

## Лабораторная работа 2

### Одномерный массив (вектор)

**Цель.** Освоение основ технологии объектно-ориентированного программирования на примере создания класса «вектор» для описания одномерного массива.

#### 1. Общие указания

По определению одномерный массив – это последовательность элементов памяти (то есть элементы имеют соседние адреса) одного типа.

Таким образом, одномерный массив определяется тройкой:

1. Начальный адрес (логическое имя массива).
2. Тип элемента
3. Кол-во элементов данного типа (размерность).

Память класса имеет вид

```
[private: ]  
<тип> *<имя>;  
int dim; // размерность
```

#### Методы класса

1. Обязательными являются конструктор[ы], деструктор;
2. Необходимыми являются ввод, вывод элементов вектора;
3. 3 функции обработки в соответствии с вариантом задания.

#### 2. Пример реализации

**Задание 1.** Создать класс «вектор» для описания одномерного массива и вычислить сумму его элементов.

**Шаг 1.** Создать файл “vect.hpp” с описанием класса «Vect»

```
#include <stdio.h>  
class Vect {  
    int *v; // начальный адрес одномерного массива  
    int dim; // размерность  
public:  
    Vect(int); // конструктор  
    int dimen() // функция, возвращающая размерность массива  
    { return dim; }  
    void in_val_a(FILE *fin); // ввод элементов массива из файла  
    void display(FILE *fout); // вывод массива в файл  
    void show(); // вывод массива на экран  
    int sum(); // вернуть сумму элементов массива  
}; // конец описания класса
```

**Шаг 2.** Создать файл “vect.cpp” с реализацией интерфейса класса «Vect».

```
#include <iostream.h>  
#include "vect.hpp"  
Vect::Vect(int d) { // конструктор  
    if (d<=0)  
        printf("\n Size error:");  
    else {  
        dim = d;  
        v = new int[d]; //задаем размерность вектора, выделяем память  
    }  
};
```

```
}

void Vect::in_val_a(FILE *fin) {
    int i;
    for (i = 0; i < dim; i++) {
        fscanf(fin, "%d", &v[i]); //инициализация массива элементами из файла
    }
}

int Vect::sum() { // подсчет суммы элементов вектора
    int i, s = 0;
    for (i = 0; i < dim; i++) {
        s = s+v[i];
    }
    return s;
}

void Vect::display(FILE *fout) { // вывод вектора в файл
    int i;
    for (i = 0; i < dim; i++)
        fprintf(fout, "\n member i=%d equal a=%d- ", i, v[i]);
}

void Vect::show() {
    int i;
    for (i = 0; i < dim; i++)
        cout << "num" << i << "=" << v[i] << "\n";
}
```

### Шаг 3. Главная программа (файл “vect\_main.cpp”)

```
#include "vect.cpp"
void main() {
    FILE *ff,*fin;
    int i,n;
    int sx;
    ff = fopen("rez.dat", "w+");
    fin = fopen("init.dat", "r");

    Vect x(10);
    x.in_val_a(fin);
    x.display(ff);
    x.show();
    getch();
    sx = x.sum();
    fprintf(ff, "\n sum=%d", sx);
    fin = fopen("int2.dat", "r");

    Vect y(10);
    y.in_val_a(fin);
    y.display(ff);
    y.show();
    getch();

    sx = y.sum();
    fprintf(ff, "\n sum=%d", sx);
    fclose(ff);
    fclose(fin);
}
```

