**Работа № 3.  
Классы. Программирование линейных алгоритмов с использованием конструктора, деструктора, friend - функции инициализации set() и функции вывода результатов print()**

**Теоретические сведения**

**Задание к работе**

**Варианты заданий**

**Контрольные вопросы**

**Цель работы –** изучить основные способы работы по созданию конструктора класса с захватом динамической памяти и деструктора для ее освобождения, применение friend – функции, и изучение ее особенностей.

## Теоретические сведения

Конструктор класса

Конструктор – это метод класса, имя которого совпадает с именем класса. Конструктор вызывается автоматически после выделения памяти для пе­ременной и обеспечивает инициализацию компонент – данных. Конструктор не имеет никакого типа (даже типа void) и не возвращает никакого значения в ре­зультате своей работы. Конструктор нельзя вызывать как обычную компо­нентную функцию в программе. Для класса может быть объявлено несколько конструкторов, различающихся числом и типами параметров. При этом даже если для объектного типа не определено ни одного конструктора, компилятор создает для него конст­руктор по умолчанию, не использующий параметров, а также конструктор ко­пирования, необходимый в том случае, если переменная объектного типа передается в конструктор как аргумент. В этом случае создаваемый объект будет точной копией аргумента конструктора.

class my\_Fun{

// компоненты-данные   
double x;  
unsigned size;

public:

// объявление конструктора 1 (с параметрами)  
my\_Fun (double X=0);

// объявление конструктора 2 (без параметров)  
my\_Fun(void);

// объявление и описание деструктора  
~my\_Fun ()  
{  
cout<<"Destroyed object... "<<endl;  
}

// описание конструктора 1my\_Fun::my\_Fun (double X){cout<<"Constructor1...."<<endl;x=X;}// описание конструктора 2my\_Fun::my\_Fun (void){cout<<"Constructor2..."<<endl;x=5.0;}}

Деструкторкласса

Еще одним специальным методом класса является деструктор. Деструк­тор вызывается перед освобождением памяти, занимаемой объектной пере­менной,  и предназначен для выполнения дополнительных действий, связан­ных с уничтожением объектной переменной, например, для освобождения ди­намической памяти, закрытия, уничтожения файлов и т.п.  Деструктор всегда имеет то же имя, что и имя класса, но перед именем записывается знак ~ (тильда). Деструктор не имеет параметров и подобно кон­структору не возвращает никакого значения. Таким образом, деструктор не может быть перегружен и должен существовать в классе в единственном эк­земпляре. Деструктор вызывается автоматически при уничтожении объекта. Таким образом, для статически определенных объектов деструктор вызывает­ся, когда заканчивается блок программы, в котором определен объект (блок в данном случае – составной оператор или тело функции). Для объектов, память для которых выделена динамически, деструктор вызывается при уничтожении объекта операцией delete.

Дружественная функция(friend)

В языке С++ одна и та же функция не может быть компонентом двух разных классов. Чтобы предоставить функции возможность выполнения действий над различными классами можно определить обычную функцию языка С++ и предоставить ей право доступа к элементам класса типа private, protected. Для этого нужно в описании класса поместить заголовок функции, перед которым поставить ключевое слово friend. Дружественная функция не является методом класса, не зависит от позиции в классе и спецификаторов прав доступа. Friend – функции получают доступ к членам класса через указатель, передаваемый им явно. Можно сделать все функции класса Y друзьями класса X в одном объявлении.

## Задание к работе

Общая постановка. Пользовательский класс Х должен содержать необходимые элементы-данные, которые создаются в динамической области памяти.

* Конструктор для их создания (операция new) и установки их начальных значений: Х()**;**
* деструктор: ~Х()**;**
* friend – функция печати: friend void print()**;**
* функция, решающая поставленную задачу: friend Void Run()**.**

Код методов и функций – вне пространства определения класса.

## Варианты заданий

1. lab_clip_image035  
     
   При x=0.1722, y=6.33, z=3.25lab_clip_image037 **lab_clip_image039=-205.305.**