**Учебная практика № 5 Динамическое выделение памяти под объекты.**

Цель работы: научиться работать с динамическим массивом. (*2 часа)*

Учебник Васильева А.Н. Самоучитель С++. Глава 8. Работа с объектами. Массивы объектов, динамическое выделение памяти под объекты (стр. 285 - 305). Книга имеется в свободном доступе в интернете.

**Динамическая память** Выделяется и освобождается — с помощью специальных инструкций (т. е. по инициативе разработчика). Это позволяет по ходу работы программы контролировать и корректировать объём используемой памяти и, следовательно, создавать программы способные обрабатывать большие объёмы данных, обходя ограниченность физической памяти машины.

Доступ к динамической памяти возможен только через указатели, т. е. нельзя создавать в неё переменные, но можно выделять её фрагменты и связывать из с некоторым указателем.

**Работа с динамической памятью**

Выделяется память с помощью оператора new, а освобождается — с помощью оператораdelete.

В момент, когда динамическая память выделена, она должна быть связана с некоторым указателем, подходящего типа (при выделении указывается тип и количество необходимых ячеек данного типа).

Если не освобождать динамическую память, то она будет занята до завершения программы, что неприемлемо.

При выделении одной динамической переменной (одной ячейки памяти), можно сразу инициализировать её значение:

int\* p;

p = new int(10);

cout << \*p; // 10

delete p; // память освобождена

Можно выделять сразу несколько ячеек динамической памяти, получая динамический массив.

Для этого его размер указывается в квадратных скобках после типа.

Чтобы удалить динамический массив и освободить память используется оператор delete[].

int\* p;

p = new int[13];

for (int i=0; i<13; i++) {

\*(p+i) = i + 1;

cout << \*(p+i) << ' '; // 1 2 3 ... 13

}

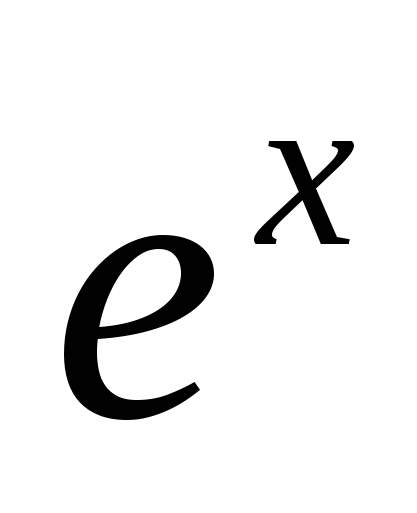
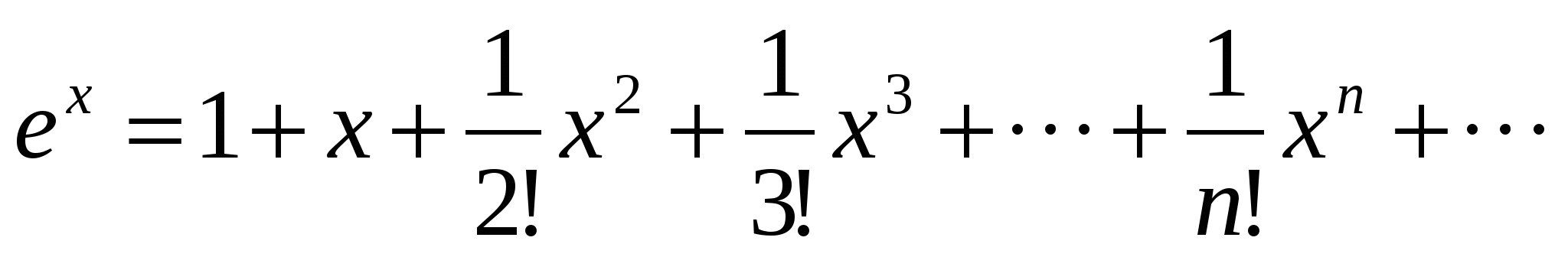
delete[] p; // память освобождена, из неё удалены все элементы

Сразу, после создания динамический массив автоматически заполняется нулями (в отличии от обычного массива в статической или стековой памяти).

Очень важно после использования динамической памяти не забывать освобождать её в нужный момент!

**Задание 1. Вычислить сумму ряда, используя рекуррентное соотношение. Создать класс, у которого поле -динамический массив. Элементами массива являются слагаемые ряда. Массив заполняется при создании соответствующего объекта. Аргументами конструктору передается верхний индекс ряда *int n* и аргумент ряда *double x.* Включить программу выдачу расчета 10 полей.**

Образец:

//Листинг 9.9. Ряд для экспоненты **Ряд Маклорена функции **. 

