

Estrutura de Dados

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

“First, solve the problem.
Then, write the code.”

John Johnson

Estrutura de Dados

Objetivo da aula

Apresentação do docente

Revisão dos conceitos apresentados anteriormente pelo professor Nilton.

Quem é o docente ?



Orlando Saraiva Júnior

Mestre em Tecnologia (FT / Unicamp, 2013)

MBA em Gestão Estratégica de Negócios (Unifian, 2008)

Tecnólogo em Informática (CESET / Unicamp, 2005)

Assistente suporte acadêmico na Universidade Estadual Paulista (UNESP / campus Rio Claro)

Docente na Fundação Hermínio Ometto (FHO / Uniararas)

Autor do livro "Introdução à Orientação a Objetos com C++ e Python" (ISBN: 978-85-7522-548-6)

Ementa

Revisão dos conceitos básicos de tipos abstratos de dados. Pilhas, filas, alocação dinâmica, recursividade, listas encadeadas, tabelas de espalhamento e árvores. Aplicações das estruturas de dados em problemas computacionais.

Objetivos Gerais:

Aprofundar conhecimentos sobre criação e manipulação de tipos abstratos de dados: listas, pilhas, filas e árvores.

Objetivos Específicos:

Criar, manipular e aplicar, por meio de uma linguagem de programação, os tipos abstratos de dados: listas, pilhas, filas e árvores.

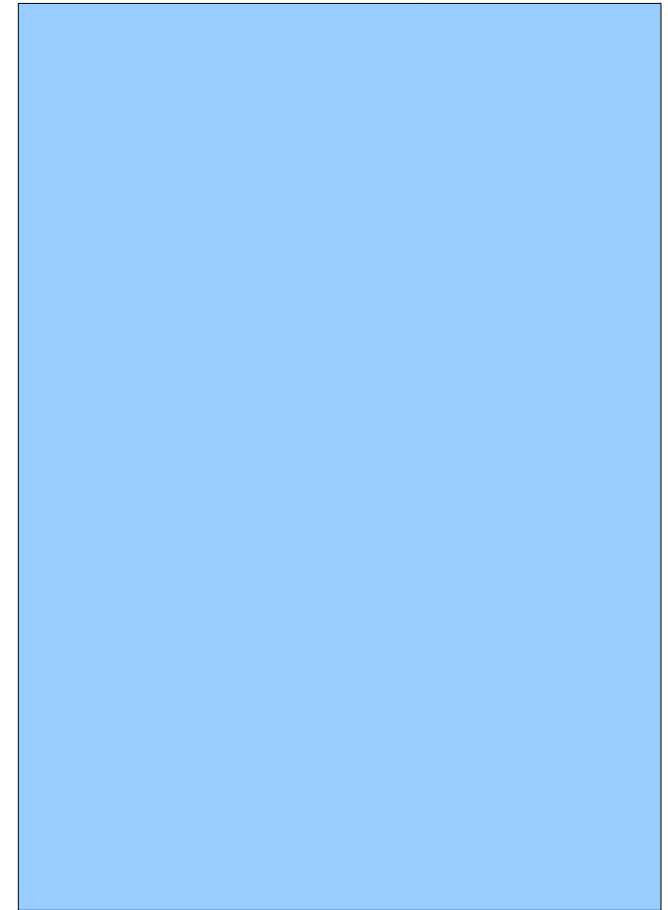
Programa 01

Programa 01.c

debugger na linha 8

numero_2
numero_1
main()

Pilha (stack)



heap

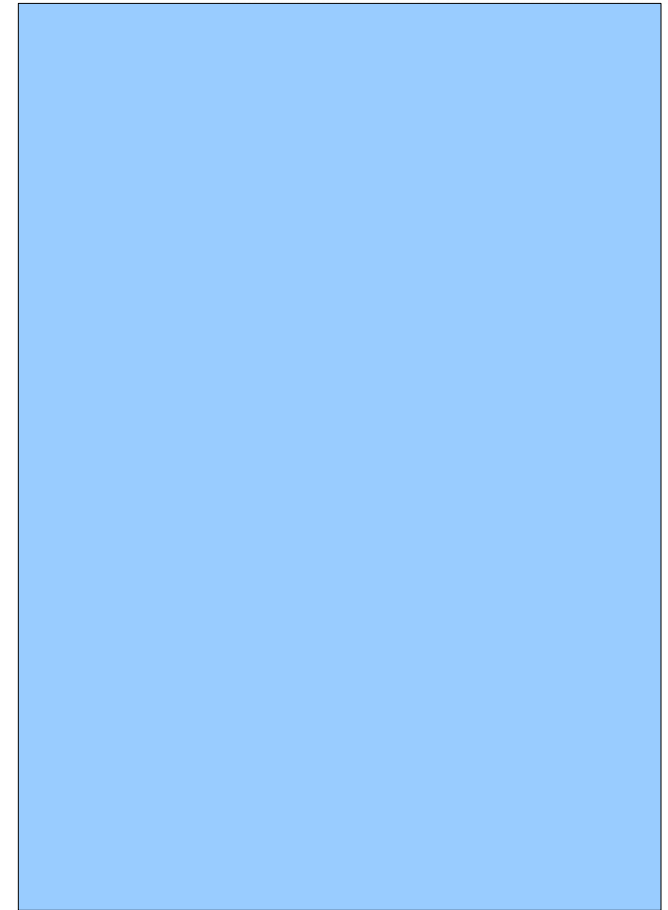
Memória estática

Programa 01.c

debugger na linha 11

numero_2: 5
numero_1: 5
main()

Pilha (stack)



heap

Memória estática

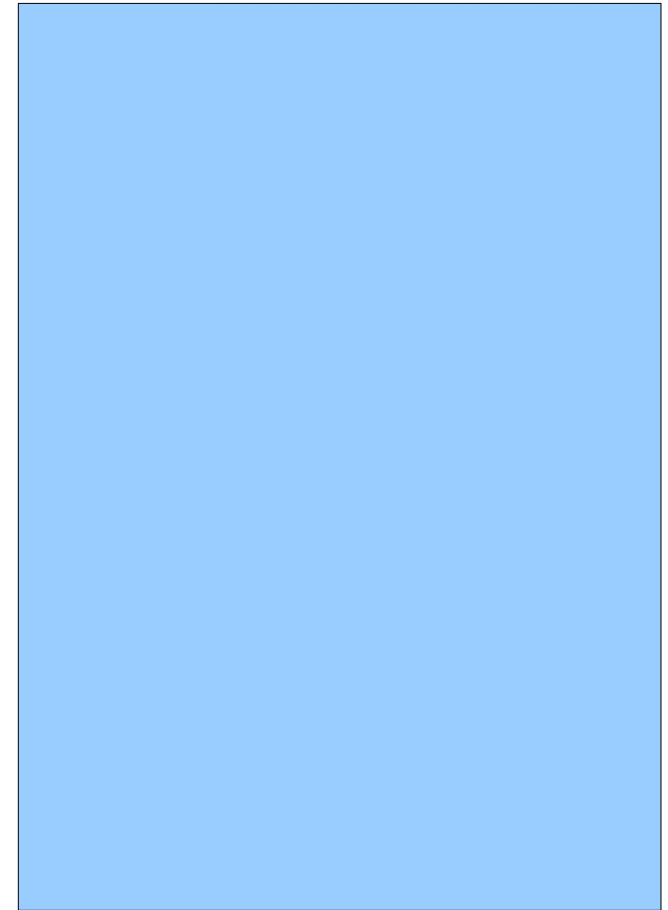
Programa 02

Programa 02.c

debugger na linha 27

numero_2
numero_1
main()

