Деструктори

Жизнен цикъл на обект

- За обекта се заделя памет и се свързва с неговото име
- Извиква се подходящ конструктор на обекта
- ... (достъп до компоненти на обект, изпълняване на операции)
- Достига се края на областта на действие на обекта
- Извиква се деструкторът на обекта
- Заделената за обекта памет се освобождава

За какво служи деструкторът?

- Извършване на заключителни действия
  - затваряне на файл, мрежова или друга връзка
  - освобождаване на заделена памет
  - уведомяване за разрушаването на обекта
- Деструкторът е противоположен на конструктора
- Извиква се автоматично при унищожаване на обекта
  - излизане от област на действие
  - извикване на delete или delete[]

Дефиниране на деструктор

- <име_на_клас>::~<име_на_клас>() { <тяло> }
- Всеки клас може да има само един деструктор
- Ако не бъде дефиниран явно деструктор, се дефинира системен такъв с празно тяло
- Ако обектът използва динамична памет, системният деструктор няма да я освободи
  - трябва да се дефинира явен такъв

Управление на динамична памет

- След като блок от динамична памет бъде заделен, указателят към нея може да бъде копиран на много места в програмата
- Коя част от програмата носи отговорност за освобождаване на динамичната памет, когато вече не е нужна?

Управление на динамична памет

- Стратегия 1: собственост
  - точно един от указателите се счита за „собственик“
  - програмата трябва да включва логика, която гарантира, че указателите, които не са собственици няма да достъпват паметта след нейното освобождаване
  - често „собственикът“ е поле на обект и паметта се освобождава от деструктора
  - често се пази единствен указател към заделената памет, която се достъпва чрез селектор на обекта
  - докато обектът е „жив“, паметта е валидна
  - има конструкции, които позволяват „прехвърляне“ на собственост

Управление на динамична памет

- Стратегия 2: умни указатели
  - споделена собственост
  - поддържа се общия брой на всички указатели към заделената памет
  - когато всички указатели бъдат „изгубени“, паметта се освобождава
  - „слаби“ указатели — указатели, които не са собственици. Не се гарантира, че сочат към валидна памет

Разрушаване на масиви от обекти

- При заделяне на масиви от обекти, се извиква конструктор за всеки обект
  - ако масивът е заделен в стека, можем да укажем отделен конструктор за всеки обект
  - ако масивът е заделен динамично, извиква се конструкторът по подразбиране
- При разрушаване на масив от обекти, за всеки един от тях трябва да се извика деструктор
  - при масиви, заделени в стека това става автоматично
  - при масиви, заделени динамично, трябва да бъде използвана операцията delete[], а не delete!
  - delete ще извика деструктор само на първия обект от масива!