Лабораторная работа № 3
ТЕМА: **Введение в объектно-ориентированное программирование на С++.**
ЦЕЛЬ: Изучение основ разработки объектно-ориентированных программ (ООП). Создание класса по обработке данных массива.

1 Теоретические вопросы подготовки к работе:

1. Организация консольного и файлового ввода-вывода в С++.
2. Структурированные типы языка С++.
3. Формирование области видимости имени в С++.
4. Понятия – «объект» и «класс». Инкапсуляция. Конструкторы и деструкторы.

2 Контрольные вопросы по теме:

1. Определение типа с помощью оператора typedef в С++.
2. Особенности ввода-вывода с помощью <iostream>.
3. Приведите описание типов (перечисление, запись, объединение, поле бит) и способы доступа к их элементам в С++.
4. Что такое инкапсуляция?
5. Приведите описание понятий – «объект» и «класс».
6. Дайте определение области действия «класс» и доступа к элементам класса. Спецификаторы доступа. Использование указателя this.
7. Выполните сравнение описателей struct и class в С++.
8. Опишите принцип ООП – отделение интерфейса от реализации.
9. Определение встраиваемых функций-членов. Преимущества использования.
10. Использование константных объектов и реализация принципа наименьших привилегий.
11. Перегрузка операций, операторов, функций.
12. В каких случаях используется перегрузка, переопределение или сокрытие?
13. Назначение и виды конструкторов.
14. Назначение и виды деструкторов.
15. Формирование и удаление массива объектов класса.

3 Индивидуальные задания:

3.1 Основные задания:

Напишите программу по обработке массива, как объекта созданного Вами класса «Массив», согласно варианту и выполните на тестовых данных. В отчете представьте листинг программы и результатов вычислений. Операции с массивами следует реализовать программно, используя указатели, а не индексы:

1) Дано целое N, вещественная матрица размера N x 5. Найти среднее арифметическое каждого из столбцов, имеющих четные номера и каждой из строк, имеющей нечетные номера.

2) Дана вещественная матрица M x N. Найти 2 суммы: наибольших значений элементов ее строк, наименьших значений ее столбцов.

3) Дана вещественная матрица порядка 5. Получить целочисленную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше суммы 2-х элементов, расположенных в его строке -

первого элемента в строке и элемента, расположенного на главной диагонали в этой строке, и равен нулю в противном случае.

4) Дана вещественная, квадратная матрица порядка N. Найти сумму элементов той строки в которой разница между максимальным и минимальным элементами является максимальной.

5) Даны натуральные числа i, j, вещественная матрица размера M x N. Поменять местами в матрице i-й и j-й столбцы.

6) Дана вещественная квадратная матрица порядка M. Упорядочить строки матрицы по возростанию диагональных элементов.

7) Дано целое число N. Сформировать квадратную матрицу порядка N:

$a(i, j) = \sin(i - j) + \cos(i + j)$;

Упорядочить строки матрицы по росту сумм элементов.

8) Дана вещественная квадратная матрица порядка N. Удалить из матрицы элементы главной диагонали.

9) Дано целое число N. Сформировать матрицу a(i,j) порядка N, для которой:

$a(i, j) = \sin(i + j + k)$;

где k – псевдослучайное число. Упорядочить строки матрицы по убыванию минимальных элементов строк.

10) Дана целочисленная квадратная матрица порядка 5. Удалить из матрицы те строки, для которых количество ненулевых элементов больше количества нулевых элементов в строке.

11) Дано натуральное число N, вещественная матрица размера N x N, вещественное число x. Получить последовательность из 0 и 1 $(b_1, b_2, ..., b_N)$, где $b_i = 1$, если элементы i-й строки не превосходят x, и $b_i = 0$, в противном случае.

12) Даны три матрицы размером M x N. Упорядочить по росту сумм элементов строк строки той матрицы, у которой больше нулевых строк.

13) Дана целочисленная матрица порядка NxN. Получить $b_i, i = \overline{1, N}$ по следующей формуле:

$b(i) = \max(a(i, j)) * \min(a(j, i))$,

причем max и min берутся по j.

14) Дано целое N. Сформировать матрицу a(i,j) порядка N, где

$a(i, j) = \cos(i * 0.5 + n * k)$,

где k- псевдослучайное число. Упорядочить строки матрицы по росту максимальных, положительных элементов строк.

15) Даны целые числа $a_1, a_2, ..., a_n$. Получить целочисленную матрицу b(i,j) порядка n, для которой:

$b(i, j) = a(i) - 3 * a(j)$;

Упорядочить строки матрицы по неубыванию разницы между первым и последним элементами строк.

3.2 Усложненные задания:

Определите класс для дробей - рациональных чисел, являющихся отношением 2-х целых чисел. Напишите функции-члены для сложения, вычитания, умножения и деления дробей и приведения дроби к наименьшему знаменателю. Выполните тестирование и отладку созданной Вами программы обработки дробей.

Приложение А

Функции ANSI C++ по работе с консолью и файлами (iostream)

Библиотеки потоков C++ базируются на классах и позволяют эффективно интегрировать их с классами пользователя. С началом выполнения программы на языке C++ автоматически открываются 4 потока:

- **cin** - консольный поток ввода (соответствует **stdin**);
- **cout** - консольный поток вывод (соответствует **stdout**);
- **cerr** - небуферизированный поток сообщений об ошибках (соответствует **stderr**);
- **clog** - буферизированный поток сообщений об ошибках.