Pilha (stack)



heap

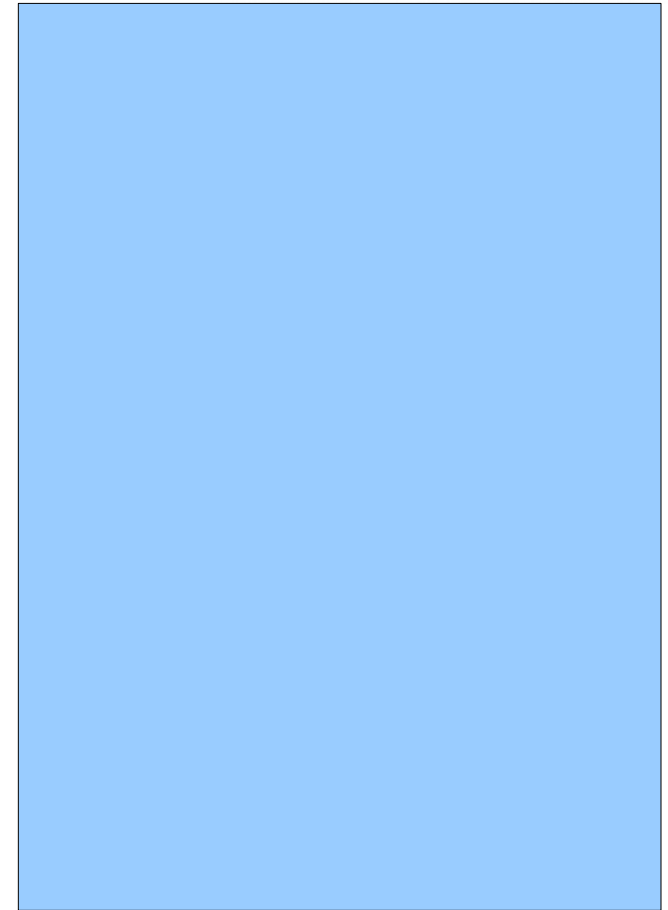
Memória estática

Programa 02.c

debugger na linha 18

a = 4.5
funcao_1()
numero_2
numero_1
main()

Pilha (stack)



heap

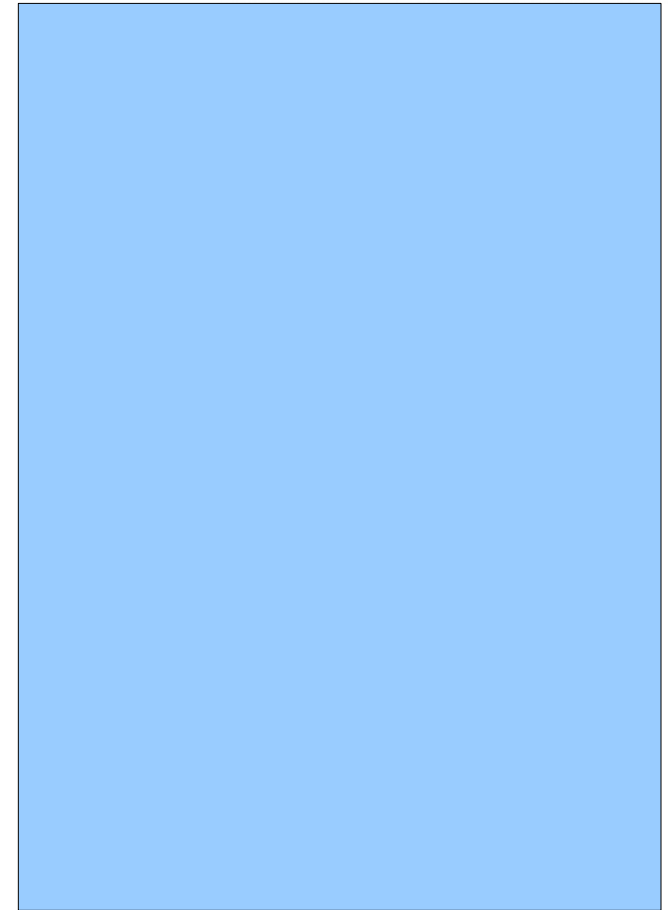
Memória estática

Programa 02.c

debugger na linha 12

b = 7
funcao_1_1()
a = 4.5
funcao_1()
numero_2
numero_1
main()

Pilha (stack)



heap

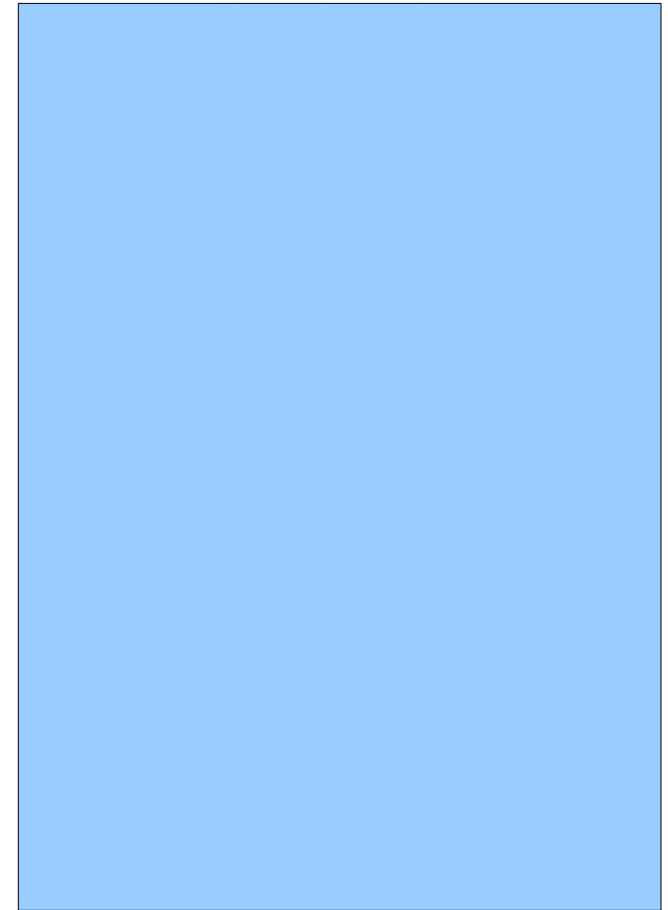
Memória estática

Programa 02.c

debugger na linha 20

a = 4.5
funcao_1()
numero_2
numero_1
main()