**Задание 2.** Создать класс «вектор» для описания одномерного массива и реализовать следующие функции его обработки:

- вычислить сумму его элементов;
- вычислить сумму элементов, имеющих четное значение;
- вычислить сумму элементов, имеющих нечетное значение;
- вычислить произведение элементов;
- вычислить произведение элементов, имеющих отрицательное значение;
- удалить элемент с индексом K.

```
// Файл "ar_ext.hpp" - описание класса
#include<stdio.h>
#include <conio.h>
class Vect {
    int *q;
    int dim;
public:
    Vect(int);
    int dmen()
        { return dim; }
    int& operator[](int);
    int& elem(int i)
        { return q[i]; }
    void in_val_a(FILE *fin);
    void display(FILE *fout);
    void show();
    int sum();
    int sum_odd();
    int sum_even();
    int product();
    int pr_neg();
    int del_el(int k);
}; // конец описания класса
```

```
// Файл "ar_ext.cpp" - реализация класса
#include "ar_ext.hpp"
Vect::Vect(int d) {
    if (d<=0)
        printf("\n Size error:");
    else {
        dim = d;
        q =new int[d];};
}

int& Vect::operator[](int i) {
    if( i < 0 || dim <= i )
        printf("Index error");
    return q[i];
}

void Vect::in_val_a(FILE *fin) {
    int i;
    for (i=0; i<dim; i++) {
        fscanf(fin,"%d",&q[i]);
    }
}

int Vect::sum() {
    int i, s = 0;
    for ( i = 0; i < dim; i++) {
        s = s + q[i];
    }
    return s;
}
```

```
    }

    int Vect::del_el(int k) { // удалить элемент с индексом k
        int i;
        for ( i = k+1; i < dim; i++) {
            q[i-1]=q[i];
        }
        dim--;
        return dim;
    }

    int Vect::sum_odd() { // сумма нечетных элементов
        int t, i, s = 0;
        for ( i = 0; i < dim; i++) {
            t = q[i] % 2;
            if (t!=1) {
                s = s + q[i];
            }
        }
        return s;
    }

    int Vect::sum_even() { // сумма четных элементов
        int t, i, s = 0;
        for ( i = 0; i < dim; i++) {
            t = q[i] % 2;
            if (t > 0) {
                s = s + q[i];
            }
        }
        return s;
    }

    int Vect::product() { // произведение всех элементов
        int i;
        int p = 1;
        for (i = 0; i < dim; i++) {
            p = p * (q[i]);
        }
        return p;
    }

    int Vect::pr_neg() { // произведение отрицательных элементов {
        int i;
        int p = 1;
        for ( i=0; i<dim; i++) {
            if (q[i] < 0)
                p = p * (q[i]);
        }
        return p;
    }

    void Vect::display(FILE *fout) { // вывод в файл
        int i;
        for ( i=0; i<dim; i++)
            fprintf(fout, "\n member i=%d equal a=%d- ", i, q[i]);
    }

    void Vect::show() {
        int i;
        for ( i=0; i<dim; i++)
            cout << "num" << i << "=" << v[i] << "\n";
    }
}
```

```
// Файл "ar2_main.cpp"
#include "ar_ext.cpp"
void main(void) {
    FILE *f1,*f2,*fr;
    int a,s;
    int Sx;
    int Px;
    fr=fopen("O.dat","w+");
    f1=fopen("A1.dat","r");

    clrscr();
    fprintf(fr,"\n VECT X:");
    Vect x(6);
    x.in_val_a(f1);
    x.display(fr);
    x.show();
    getch();

    Sx = x.sum_odd();
    Px = x.product();
    fprintf(fr,"\n sum of odd=%d,\n product=%d",Sx,Px);
    Sx = x.sum_even();
    fprintf(fr,"\n sum of even=%d",Sx);
    s = x.del_el(4);
    fprintf(fr,"\n VECT X (after deleting of 4-th element: dim=%d",x.dmen());
    x.display(fr);
    fprintf(fr,"\n VECT Y:");
    f2 = fopen("B2.dat","r");
    Vect y(5);
    y.in_val_a(f2);
    y.display(fr);
    y.show();
    getch();

    Sx = y.sum();
    Px = y.pr_neg();
    fprintf(fr,"\n sum=%d,\n product(for negatives) =%d",Sx,Px);
    fclose(f1);
    fclose(f2);
    fclose(fr);
}
```

