

Modelação e Design

12: Diagrama de Classes: Design

Leonor Melo
leonor@isec.pt

1

Diagrama de classes

- Operações num diagrama de classes
- Classes com membros estático

12 Diagramas de classe

2

Comportamen to da classe: operações

- A secção das operações no diagrama de classes descreve
 - o que a classe consegue fazer
 - mas não descreve como o irá fazer
- Apresenta apenas a "assinatura":
 - visibilidade,
 - nome da operação,
 - tipos de dados que recebe,
 - tipos de dados que devolve

12 Diagramas de classe

3

Comportamen to da classe: operações

- Operação:
 - "contrato" que indica que a classe irá conter um comportamento que fará o que a assinatura da operação sugere
- Conjunto das operações
 - serviços que a classe disponibiliza devido à sua "natureza"
 - mensagens que podem ser enviadas a essa classe e/ou suas instancias

12 Diagramas de classe

4

Comportamento da classe: operações UML - 1

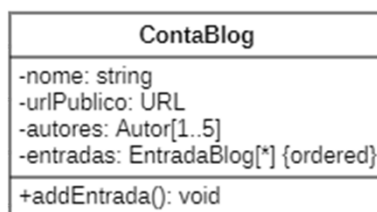
- Formato completo:
 - [visibilidade] **nome** ([parâmetro [, parâmetro]]) [: tipo] [multiplicidade] [{ propriedade [, propriedade]]}
- Também pode ser usando a sintaxe de uma linguagem de programação
 - +getJogador(nome : String) : Jogador {exception IOException}
 - public Jogador getJogador(String nome) throws IOException

12 Diagramas de classe

5

Comportamento da classe: operações UML - 2

- Assinatura da operação deve ter pelo menos:
 - visibilidade
 - nome
 - par de parêntesis com os parâmetros lá dentro
 - tipo de dados devolvido

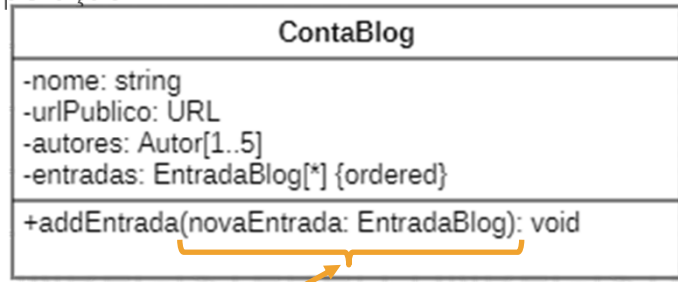


12 Diagramas de classe

6

Comportamen
to da classe:
operações:
parâmetros -1

- Parâmetros
 - especificar a informação passada à operação



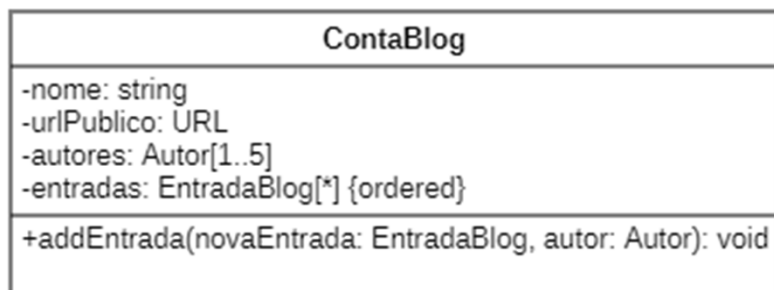
- addEntrada mais realista

12 Diagramas de classe

7

Comportamen
to da classe:
operações:
parâmetros -2

- Para cada parâmetro é preciso indicar
 - pelo menos o nome
- Podemos passar mais que um parâmetro
 - separandos por vírgulas



12 Diagramas de classe

8

Comportamen
to da classe:
operações:
parâmetros -3

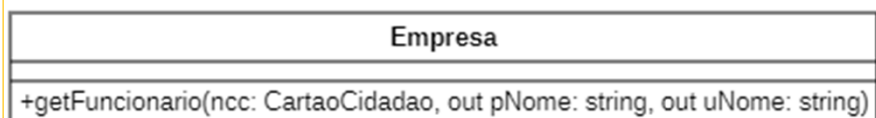
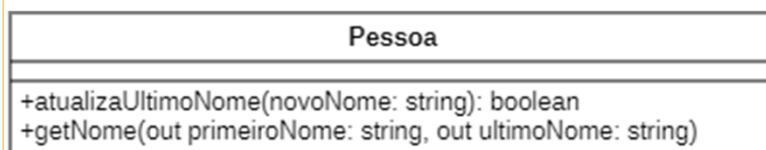
- Sintaxe completa dos parâmetros:
- [in / out / inout] nome [: tipo]
[multiplicidade] [= default] [{propriedade
[, propriedade]}]