Pilha (stack)



heap

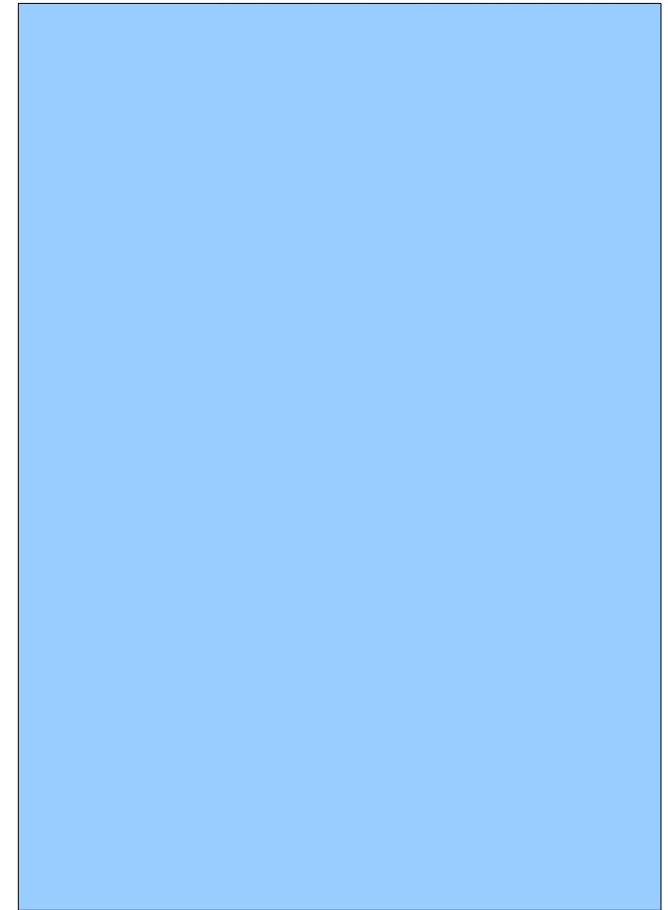
Memória estática

Programa 02.c

debugger na linha 49 (entrou na função na linha 38)

	auxiliar
	valor_1 = 5
	valor_2 = 6
nao_troca_valores(v_1, v_2)	
	numero_2 = 5
	numero_1 = 6
	main()

Pilha (stack)

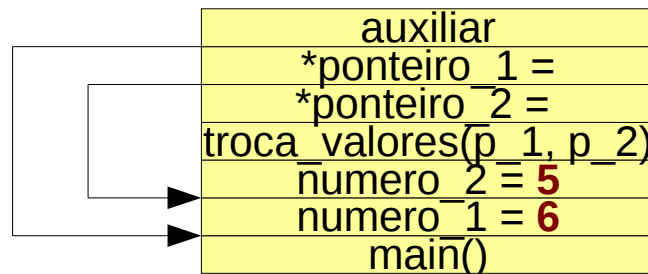


heap

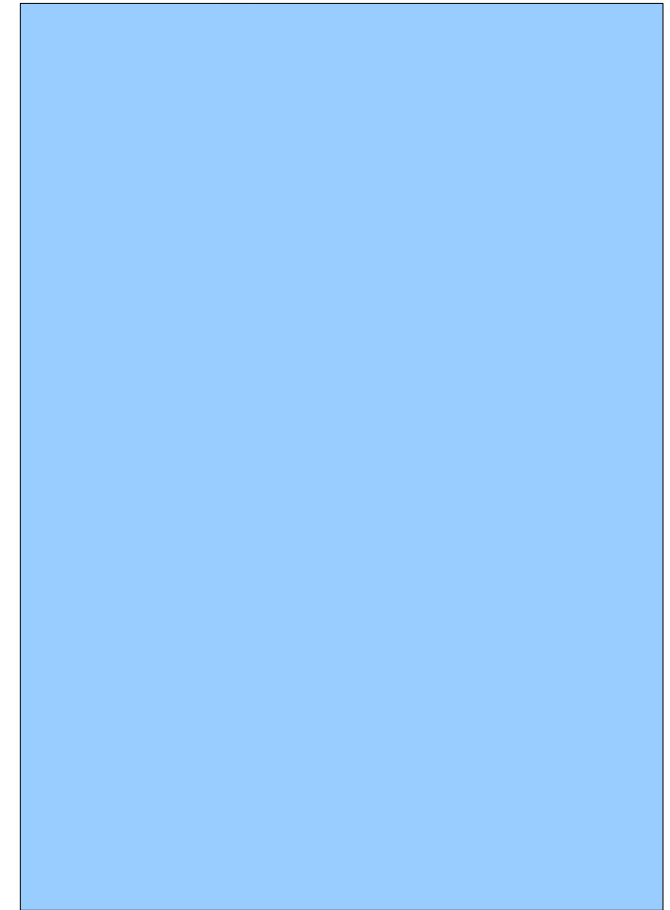
Memória estática

Programa 02.c

debugger na linha 58 (entrou na função na linha 41)



Pilha (stack)



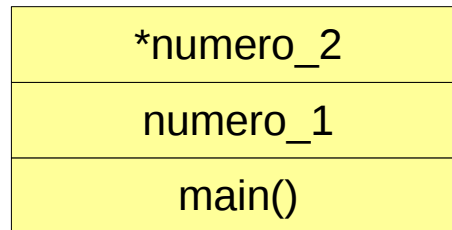
heap

Memória estática

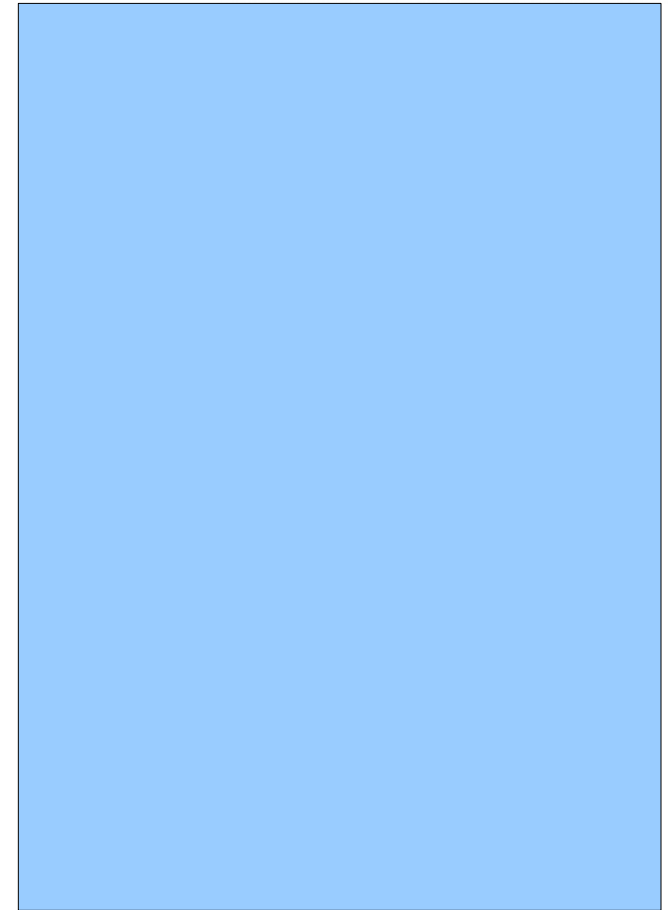
Programa 03

Programa 03.c

debugger na linha 9



Pilha (stack)