Операции побитового сдвига в С (>>,<<) используются в C++ путем перегрузки как операции помещения данных в поток. Например, cout<<"Hello,world!\n";
Управляющие коды (esc- последовательности) используются также, как в С (см. с. 11).
Базовым потоковым классом является класс **ios**. Потомки данного класса – istream (обеспечивает ввод) и ostream (обеспечивает вывод). Класс iostream является потомком классов istream, ostream и объединяет возможности этих классов. В классе ios объявлены значения элементов перечислимого типа и методы, которые обеспечивают точную настройку вывода. В таблице 1 представлены функции-члены класса ios, поддерживающие консольный вывод в поток.

Таблица 1 Функции-члены класса ios для организации консольного вывода в поток

Функция-член	Назначение
flags	Устанавливает или читает флаги формата
setf	Устанавливает флаги формата
unsetf	Сбрасывает флаги формата
width	Устанавливает ширину вывода
fill	Устанавливает или читает символ-заполнитель
precision	Устанавливает или читает точность

Например, в программе ниже функция-член flags устанавливает флаги hex, uppercase, showbase для отображения целого числа как шестнацатиричного с использованием символов верхнего регистра и отображением системы счисления (OX). Подробную документацию и методах класса ios смотрите в документации MSDN.

```
#include<iostream>
using namespace std;
void main()
{cout.flags(ios::hex|ios::uppercase|ios::showbase);
cout<<13<<"\n"; //выводит OXD
}
```

Рисунок 1 Использование функции-члена flags

В случае, если прерывать поток вывода нежелательно, используют функции-члены, которые непосредственно действуют на поток – манипуляторы. Для работы манипуляторов необходимо подключение файла iomanip. В таблицах 2 и 3 приведены манипуляторы с параметрами и без.

Таблица 2 Манипуляторы с параметрами

Манипулятор	Назначение
setw(int nWidth=0)	Устанавливает минимальную ширину следующего поля (по умолчанию 0 – адекватная ширина вывода)
setbase(int nBase=10)	Устанавливает основание системы счисления – 8,10,16 (по умолчанию 10)
setfill(char cFill=’ ‘)	Устанавливает символ заполнитель (по умолчанию символ пробела ‘ ‘)
setprecision(int nPrec=6)	Устанавливает точность (по умолчанию 6)
setiosflags(long IFlags)	Устанавливает флаги формата потока
resetiosflags(long IFlags)	Сбрасывает флаги формата потока

Таблица 3 Манипуляторы без параметров

Манипулятор	Назначение
binary	Устанавливает двоичный режим ввода-вывода потока и файлов
text	Устанавливает текстовый режим (по умолчанию) ввода-вывода потока и файлов
dec	Форматирует целые числа как десятичные
hex	Форматирует целые числа как шестнадцатеричные
oct	Форматирует целые числа как восьмеричные
ws	Экстрактор пробельных символов
endl	Переход на новую строку
ends	Завершает строку нулевым знаком
flush	Закрывает буферы ввода-вывода

Например, в программе ниже выводится значение переменной X используя манипуляторы setfill, setprecision, setw, setiosflags и endl.

```
#include<iostream>
#include<iomanip>
using namespace std;
void main()
{       double X=999.555;
cout<<"Number is "
        <<setfill('*')//символ заполнения
        <<setprecision(3)//точность
        <<setw(25)//ширина поля вывода 25 позиций
        <<setiosflags(ios::scientific|//научный формат
```

```
                    ios::showpos|//ведущий знак +
                    ios::right)//выравнивание вправо
        <<X
        <<endl;//новая строка
}
```

Рисунок 2 Листинг программы, использующей манипуляторы

```
Number is ***************+9.996e+002
Press any key to continue
```

Рисунок 3 Результаты работы программы

Приложение Б

Пример программы по работе с классом «массив»

Программа создает объект «массив» и обеспечивает доступ к его элементам.

```
#include <iostream.h>
#include <iomanip.h>
#include <math.h>
const int MAX_ELEMS = 10;
class myArray
{  public:
   myArray(double fInitVal = 0);
   double& operator[](int nIndex)
     { return m_fArray[nIndex]; }
   void show(const char* pszMsg = "",
           const int nNumElems = MAX_ELEMS,
           const int bOneLine = 1); //Для вывода «в столбец» измените на 0
 protected:
   double m_fArray[MAX_ELEMS];};
myArray::myArray(double fInitVal)
{
 for (int i = 0; i < MAX_ELEMS; i++)
   m_fArray[i] = fInitVal;
}
void myArray::show(const char* pszMsg,
                const int nNumElems,
                const int bOneLine)
{
 cout << pszMsg << endl;
 if (bOneLine) {
   for (int i = 0; i < nNumElems; i++)
     cout << m_fArray[i] << ' ';
   cout << endl;
 }
 else {
   for (int i = 0; i < nNumElems; i++)
     cout << m_fArray[i] << endl;
   cout << endl;
```

```
  }
}
int main()
{
  double fXArr[MAX_ELEMS];
  myArray Array;
//заполнение массива fXArr
  for (int i = 0; i < MAX_ELEMS; i++)
    fXArr[i] = 2.0 + (double)i;
//определение значений объекта Array
  for (i = 0; i < MAX_ELEMS; i++)
    Array[i] = sqrt(fXArr[i]);
  cout.precision(4);
//просмотр массива
  Array.show("Initial array is:");
  cout << endl;
  // удвоение элеметов объекта Array
  for (i = 0; i < MAX_ELEMS; i++)
    Array[i] += Array[i];
  // просмотр массива
  Array.show("After doubling values, array is:");
  return 0;
}
```

Рисунок 1 Листинг программы

```
Initial array is:
1.414 1.732 2 2.236 2.449 2.646 2.828 3 3.162 3.317

After doubling values, array is:
2.828 3.464 4 4.472 4.899 5.292 5.657 6 6.325 6.633
Press any key to continue
```

Рисунок 2 Результаты работы программы