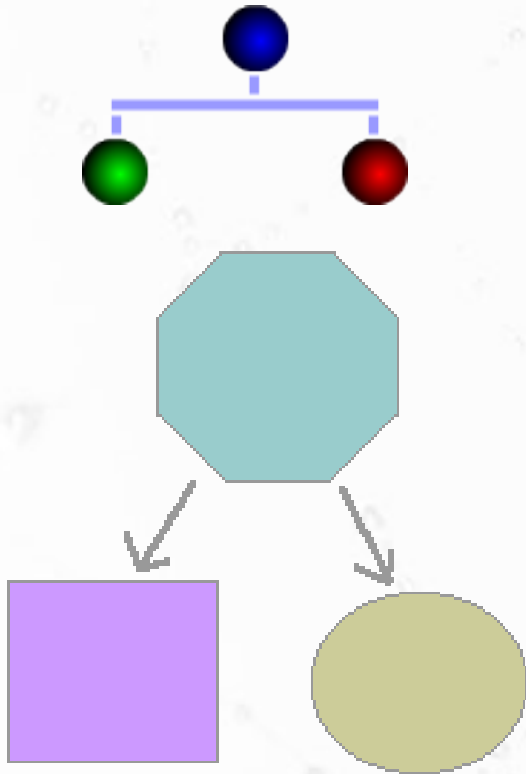


Programação Orientada a Objetos

Herança e Polimorfismo

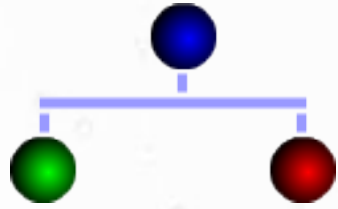
André Santanchè e Oscar Rojas
Institute of Computing - UNICAMP
Março 2019

Herança



- Capacidade das classes expandirem-se a partir das classes existentes.

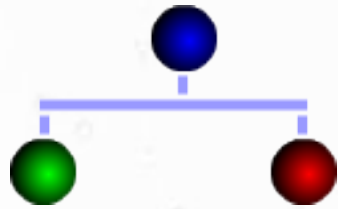
- Classe herdeira (subclasse)
 - possui os mesmos atributos da superclasse
 - herda acesso aos métodos desta superclasse
 - pode acrescentar novos atributos e métodos (extensão)



Herança em Java

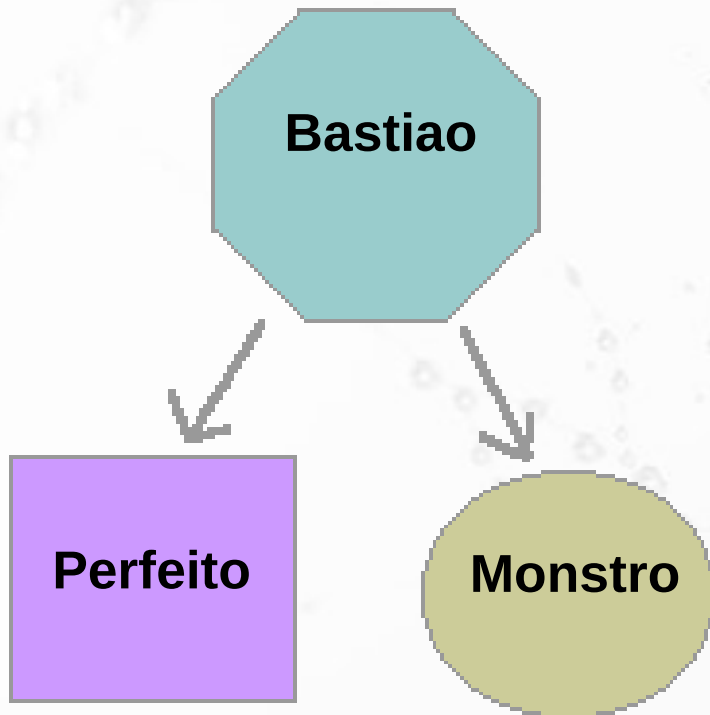
- Para se estabelecer que uma classe é herdeira de outra, após o nome da classe coloca-se a cláusula `extends` e o nome da superclasse.
- No exemplo abaixo, `Perfeito` é herdeira de `Bastiao`:

```
class Perfeito extends Bastiao
```