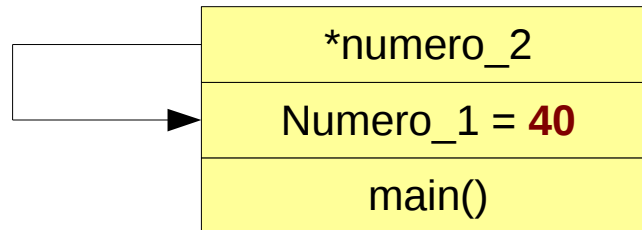


heap

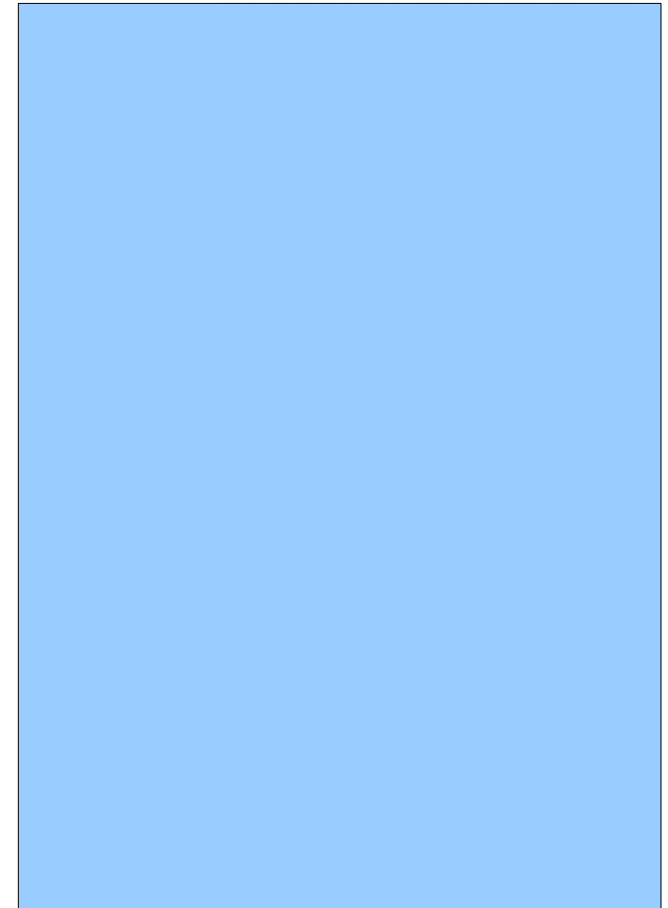
Memória estática

Programa 03.c

debugger na linha 12



Pilha (stack)