#include "stdafx.h"

#include <iostream>

using namespace std;

class MyExp{

public:

int n;

double \*p;

MyExp(int i,double x){ // конструктор

n=i;

p=new double[n+1];

p[0]=1;

for(int k=1;k<=n;k++) p[k]=p[k-1]\*x/k;

}

~MyExp(){delete [] p;}

};

int main(){

int n,i;

double x,s=0;

cout<<"enter n= ";

cin>>n;

cout<<"enter x= ";

cin>>x;

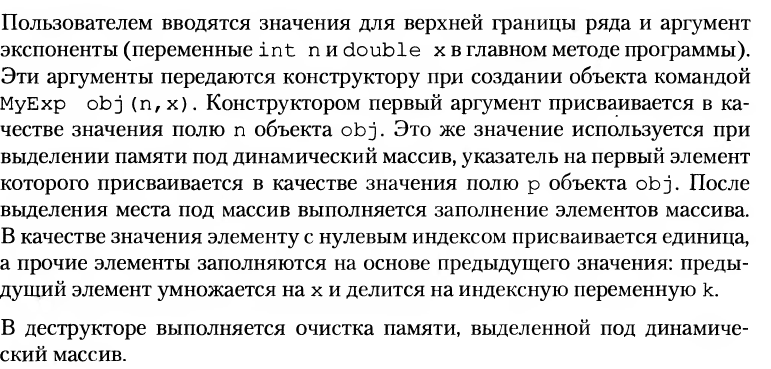
MyExp obj(n,x);

for(i=0;i<=n;i++) s+=obj.p[i];

cout<<"exp("<<x<<")= "<<s<<endl;

return 0;}





**Варианты заданий УП 5 По аналогии с примером «Ряд для экспоненты» написать программу для вычисления суммы ряда**

|  |  |
| --- | --- |
| вариант 1 для |x |<1  \operatorname{arctg}\ x = x - \frac{x^3}{3}+ \frac{x^5}{5} - \cdots\ = \sum^{\infin}_{n=0} \frac{(-1)^n}{2n+1} x^{2n+1} | вариант 7 Разложение в ряд Тейлора функции e^x |
| вариант 2 \operatorname{ch}\, \left(x\right) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots = \sum^{\infin}_{n=0} \frac{1}{(2n)!} x^{2n}, x\in\mathbb{C} | вариант 8\operatorname{sh}\, \left(x\right) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots = \sum^{\infin}_{n=0} \frac{1}{(2n+1)!} x^{2n+1}, x\in\mathbb{C} |
| вариант 3Разложение в ряд Тейлора функции ln(1-x) | вариант 9Разложение в ряд Тейлора функции ln(1+x) |
| вариант 4Разложение в ряд Тейлора функции cos(-x) | вариант 10\sin x =  x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots\ = \sum^{\infin}_{n=0} \frac{(-1)^n}{(2n+1)!} x^{2n+1}, x\in\mathbb{C} |
| вариант 5 | вариант 11 |
| вариант 6 Разложение в ряд Тейлора функции 1/(1+х) |  |
| вариант 12 | |
| Вариант 13 | |

**Задание 2. Динамическое выделение памяти под объекты.**

вариант 1

Создайте перегруженную функцию proisv (), которая перемножает 2 числа (объекта), где аргументы целый (int), вещественный (double) тип. Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 2

Создайте перегруженную функцию inc (), которая к аргументу прибавляет 1, где аргументы целый (int), вещественный (double) тип, символьный (char) тип. Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 3

Создайте перегруженную функцию decr (),которая от аргумента вычитает 1 , где аргументы целый (int), вещественный (double) тип, символьный (char) тип. Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 4

Создайте перегруженную функцию maxim() , которая находит максимум для двух переменных типа целые (int), вещественные (double), символьные (char).

Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 5

Создайте перегруженную функцию minimum() , которая находит минимум для двух переменных типа целые (int), вещественные (double), символьные (char).

Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 6

Создайте перегруженную функцию sub() , которая находит разность для двух переменных типа целые (int), вещественные (double).

Память под числа (объекты) выделяется динамически.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 7

Создайте перегруженную функцию minimum() , которая находит минимум для двух переменных типа целые (int), вещественные (double), символьные (char).

Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 8

Создайте перегруженную функцию proisv (), которая перемножает 2 числа (объекта), где аргументы целый (int), вещественный (double) тип. Память под числа (объекты) выделяется динамически.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 9

Создайте перегруженную функцию inc (), которая к аргументу прибавляет 1, где аргументы целый (int), вещественный (double) тип, символьный (char) тип. Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 10

Создайте перегруженную функцию decr (),которая от аргумента вычитает 1 , где аргументы целый (int), вещественный (double) тип, символьный (char) тип. Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 11

Создайте перегруженную функцию maxim() , которая находит максимум для двух переменных типа целые (int), вещественные (double), символьные (char).

Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса

вариант 12

Создайте перегруженную функцию minimum() , которая находит минимум для двух переменных типа целые (int), вещественные (double), символьные (char).

Память под числа (объекты) выделяется динамически, адреса объектов вывести на экран.

С помощью перегруженных

1. конструкторов
2. методов класса