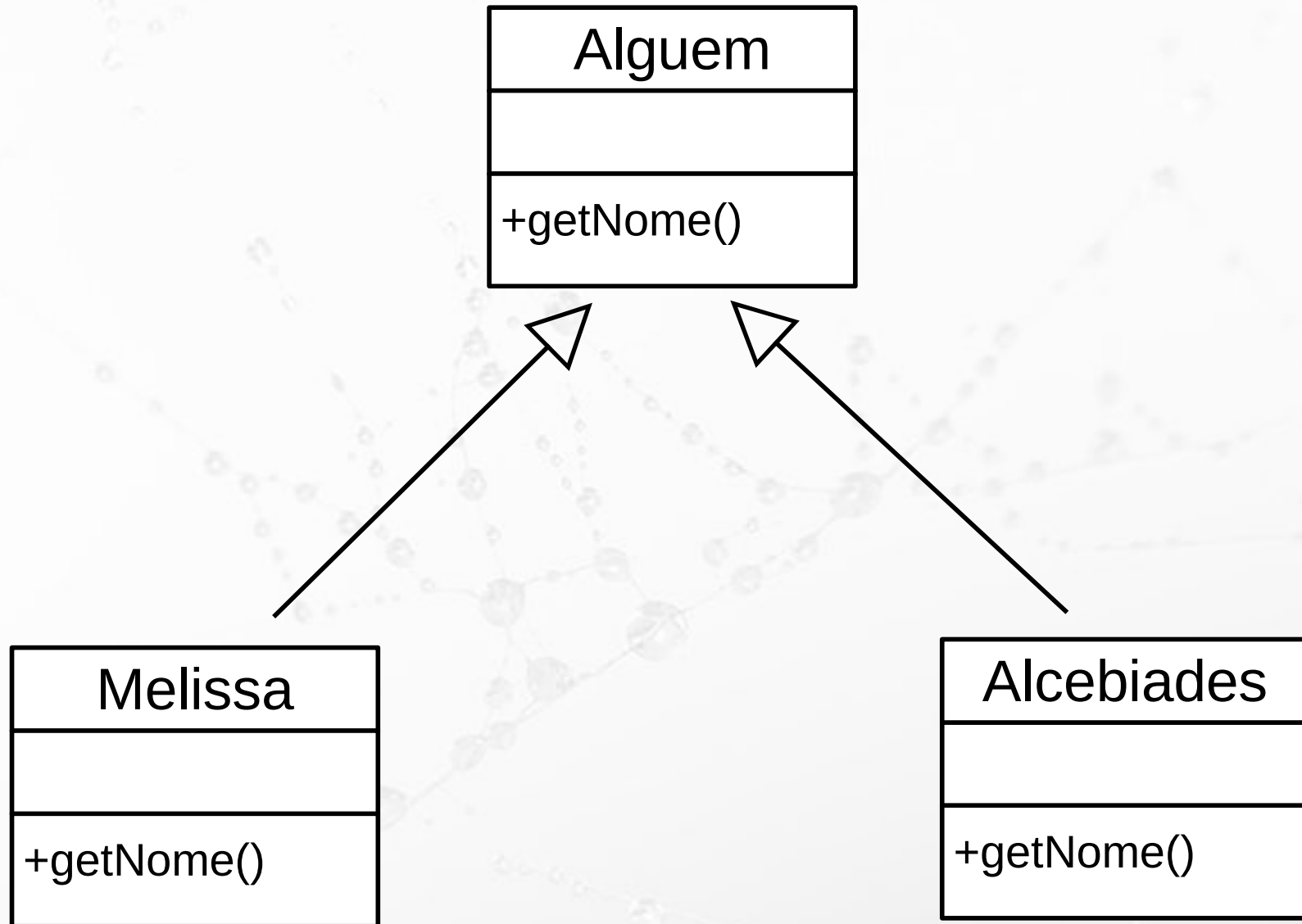


Herança em Java

- No exemplo iremos criar as classes `Perfeito` e `Monstro` que são herdeiras de `Bastiao`.
- Elas possuem todos os recursos de `Bastiao` mais as capacidades específicas.