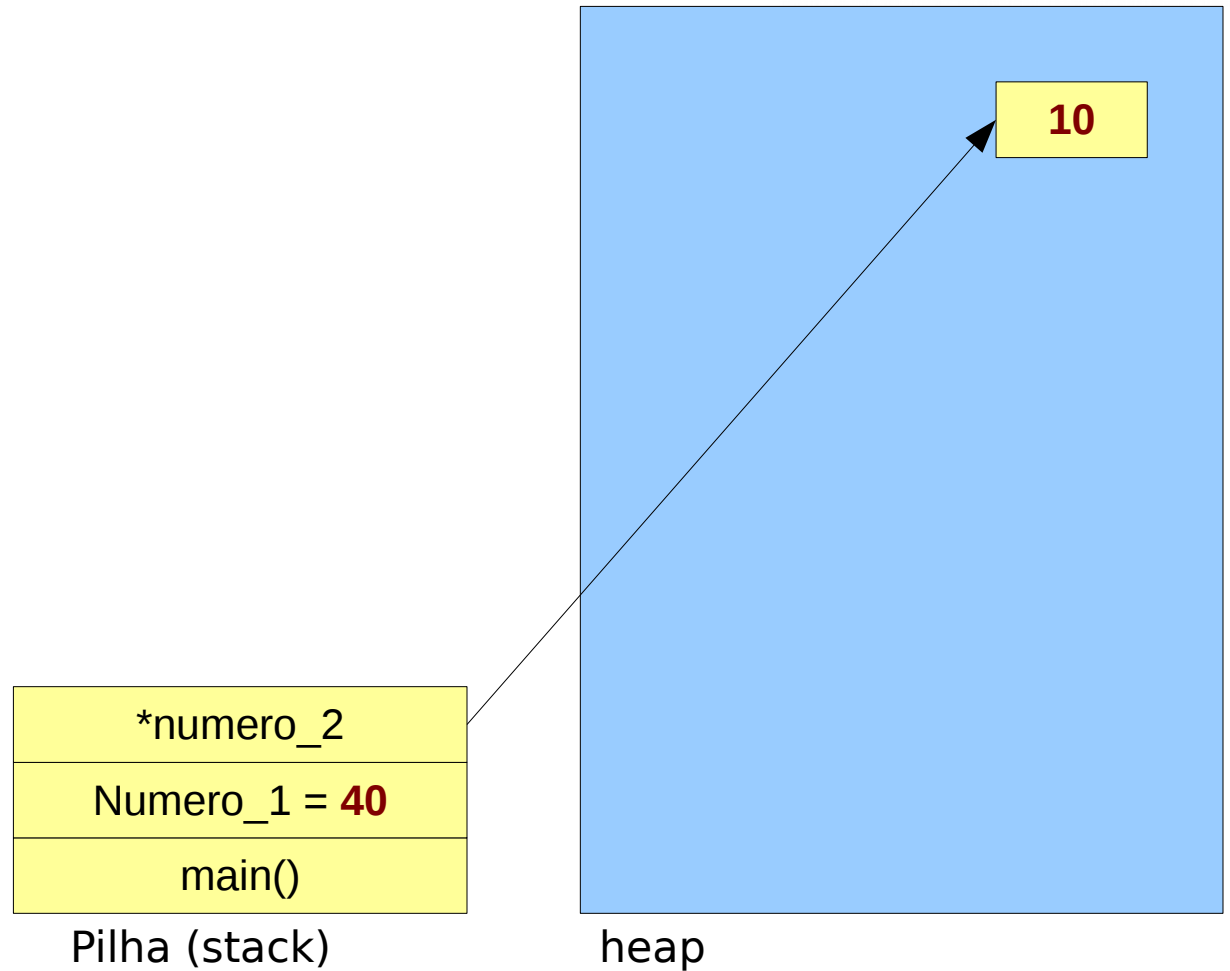


heap

Memória estática

Programa 03.c

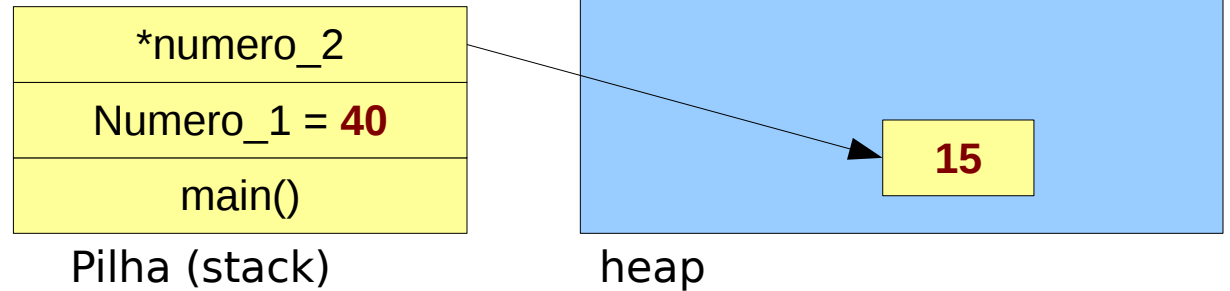
debugger na linha 16



Memória estática

Programa 03.c

debugger na linha 21



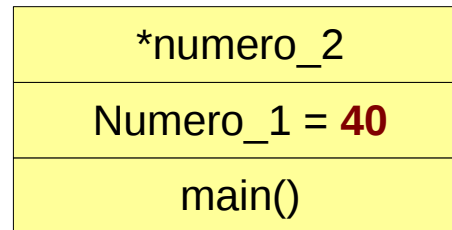
Memória estática

Programa 03.c

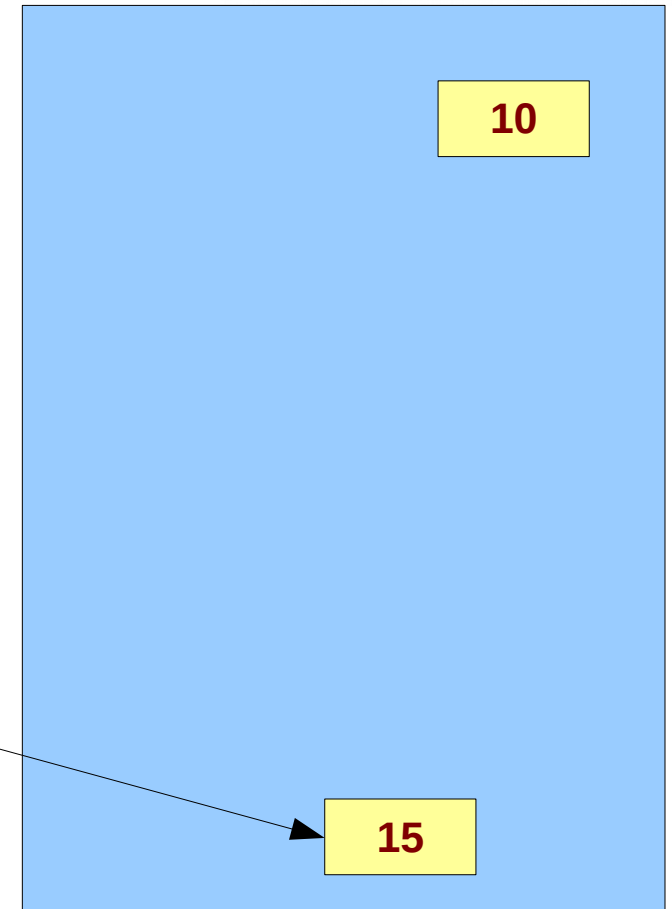
debugger na linha 21

E se o free(), linha 18 não existisse ???

LEAK de memória



Pilha (stack)



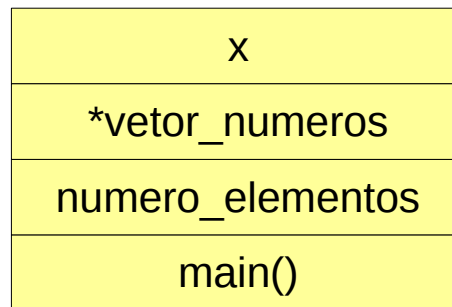
heap

Memória estática

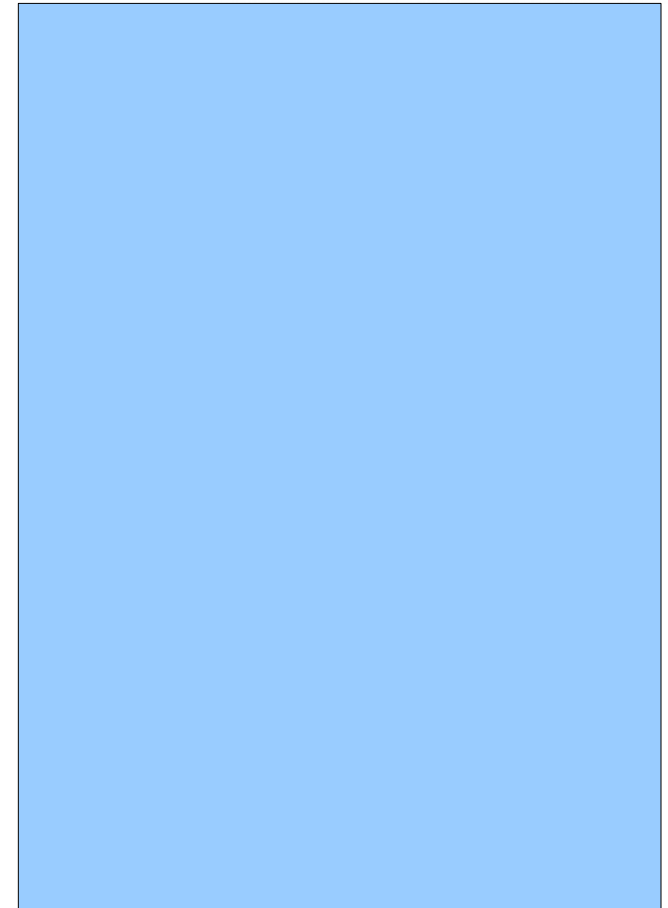
Programa 04

Programa 04.c

debugger na linha 11



Pilha (stack)

