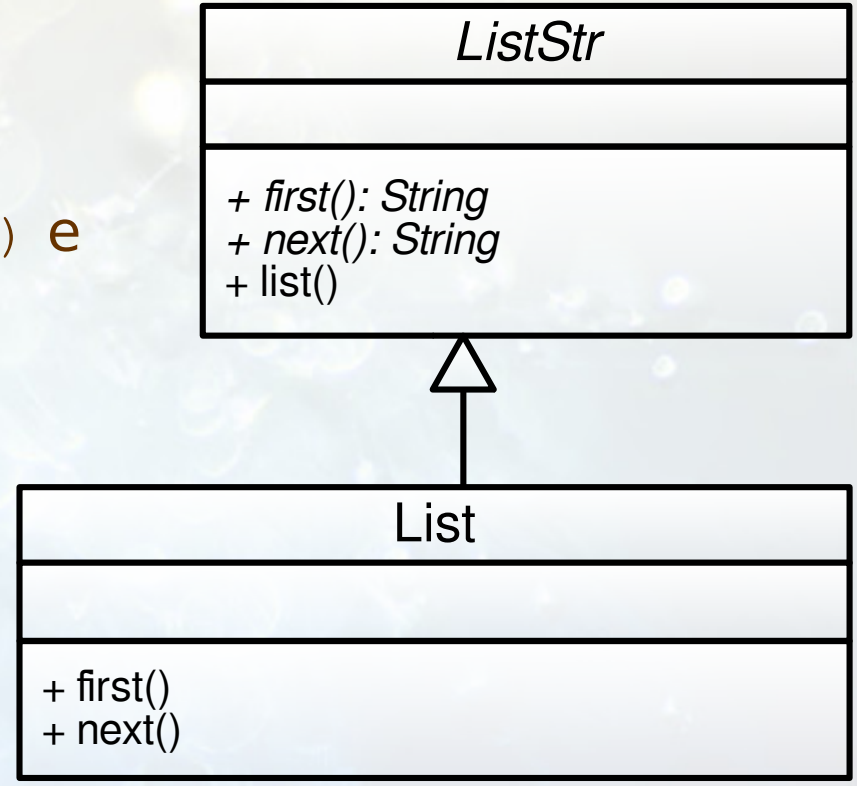
itálico

Classes Herdeiras de Abstratas

- Métodos abstratos são obrigatoriamente implementados pelos herdeiros
- Classes herdeiras de abstratas também podem ser abstratas
  - podem repassar a responsabilidade de implementar métodos abstratos para a geração seguinte

Tarefa

- Implemente uma classe herdeira da classe abstrata ListStr
  - armazene uma lista de Strings
  - implemente os métodos `first()` e `next()`



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 Picture by Neal Fowler [https://www.flickr.com/photos/31878512@N06/] por sua fotografia “Explosion” usada na capa e nos fundos, disponível em [https://flic.kr/p/oCNoe6]. Vide licença específica da fotografia.