

A top-down view of a desk. On the left, a small potted plant with green leaves is partially visible. In the center, a black laptop is open. To the right of the laptop, a pair of black-rimmed glasses lies on the desk. Below the glasses, a hand is holding a black pencil, poised to write on a white sheet of paper. The paper has a small, circular logo on it. The background is a dark, textured surface.

Evitando

Duplicação de

Código com

Orientação a

Objetos e Generics

Avisos!

- © Os exemplos de negócio utilizados são extremamente simplificados!
- © Não existe bala de prata no desenvolvimento de software!
- © Código e Slides em:
github.com/bennersaude/UmDesignPatternPorSemana

Tópicos

© Orientação a Objetos

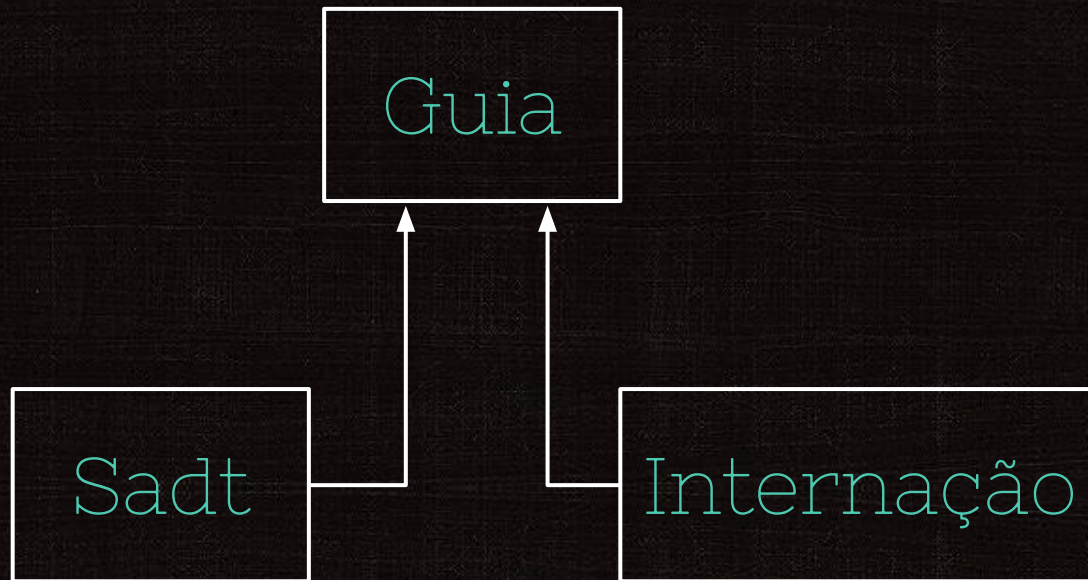
- Herança
- Interfaces
- Polimorfismo de Parâmetros