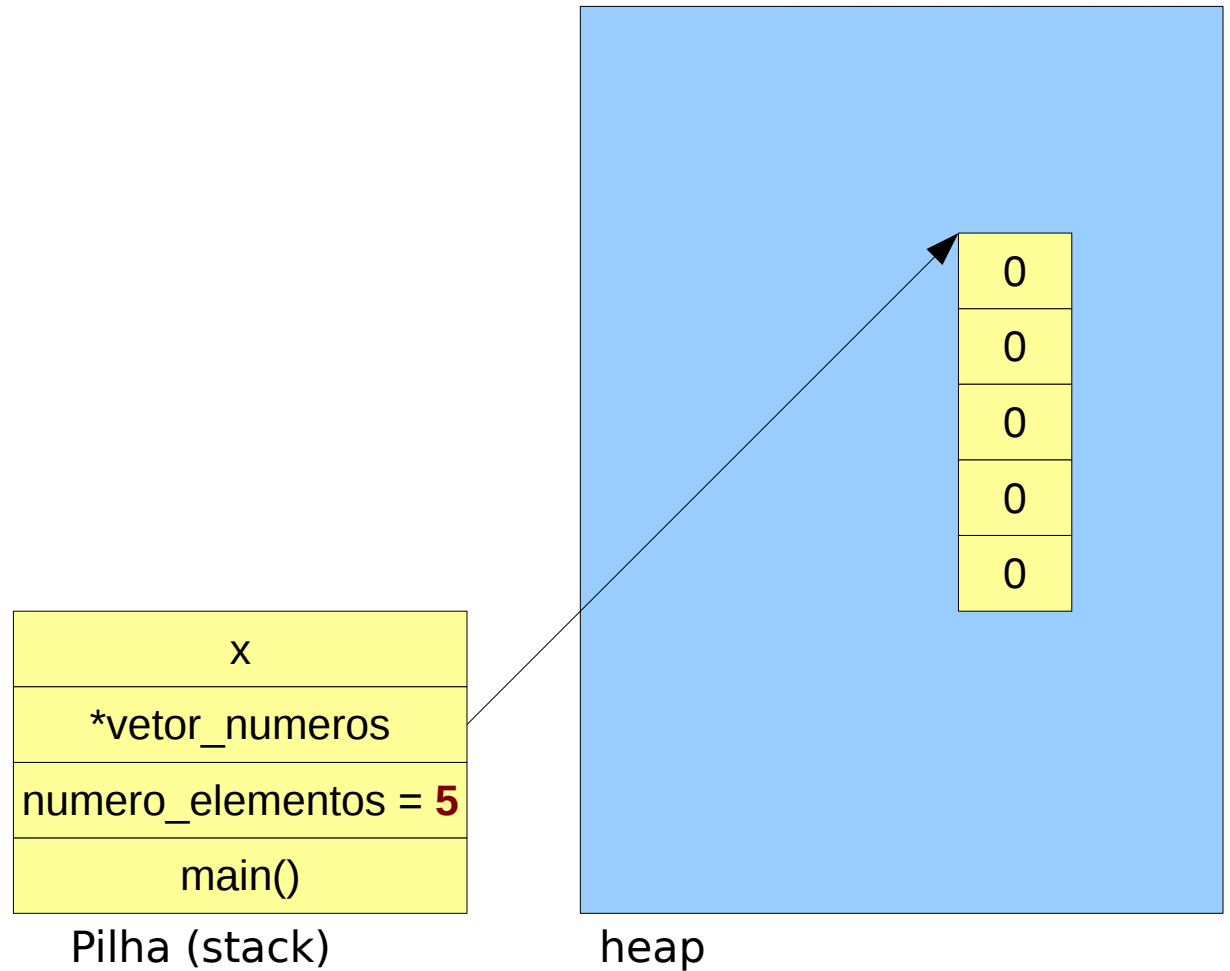


heap

Memória estática

Programa 04.c

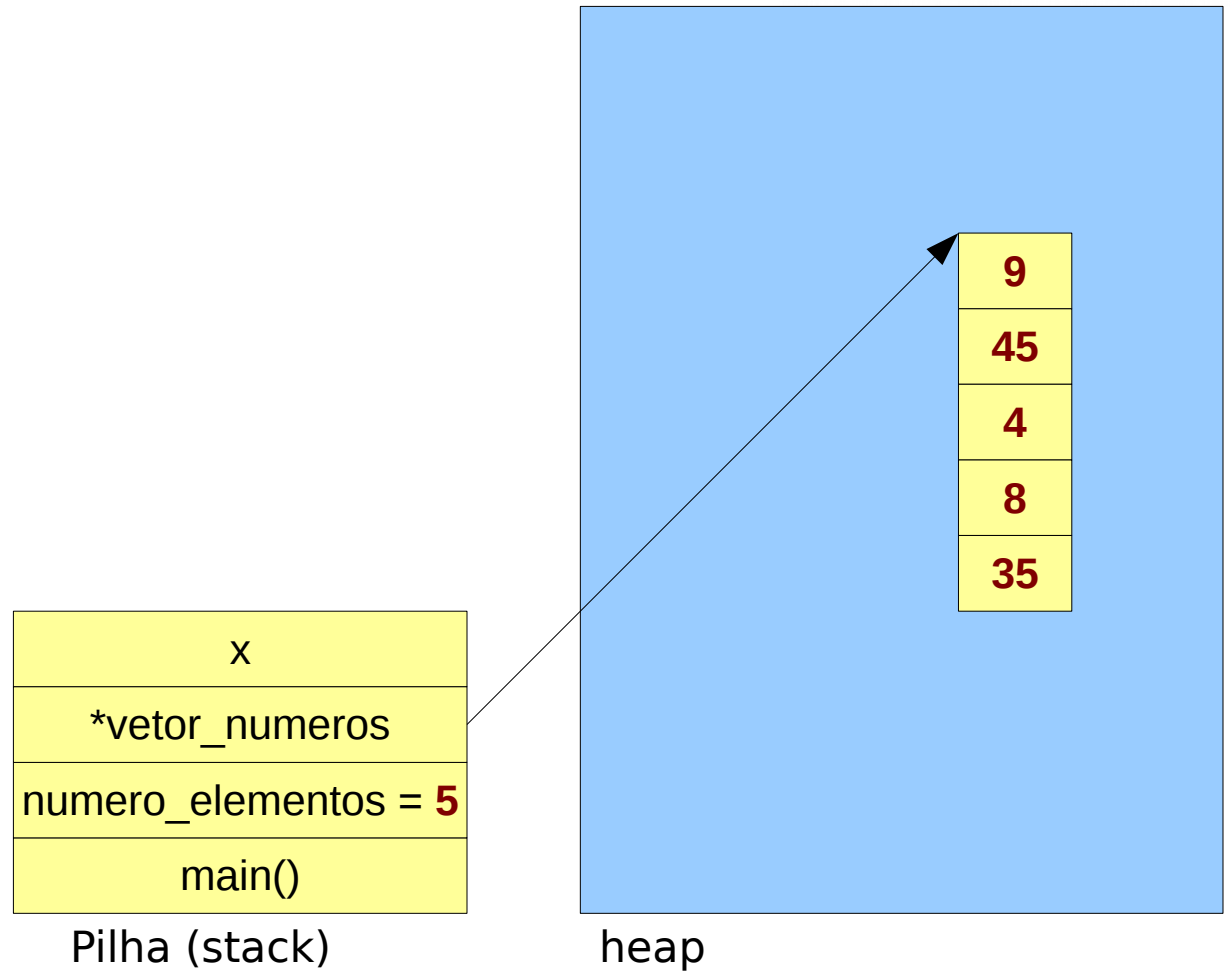
debugger na linha 14



Memória estática

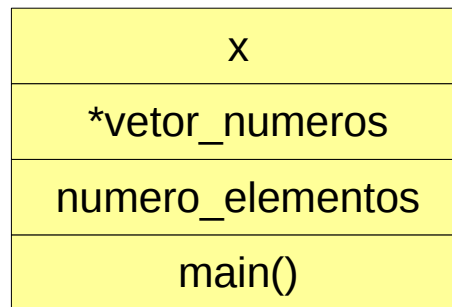
Programa 04.c

debugger na linha 20

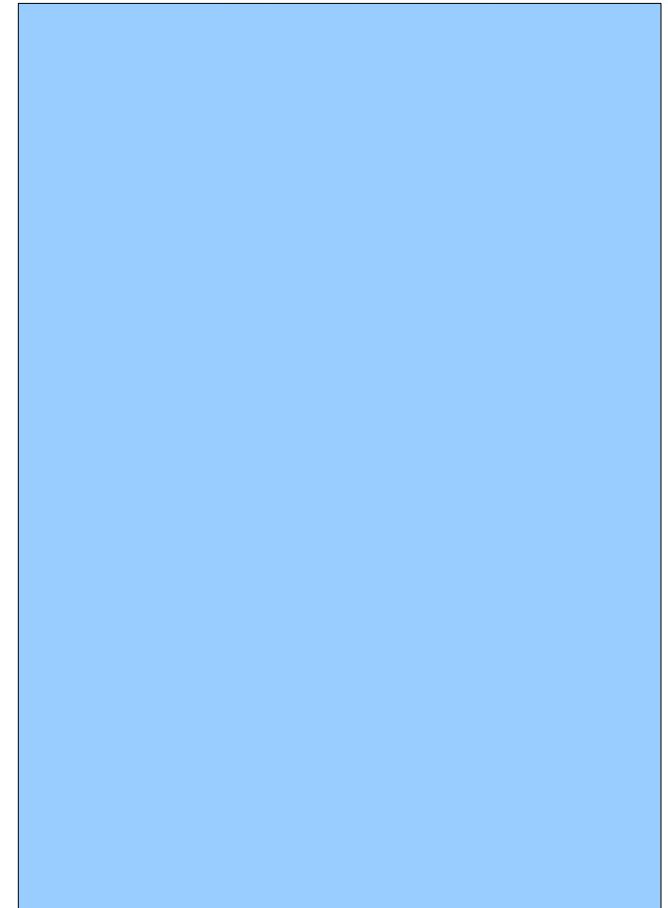