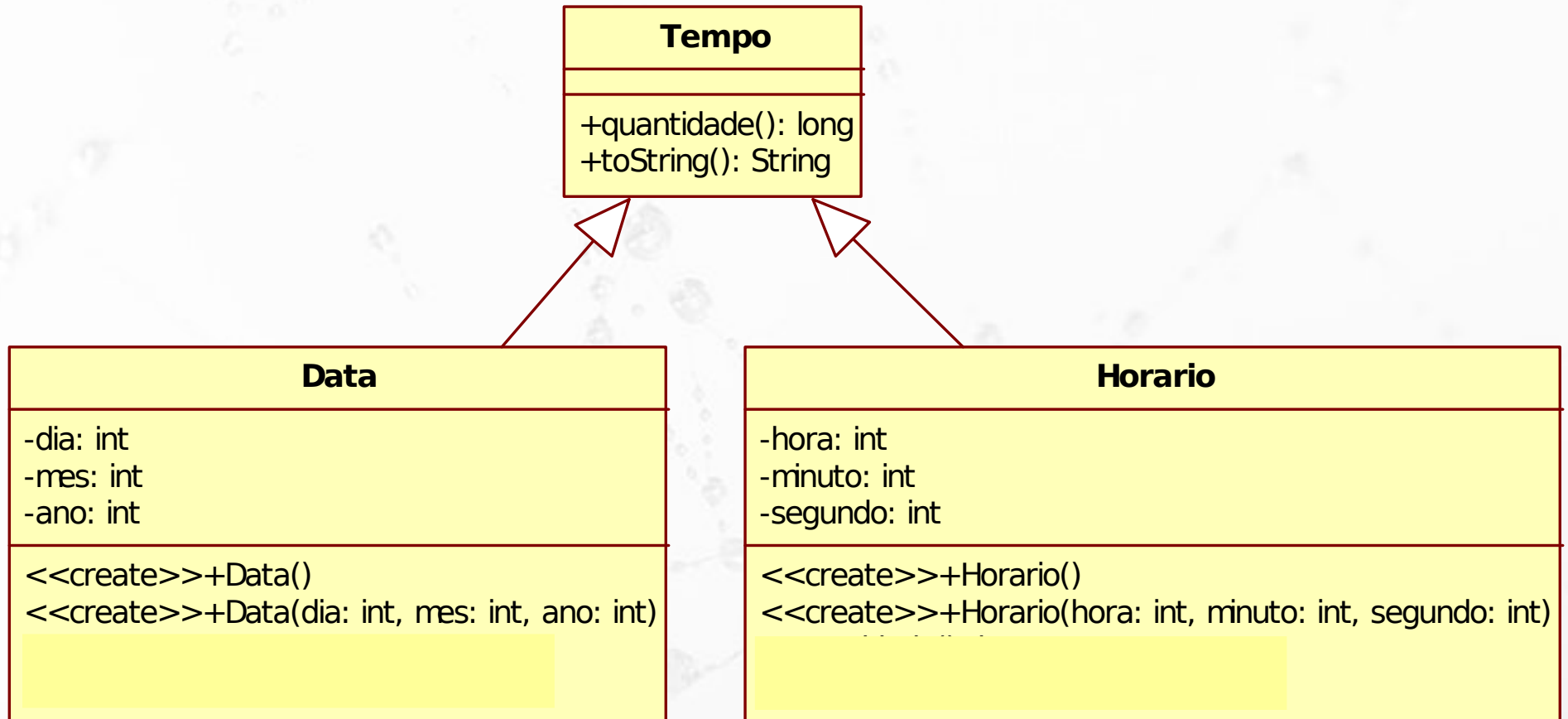


Herança - Alguem



Herança

Exemplo do Tempo



Referência à Superclasse

`super`

- Referência explícita à superclasse
- Uso comum - quando um método da classe herdeira sobrescreve um método da superclasse.
- A cláusula `super` representa a superclasse:

`super.metodo(...)`

Construtor da Superclasse

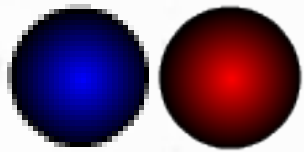
`super`

- Chamando o construtor da superclasse:
 - Apenas a cláusula `super` na forma de método, com parâmetros (se houver):

```
super (...)
```

- Só pode ser chamado do construtor da classe
- Se for chamado, tem que ser a primeira instrução do construtor