© Generics

- Classes Genéricas
- Parâmetros Genéricos
- Restrições

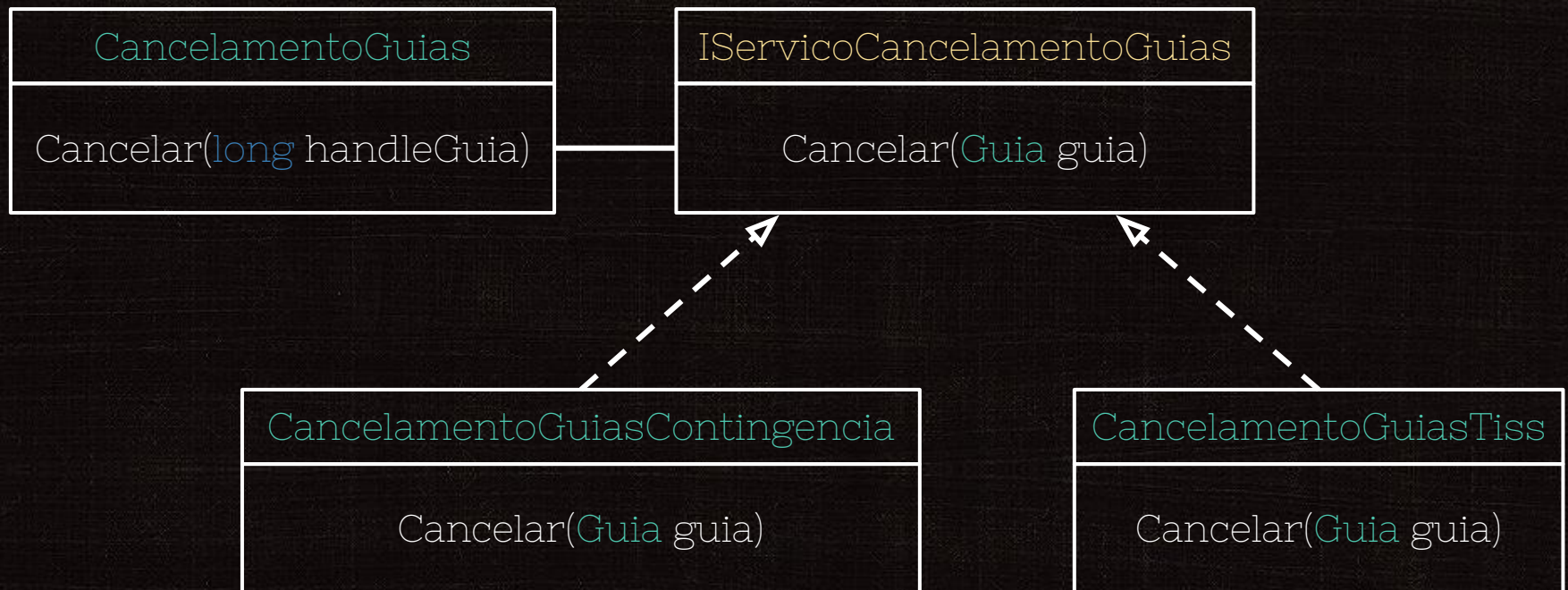
Herança

Permite criar classes que são especializações de uma outra classe já existente.



Interfaces

Permite que objetos se comuniquem uns com os outros, sem se conhecer diretamente.

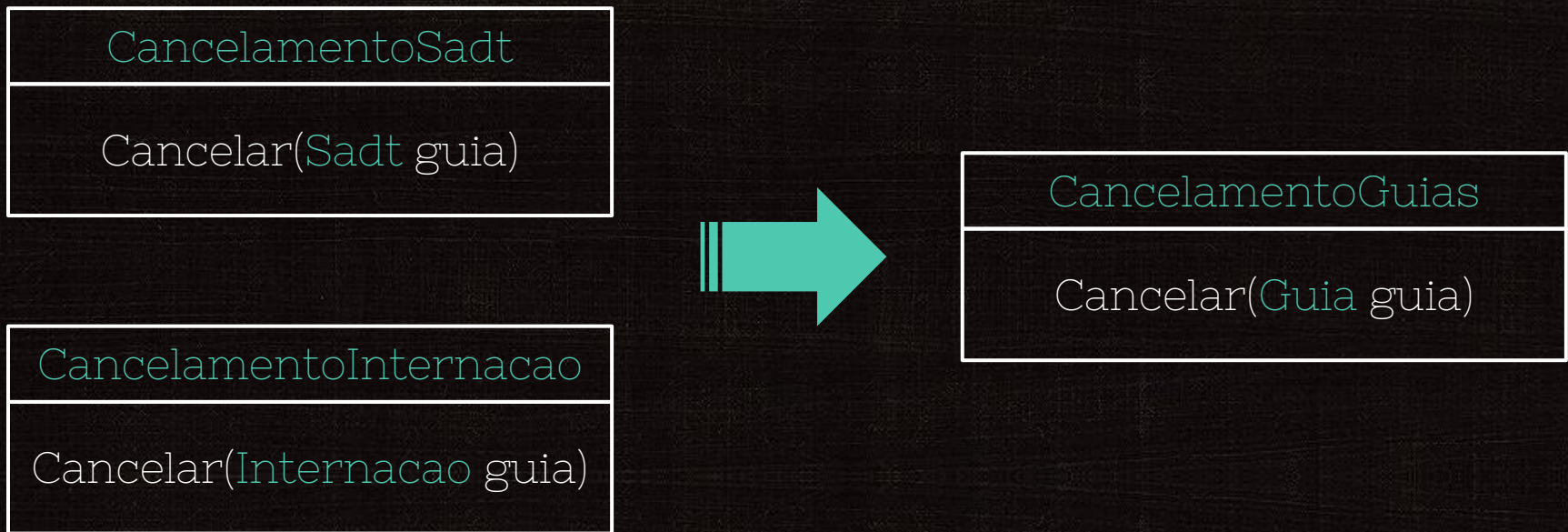




Show me the code!

Polimorfismo de Parâmetros

Permite que um método aceite parâmetros de tipos diferentes.





Show me the code!



Generics?

Ninguém nasce
sabendo!

stackoverflow.com 

Generics

Tipos genéricos trazem o conceito de parâmetros de tipo, que possibilitam a criação de classes e métodos que adiam a especificação de um ou mais tipos até que a classe ou método seja declarado e instanciado pelo código que o utiliza.

Os tipos genéricos estão presentes desde a versão 2.0 do c#

Generics

- © Use tipos genéricos para maximizar a reutilização de código, o desempenho e a segurança de tipo.
- © Você pode criar suas próprias interfaces genéricas, classes, métodos, eventos e delegates.
- © Classes genéricas podem ser restringidas para habilitar o acesso aos métodos em tipos de dados específicos.
- © Informações sobre os tipos que são usados em um tipo de dados genérico podem ser obtidas no tempo de execução por meio de reflexão.



Show me the code!