**21. Дефиниране на конструктори и деструктори. Дефиниране на методи на класа.**

Конструктурът на един клас се извиква всеки път, когато се създава обект от този клас.

И поради това всякаква инициализация, която трябва да се извърши за даден обект,

може да се изпълни автоматично от функцията конструктор.

Конструкторът има същото име като името на класа, към който принадлежи, и не

притежава тип на връщан резултат. Един клас може да има повече от един конструктор,

с различни аргументи. Ако в класът липсва дефиниран конструктор, то той се добавя

автоматично от компилатора, като той само заделя памет за член-променливите без

никаква инициализация, и без параметри. Общият вид на конструктора е:

<име-на-клас>::<име-на-клас>(<списък формални аргументи>)

{ //Тяло на конструктор}

Деструкторът се извиква автоматично при разрушаването на обект от класа.

Деструктора служи главно за освобождава заделената за обекта динамична памет.

Деструкторът има същото име като класа, но предшествано от символа ~. Деструкторът

не може да връща резултат и не може да има аргументи. В един клас може да има само

един деструктор. Ако липсва, то той бива генериран автоматично от компилатора, като

в него просто се освобождават член-променливите на класа. Общият вид на

деструктора е:

<име-на-клас>:: ~<име-на-клас>()

{ //Тяло на деструктор }

Конструкторите и деструкторите не се наследяват автоматично и трябва ръчно да се

предефинират при наследени класове.

Функциите, членове на един клас се наричат методи(член-функции).

Дефиницията на член-функция има следния синтаксис:

<тип-връщан-резултат> <име-на-клас>::<име-на-функция>(<списък с параметри>)

{ // Tяло на функцията }