Polimorfismo



Princípios do Paradigma Polimorfismo

- "Que se apresenta sob numerosas formas"
- "Capacidade de uma referência de classe se associar a instâncias de diferentes classes em tempo de execução".
- "Habilidade das mais importantes dos sistemas orientados a objetos, e que consiste em as operações automaticamente se adequarem aos objetos aos quais estão sendo aplicadas."
[Meyer]

Sobrecarga de Métodos

- Sobrecarga de método: técnica que envolve criar vários métodos com o mesmo nome e implementações diferentes.
- Tipos:
 - sobrecarga na mesma classe
 - assinaturas têm que ser diferentes
 - identificados pela assinatura
 - sobrecarga em classes herdeiras
 - assinaturas podem ser iguais ou diferentes
 - tratado na aula de herança

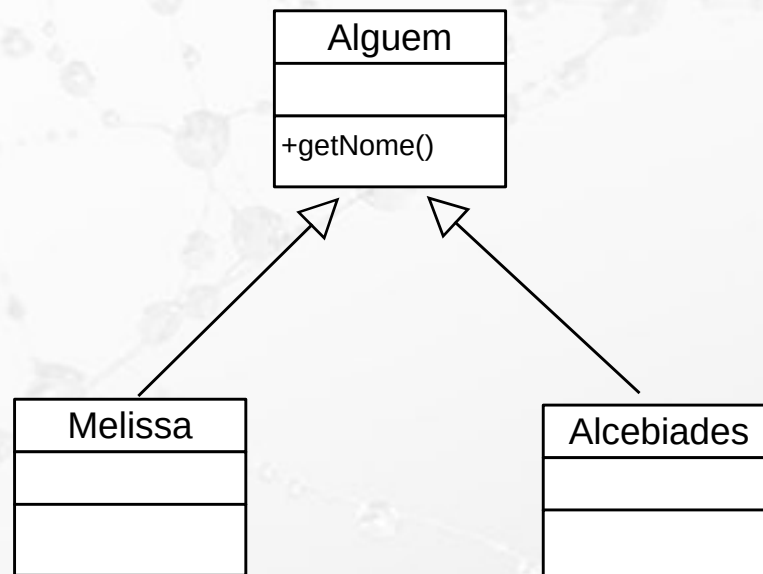
Sobrecarga em Classes Herdeiras

- Métodos podem ter mesma assinatura
- Neste caso, a decisão de qual método chamar (superclasse ou subclasse) depende:
 - do tipo do ponteiro
 - do tipo de amarração

Princípios do Polimorfismo com Herança

- Uma variável declarada em uma classe pode ser instanciada em qualquer subclasse

```
Alguem umaPessoa = new Melissa();
```



Princípios do Polimorfismo com Herança

- A decisão de quem chamar depende do tipo de amarração:
 - **Estática**
 - **Dinâmica**

Polimorfismo

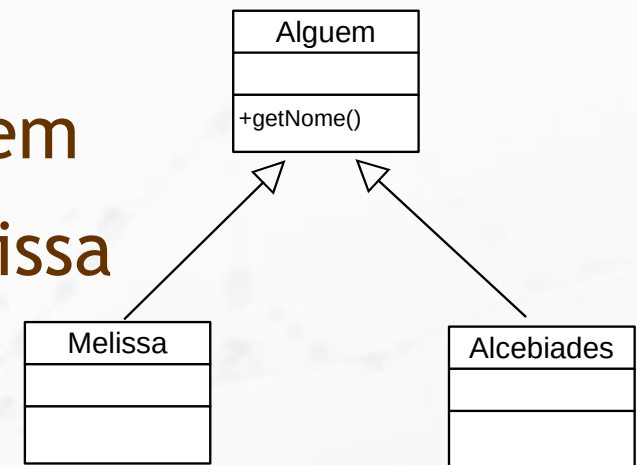
Amarração Estática x Dinâmica

- **Amarração:** ligação da chamada de um método ao método.
- **Amarração estática** (*static binding*): define permanentemente o endereço do método acionado durante a compilação.
- **Amarração dinâmica ou tardia** (*dynamic or late binding*): determina o endereço do método acionado no momento da execução.

