

O objetivo desse exercício é praticar alguns conceitos básicos de classes e objetos, bem como do container list da VPL.

Você deverá criar uma classe com um nome a sua escolha. Cada objeto dessa classe tem um atributo id do tipo inteiro. O valor desse campo pode ser atribuído de duas formas diferentes:

- \* Poderá ser gerado de forma automática e incremental quando um novo objeto for criado (O primeiro objeto vai ter o id 1, o segundo o id 2, etc).
- \* Poderá ser fornecido como parâmetro. Nesse caso, só ids negativos podem ser fornecidos e o id automático não é alterado.

Além disso, a sua classe deverá ter um atributo que conta o número de objetos existentes. Esse atributo deverá ser incrementado no construtor e decrementado do destrutor. Todos os atributos da sua classe devem ser privados. Sua classe deverá ter métodos para retornar o id de um objeto e também o número de objetos existentes no momento. Sua classe também deve ter um método que retorna o endereço de memória no qual o objeto está alocado.

O seu programa principal deverá criar uma lista de objetos e receber comandos para manipular essa lista. Mas especificamente, ele vai ser um loop que lê comandos da entrada (A, C, R, N, P, L, E) juntamente com eventuais parâmetros e faz as operações de acordo. Para cada comando, o sistema imprime uma mensagem com uma quebra de linha (endl) no final. Importante: as mensagens devem ser exatamente como especificado abaixo para evitar erros na correção automática. As operações são:

A: adiciona um novo objeto com id automático no final da lista. Deve ser impresso o id e o endereço de memória do objeto criado.

C: adiciona um novo objeto com id fornecido no início da lista. Deve ser impresso o id e o endereço de memória do objeto criado o "ERRO" caso o id não seja negativo.

R: remove (e deleta) um objeto da frente da lista. Deve ser impresso o id e o endereço de memória do objeto removido. Caso a lista esteja vazia deve ser impresso "ERRO"

N: imprime o valor do contador com número de objetos existentes.

P <i>: imprime o id e o endereço de memória do i-ésimo elemento da lista. Se i for um valor inválido (menor que 1 ou maior que o número de elementos) seu programa deverá imprimir "ERRO"

L: Imprime, para todos os elementos da lista, o id e o endereço de memória do objeto, sendo um elemento por linha com quebra de linha no final.

E: termina a execução do programa.

A sequência abaixo mostra um exemplo de entrada com a respectiva saída (os endereços podem variar de acordo com a execução):

A

1 0x5627ff3f02c0

C -10

-10 0x5627ff3f0710

C 5

ERRO

A

2 0x5627ff3f0750

L

-10 0x5627ff3f0710

1 0x5627ff3f02c0

2 0x5627ff3f0750

R

-10 0x5627ff3f0710

A

3 0x5627ff3f0710

R

1 0x5627ff3f02c0

A

4 0x5627ff3f02c0

N

3

P 5

ERRO

P 3

4 0x5627ff3f02c0

L

2 0x5627ff3f0750

3 0x5627ff3f0710

4 0x5627ff3f02c0

E