### 3. Варианты задания

Создать класс «вектор» для описания одномерного массива и произвести с ним следующие операции, согласно варианту:

№	Задание
1.	Дан массив целых чисел $X=(x_1,x_2,...,x_n)$ . Сформировать массив $Y=(y_1,y_2,...,y_m)$ , поместив в него в порядке убывания все различные (неповторяющиеся) числа, входящие в массив $X$ .
	Определить, насколько отличаются средние арифметические значения элементов массивов $X$ и $Y$ .
	Определить наибольший общий делитель всех чисел массива.
2.	Массив $X=(x_1,x_2,...,x_n)$ содержит большое количество нулевых элементов. Определить положение и размер наиболее длинной серии таких элементов.
	И удалить ее из состава массива. (Длина массива при этом уменьшится)
	Удалить из нового массива $X=(x_1,x_2,...,x_n)$ все элементы, превышающие его среднее арифметическое значение $S$ , кроме первого такого элемента, и определить, как при этом изменилось значение $S$ . Буферный массив не использовать.
3.	Заданы два массива $X=(x_1,x_2,...,x_n)$ и $Y=(y_1,y_2,...,y_m)$ , в состав которых входят натуральные числа, причем в каждом из этих массивов нет повторяющихся элементов. Сформировать массив $Z$ , включив в него все элементы, которые одновременно содержатся в массиве $X$ и массиве $Y$ .
	Подсчитать количество неповторяющихся элементов в обоих массивах.
	Все положительные числа в массиве $Z$ переставить в обратном порядке, не изменяя положения остальных чисел. Буферный массив не использовать.