Programa 04.c

debugger na linha 25



Pilha (stack)



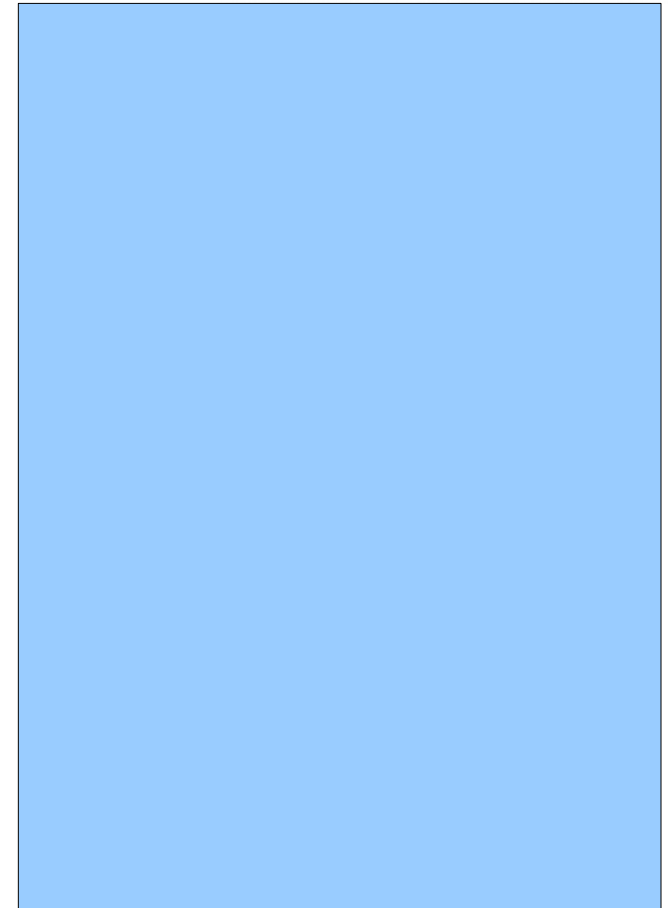
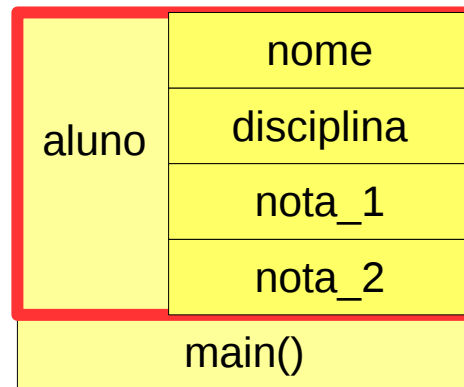
heap

Memória estática

Programa 05

Programa 05.c

debugger na linha 15



Memória estática

Pilha (stack)