12 Diagramas de classe

9

Comportamen
to da classe:
operações:
parâmetros -4

- Exemplos:

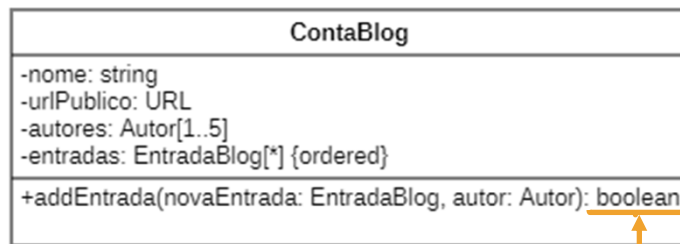


12 Diagramas de classe

10

Comportamen
to da classe:
operações:
tipo do retorno

- Tipo de retorno
 - tipo do objeto que será devolvido pela operação



- *verdadeiro* se conseguiu adicionar, *falso* se não conseguiu

12 Diagramas de classe

11

Comportamen
to da classe:
construtores

- Construtores
 - funções especiais
 - criam o objeto e inicializam corretamente seus atributos
 - podem receber parâmetros
 - Devolvem o objeto construído
 - Denotam-se como operações associadas a objetos, mas na realidade são executadas pelo compilador
 - Só se representam em diagramas de classe de design de baixo nível

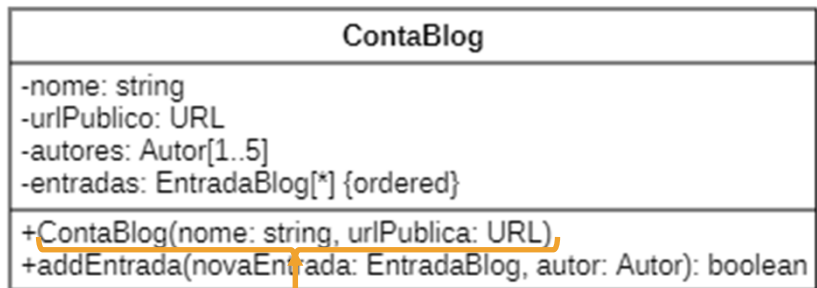
12 Diagramas de classe

12

Comportamento da classe: construtores

- Construtores

- nome operação = nome da classe
- não precisam de indicar tipo de retorno



- devolve uma ContaBlog

12 Diagramas de classe

13

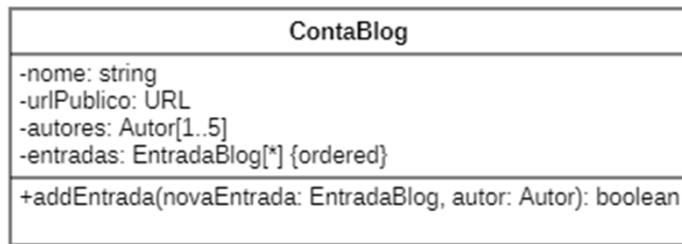
Partes estáticas e não estáticas da classe

- Atributos e operações podem ser estáticos
- Estático
 - associado com a classe
- não estático
 - associado com a instancia

12 Diagramas de classe

14

Partes não - estáticas da classe - 1

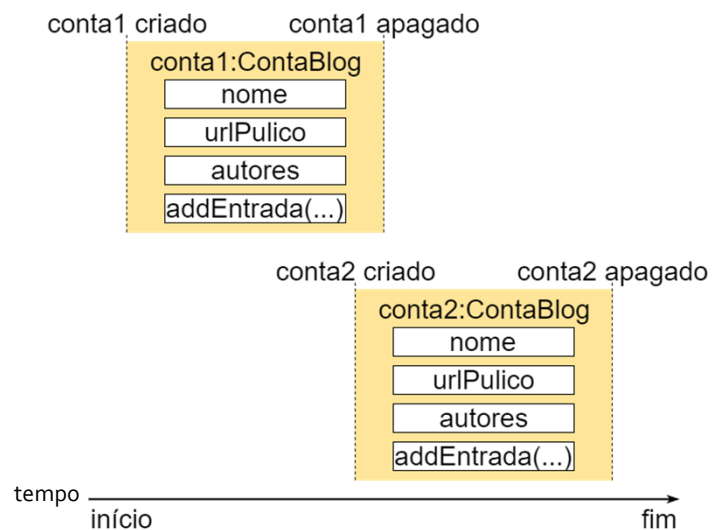


- Cada objeto da classe ContaBlog tem
 - 4 atributos
 - 1 operação
- Partes não estáticas da classe têm a duração das instancias da qual fazem parte

12 Diagramas de classe

15

Partes não - estáticas da classe - 2



12 Diagramas de classe

16

Partes estáticas da classe

- As classes "existem" assim que o programa se inicia e duram até que o programa termine
 - não dependem da existência de nenhum objeto dessa classe
- Membros estático
 - só têm uma "cópia" (associada à classe)
 - estão disponíveis durante toda a execução

12 Diagramas de classe

17

Partes estáticas da classe - 1

- Problema:
 - queremos criar um atributo privado, `contaContas`, que registre o número de objetos `ContaBlog` existentes em cada momento
- Como fazer?