№	Задание
4.	<p>Задан целочисленный массив <math>X=(x_1, x_2, \dots, x_n)</math>, в котором могут быть одинаковые числа. Найти максимальный и минимальный элементы среди неповторяющихся чисел.</p> <p>и обменять их местами. Учесть частный случай, когда в массиве нет неповторяющихся чисел.</p> <p>Подсчитать количество повторяющихся элементов.</p>
5.	<p>Из массива целых положительных чисел <math>X=(x_1, x_2, \dots, x_n)</math> удалить все четные по значению элементы, кроме последнего,</p> <p>после чего оставшиеся числа расположить в порядке возрастания. Учесть частные случаи ( в массиве нет четных элементов, имеется только один четный элемент, все элементы - четные). Буферный массив не использовать.</p> <p>При однократном просмотре массива <math>X</math> найти два максимальных по модулю элемента, кратных соответственно числам 2 и 3, и, если такие элементы существуют и они не совпадают друг с другом, переставить их местами в массиве.</p>
6.	<p>В массиве <math>X=(x_1, x_2, \dots, x_n)</math> поменять местами первый и второй отрицательные элементы, третий и четвертый отрицательные элементы и т.д. Если количество отрицательных элементов в массиве меньше двух, преобразование массива не производить.</p> <p>Определить, как изменилось положение минимального и максимального элементов массива <math>X</math> при его преобразовании, для этого</p> <p>написать функции нахождения минимума и максимума.</p>
7.	<p>Элементы массива <math>X = (x_1, x_2, \dots, x_n)</math> - это последовательность цифр целого числа, записанного в системе счисления с основанием <math>q</math>, <math>1 &lt; q \leq 10</math>, <math>0 \leq x[i] &lt; q</math>. Переставить цифры числа в обратном порядке</p> <p>и отпечатать десятичное значение этого числа до и после перестановки.</p> <p>Найти максимальный и минимальный элементы среди неповторяющихся чисел.</p>
8.	<p>Известно, что в целочисленном массиве <math>X=(x_1, x_2, \dots, x_n)</math> три и только три числа равны между собой. Найти эти числа</p> <p>и переместить их в начало массива, сдвинув остальные числа к концу этого массива.</p> <p>найти самый длинный подмассив, который является арифметической прогрессией.</p>
9.	<p>За однократный просмотр массива найти его максимальный положительный элемент <math>X_{\max}</math></p> <p>и определить среднее арифметическое значение</p> <p>Найти НОК всех элементов массива, за исключением элементов, равных <math>X_{\max}</math>.</p> <p>Указание. В программе должны быть учтены частные случаи, в том числе:</p> <ul style="list-style-type: none"> <li>- в массиве нет положительных элементов;</li> <li>- все элементы массива положительны и равны друг другу.</li> </ul>
10.	<p>В целочисленном массиве <math>X=(x_1, x_2, \dots, x_n)</math> каждую пару <math>x[i]</math> и <math>x[j]</math> (<math>i, j=1, \dots, n</math>, <math>i &lt; j</math>) нечетных элементов преобразовать в четные элементы по формулам: <math>x[i]:=x[i]+1</math>, <math>x[j]:=x[j]-1</math>. Пары элементов <math>i, j</math> выбирать в порядке их следования в массиве <math>X</math>.</p> <p>Определить, как при этом изменились среднее арифметическое массива <math>X</math>.</p> <p>Подсчитать количество повторяющихся элементов.</p>
11.	<p>Даны два целочисленных массива <math>X=(x_1, x_2, \dots, x_n)</math> и <math>Y=(y_1, y_2, \dots, y_m)</math>. Пусть в массиве <math>X</math> имеется <math>k_1</math> четных элементов, а в массиве <math>y</math> - <math>k_2</math> нечетных элементов. Обменять местами <math>k=\min(k_1, k_2)</math> четных элементов массива <math>X</math> с нечетными элементами массива <math>Y</math> (в порядке их следования в массивах <math>X</math> и <math>Y</math>). Учесть, что в частном случае может быть <math>k=0</math>.</p> <p>Создать третий массив, в который записать элементы двух исходных массивов в порядке чет-нечет.</p> <p>Найти сумму элементов двух исходных массивов.</p>
12.	<p>Преобразовать массив <math>X</math>, расположив вначале его отрицательные, а затем неотрицательные элементы, сохранив при этом в группе отрицательных элементов их исходный относительный порядок, а в группе неотрицательных элементов изменив его на обратный.</p> <p>Определить, как при этом изменилось положение минимального по модулю элемента массива <math>X</math>. Буферный массив не использовать.</p> <p>Создать второй массив, в который записать произведения <math>i</math>-го и <math>n</math>-го, <math>(i+1)</math>-го и <math>(n-1)</math>-го и т.д. элементов.</p>
13.	<p>Заданный вещественный массив <math>X=(x_1, x_2, \dots, x_n)</math> усреднить следующим образом: максимальный и минимальный элементы заменить их средним арифметическим значением, то же сделать по отношению к максимальному и минимальному элементам преобразованного массива <math>X</math> и т.д.</p>