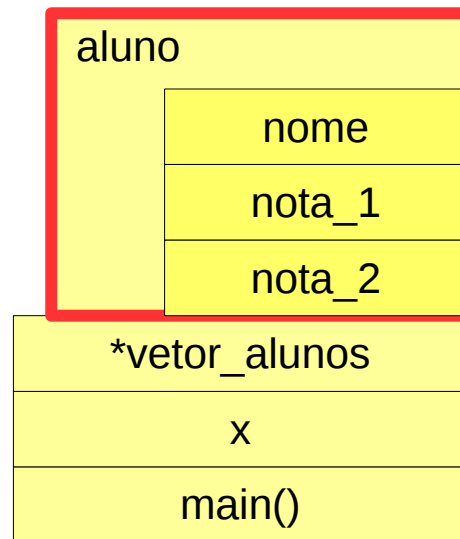
heap

Programa 06

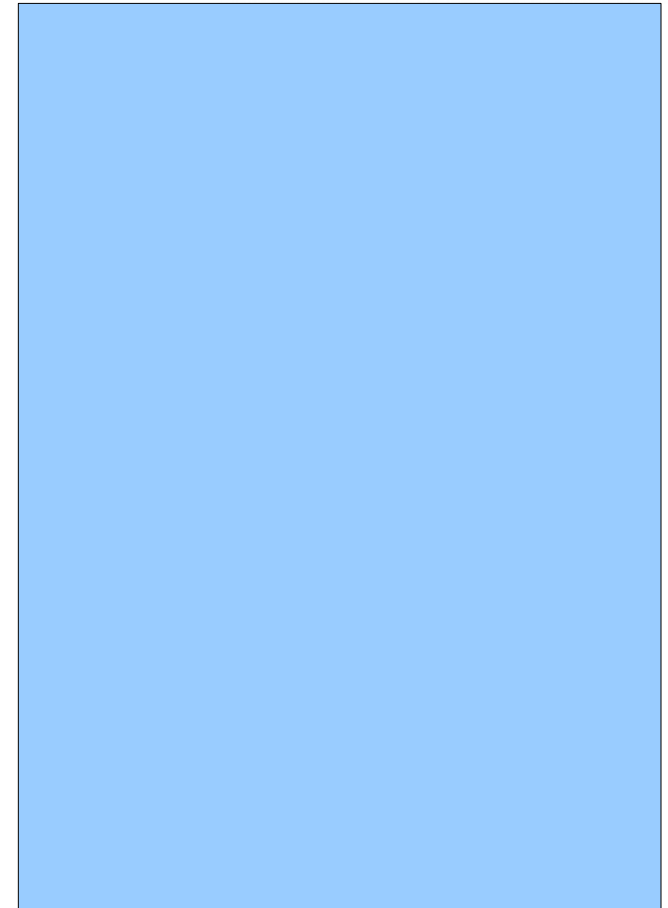
Programa 06.c

debugger na linha 19

contador = 0



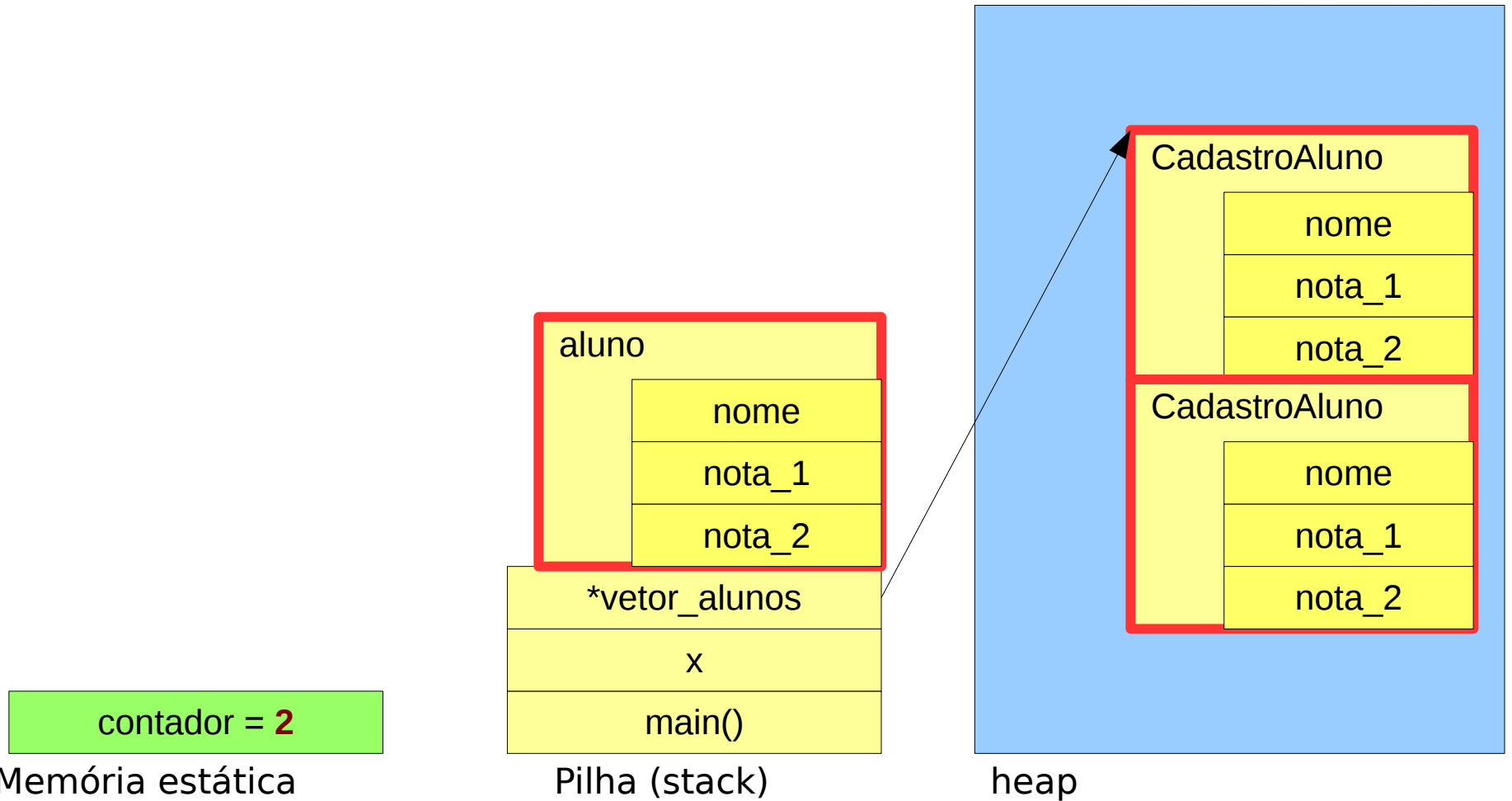
Pilha (stack)



heap

Programa 06.c

debugger na linha 24



Dúvidas

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

Programa 02: Altere o programa para que um estouro de pilha ocorra.

Programa 03: Altere o programa e teste ponteiros com outros tipos primitivos (float, double, por exemplo)

Programa 04: Altere o programa, para que preencha os valores do vetor automaticamente. Após esta alteração, teste o programa com diversos parâmetros. Tente causar um erro de execução.

Programa 06: Altere o programa para que o CadastroAluno seja mais sofisticado, permitindo armazenar mais informações do aluno. Faça uso de struct dentro de struct, se for necessário.