12 Diagramas de classe

18

Partes estáticas da classe - 2

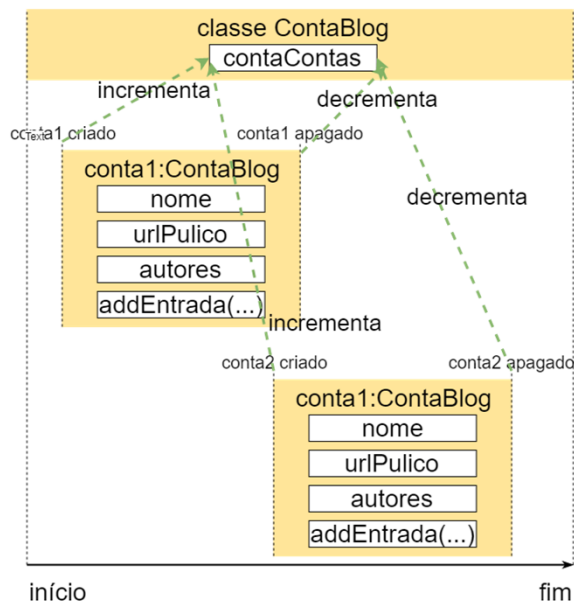
- contaContas
 - regista o número de objetos ContaBlog existentes em determinado momento
 - faz sentido que exista apenas um para toda a classe



12 Diagramas de classe

19

Partes estáticas da classe – 3



12 Diagramas de classe

20

padrão de design

- Design pattern
 - Solução para um problema que existe em determinado contexto
- Contexto:
 - situação à qual o problema se aplica
- Problema:
 - objetivo que se está a tentar alcançar (restrições encontradas nesse contexto)
- Solução:
 - design genérico que pode ser aplicado nessa situação e que resolve o problema

12 Diagramas de classe

21

padrão de design

- Uma solução frequentemente usada para um problema comum
- Problema: odor no lava-louça
- Padrão: sifão em S
- Problema: cruzamento na autoestrada
- Padrão: trevo de quatro-folhas



12 Diagramas de classe

22

padrão de design de software singleton -1

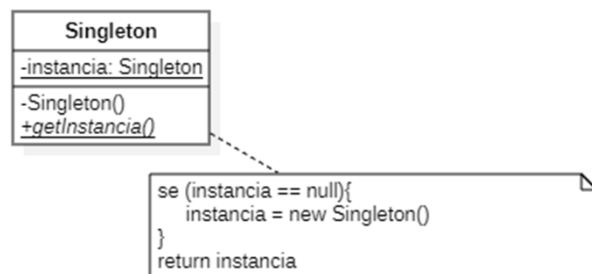
- Padrão de software singleton
 - A aplicação necessita de uma e só uma instância do objeto.
 - Essa instancia tem de ser globalmente acessivel
- Exemplo:
 - classe que regista as configurações do sistema
 - criar mais do que um objeto deste tipo durante a execução do programa pode originar problemas

12 Diagramas de classe

23

padrão de design de software singleton -2

- Padrão de software singleton
 - A aplicação necessita de uma e só uma instância do objeto.
 - Essa instancia tem de ser globalmente acessivel



12 Diagramas de classe

24

padrão de
design de
software
singleton -3

Exemplo:

classe que regista as configurações do sistema

criar mais do que um objeto deste tipo durante a execução do programa pode originar problemas

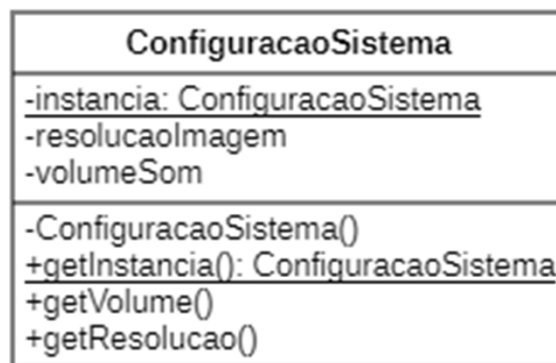
12 Diagramas de classe

25

padrão de
design de
software
singleton -4

Exemplo:

classe que regista as configurações do sistema



12 Diagramas de classe

26