

Tópicos Abordados



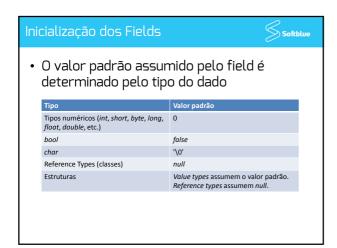
- Inicialização dos fields
- Construtores de classes e estruturas
 - Construtores com parâmetros
 - Sobrecarga de construtores
 - Construtor padrão
- O operador this
- Encadeando construtores

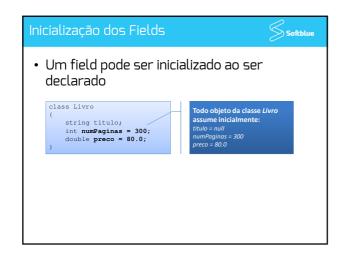
Tópicos Abordados

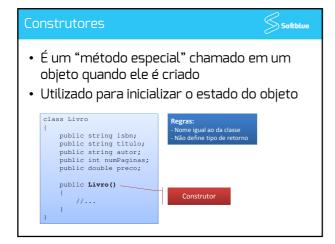


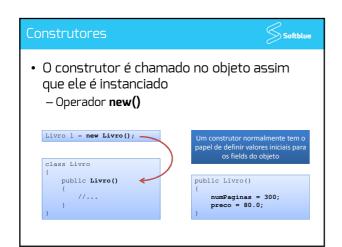
- Modificador static
 - Fields estáticos
 - Métodos estáticos
 - Construtores estáticos
 - Classes estáticas
 - O modificador *static* em estruturas
- Constantes
- · Read-only fields

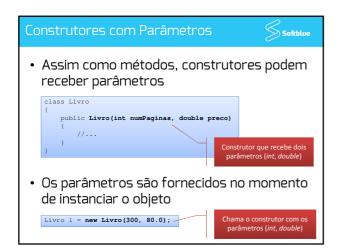


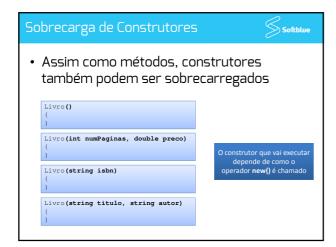


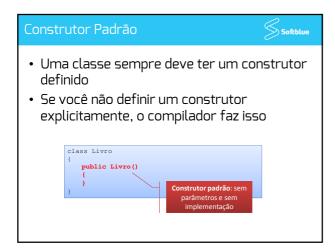


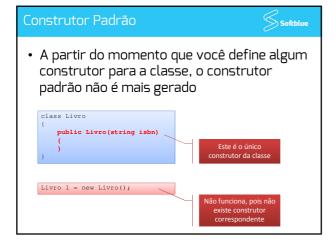




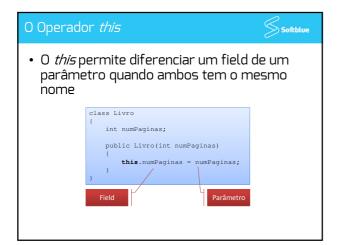


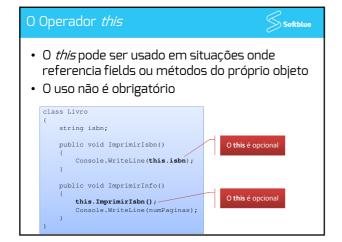






O Operador this O this é uma forma do objeto referenciar ele próprio O uso do this é comum na resolução de problemas de ambiguidade entre variáveis Class Livro (int numPaginas) public Livro(int numPaginas) numPaginas = numPaginas) ?





Em algumas situações pode ser interessante fazer com que um construtor chame outro O operador this() é utilizado nestes casos Class Livro (this Livro(string titulo) : this(titulo, parametros (string, string)) public Livro(string titulo) : this(titulo, "Desconhecido") public Livro(string titulo, String autor) (this.titulo = titulo; this.autor = autor; this.aut

Construtores em Estruturas

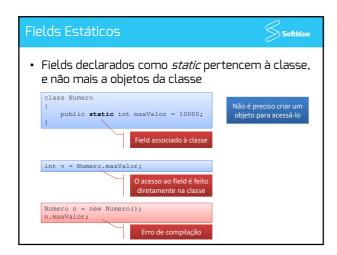


- As mesmas regras de construtores de classes funcionam para estruturas
- Algumas diferenças
 - O compilador sempre define um construtor padrão, independente de existirem outros construtores
 - Ele inicializa os fields da estrutura com o valor padrão
 - Ele não pode ser redefinido
 - Ao definir construtores com parâmetros, é preciso inicializar todos os fields manualmente

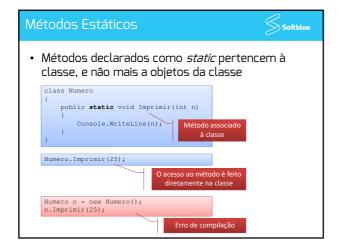
O Modificador *static*



- O modificador *static* pode ser utilizado em diversos locais
 - Fields
 - Métodos
 - Construtores
 - Classes







Elementos Estáticos e Não Estáticos



- Uma classe pode definir fields e/ou métodos estáticos e não-estáticos ao mesmo tempo
 - Não estáticos
 - Pertencem a cada objeto da classe
 - Estáticos
 - Pertencem à classe
- Regra para acesso
 - Elementos não-estáticos podem acessar elementos estáticos
 - Elementos estáticos não podem acessar elementos não-estáticos

Construtores Estáticos



- Construtores estáticos são chamados pelo CLR na primeira vez que a classe é utilizada
- Têm o objetivo de inicializar fields estáticos da classe



Construtores Estáticos

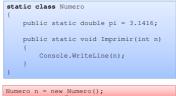


- Detalhes importantes sobre o construtor estático
 - Não recebe parâmetros
 - Não pode ser sobrecarregado
 - Não possui modificador de acesso (ex: *public*)
 - Executa apenas uma vez, independentemente do número de instâncias criadas
 - Executa antes dos construtores associados à instâncias
 - É chamado pelo CLR antes de criar a primeira instância da classe ou de acessar um elemento estático pela primeira vez

Classes Estáticas



• Uma classe definida com o modificador static não pode ter instâncias



A classe só pode ser *static* se contiver apenas elementos *static*

Erro de compilação

Constantes



• O modificador **const** pode ser aplicado a fields também



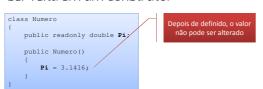
deve ser conhecido na compilação

Read-Only Fields São uma alternativa às constantes O valor não precisa ser atribuído no momento da declaração Uma vez definido, o valor não pode ser alterado Class Numero public readonly double Pi; public readonly não define o elemento como static

Read-Only Fields



• A inicialização de um read-only field só pode ser feita em um construtor



 Se o read-only field for definido como static, a inicialização só pode ocorrer em um construtor estático

Elementos Estáticos em Estruturas



- Estruturas também podem ter elementos static
 - Fields
 - Métodos
 - Construtores
- O funcionamento é igual se comparado às classes
- 0 *struct* em si não pode ser definido como *static*