Princípios do Polimorfismo com Herança

- A decisão de quem chamar depende do tipo de amarração:

- **Estática** - retorna nome de Alguem
- **Dinâmica** - retorna nome de Melissa

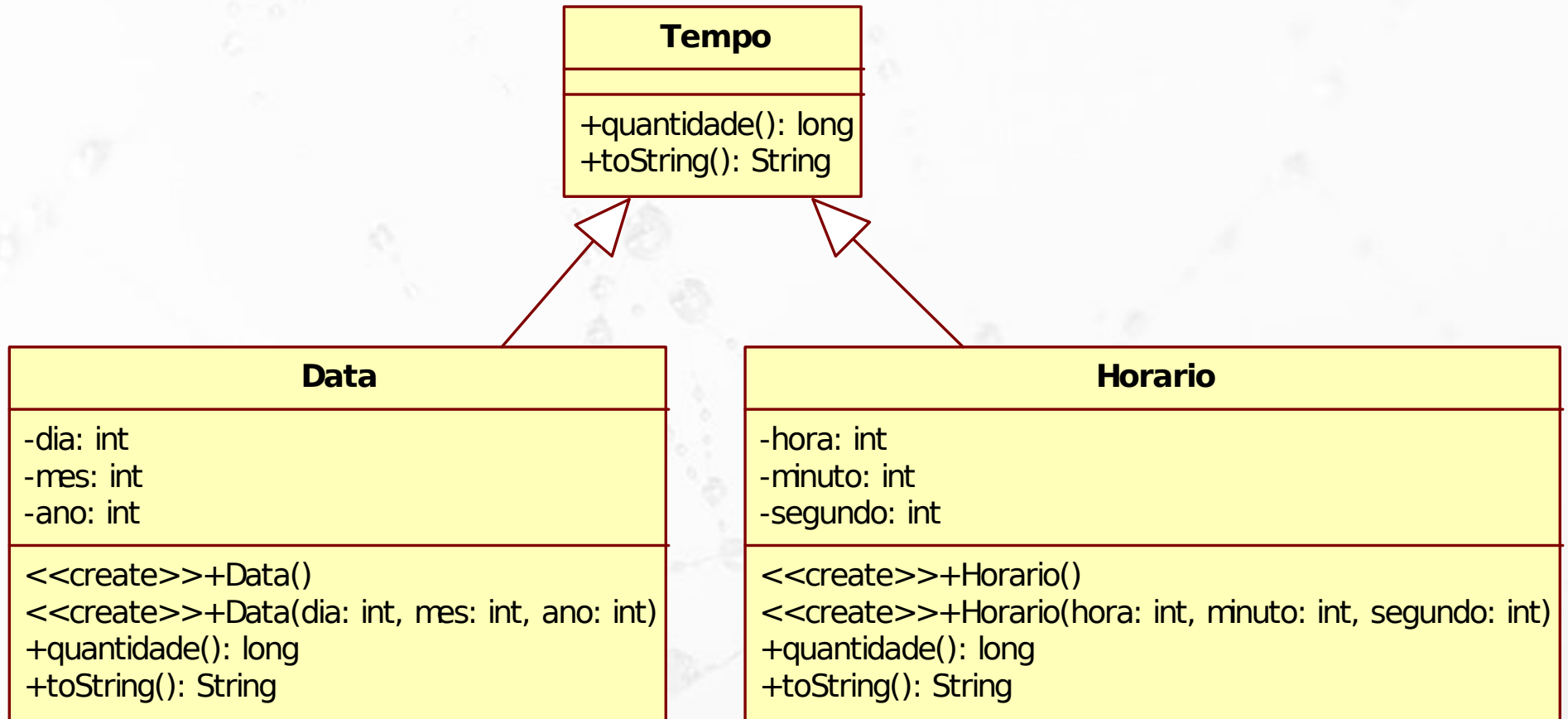


```
Alguem umaPessoa = new Melissa();
```

```
String x = umaPessoa.getNome();
```


Herança

Exemplo do Tempo



The background of the slide is a grayscale image of a spider web. The web is composed of numerous fine, intersecting lines that form a complex, radial pattern. Scattered across the web are several water droplets of varying sizes, some of which are in sharp focus, reflecting light and creating bright highlights. The overall effect is a delicate and intricate texture.

André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: <http://creativecommons.org/licenses/by-nc-sa/3.0/>