№	Задание
	<p><math>n/2</math> раз. Если в очередном цикле обработки массива <math>X</math> обнаружится, что его максимальный и минимальный элементы отличаются между собой не более чем на значение <math>\epsilon</math> (<math>\epsilon</math> - достаточно малая величина), то дальнейшее преобразование массива <math>X</math> не производить.</p> <p>Определить, как изменились среднее арифметическое значение <math>S</math></p> <p>И среднее квадратическое отклонение <math>G</math> элементов массива <math>X</math> после его преобразования.</p>
14.	<p>Вещественный массив <math>X=(x_1, x_2, \dots, x_n)</math> содержит несколько отрицательных элементов, разделяющих его на отдельные подмассивы. Сгруппировать элементы каждого подмассива в порядке возрастания. Учесть частные случаи ( в массиве нет отрицательных элементов; подмассив пустой или содержит только один элемент).</p> <p>Сформировать два отдельных массива для положительных и отрицательных элементов.</p> <p>Подсчитать наименьшее общее кратное для каждого из них.</p>
15.	<p>В вещественном массиве <math>X=(x_1, x_2, \dots, x_n)</math> каждую пару <math>x[i]</math> и <math>x[j]</math> (<math>i, j=1, \dots, n; i &lt; j</math>) отрицательных элементов преобразовать в положительные элементы по формуле <math>x[i], x[j] := \sqrt{x[i] * x[j]}</math>. Пары элементов <math>(i, j)</math> выбирать в порядке их следования в массиве <math>X</math>.</p> <p>Определить, как при этом изменилось среднее арифметическое значение элементов массива <math>X</math>.</p> <p>Сформировать массив <math>Y</math>, включив в него все неповторяющиеся элементы из массива <math>X</math>.</p>
16.	<p>Заданы два массива <math>X=(x_1, x_2, \dots, x_n)</math> и <math>Y=(y_1, y_2, \dots, y_m)</math>, в состав которых входят натуральные числа, причем в каждом из этих массивов нет повторяющихся элементов. Сформировать массив <math>Z</math>, объединив массивы <math>X</math> и <math>Y</math>; при этом в массиве <math>Z</math> также не должно быть повторяющихся элементов.</p> <p>Найти монотонные (т.е. либо неубывающие, либо невозрастающие) подпоследовательности в массиве <math>Z</math>.</p> <p>Упорядочить их в порядке убывания их длины.</p>
17.	<p>Заданы массивы <math>A=(a_1, a_2, \dots, a_n)</math> и <math>B=(b_1, b_2, \dots, b_n)</math>, <math>C=(c_1, c_2, \dots, c_n)</math>, элементы которых представляют собой координаты вершин треугольника на плоскости <math>(a_1, b_1, c_1)</math>-координата <math>x</math> 1-го треугольника, <math>(a_2, b_2, c_2)</math>- его координата <math>y</math>. и т.д.</p> <p>Найти в массивах отрицательные элементы, преобразовать их в их модули.</p> <p>Определить порядковые номера треугольников с максимальным и минимальным периметрами.</p>
18.	<p>В массиве <math>X=(x_1, x_2, \dots, x_n)</math> расположить в порядке убывания входящие в его состав положительные элементы, а затем в порядке возрастания - отрицательные элементы. Нулевые элементы, если они имеются в массиве <math>X</math>, расположить между группой положительных и группой отрицательных элементов.</p> <p>Определить, как при этом изменилось положение максимального и минимального элементов массива <math>X</math>. Буферный массив не использовать.</p> <p>При однократном просмотре исходного массива <math>X</math> определить значение и положение (индекс) трех наименьших элементов, после чего переставить их местами в обратном порядке.</p>
19.	<p>Заданы два целочисленных массива <math>X=(x_1, x_2, \dots, x_n)</math> и <math>Y=(y_1, y_2, \dots, y_m)</math>. В состав массива <math>X</math> дополнительно включить те элементы из массива <math>Y</math>, которые отсутствуют в массиве <math>X</math>.</p> <p>Определить, как при этом изменилось среднее арифметическое значение элементов массива <math>X</math>.</p> <p>В массиве <math>X</math> определить положение и размер последней серии положительных элементов, в состав которой входит от двух до пяти элементов, после чего переставить элементы серии в обратном порядке.</p>
20.	<p>При однократном просмотре целочисленного массива <math>X</math> найти два максимальных числа <math>X_{\max 1}</math> и <math>X_{\max 2}</math> соответственно среди четных и нечетных по значению элементов массива,</p> <p>после чего переставить в обратном порядке элементы подмассива, расположенного между этими числами, включая элементы <math>X_{\max 1}</math> и <math>X_{\max 2}</math>.</p> <p>Все четные числа в целочисленном массиве <math>X=(x_1, x_2, \dots, x_n)</math> переставить в обратном порядке, не изменяя положение остальных чисел.</p>
21.	<p>Элементы вещественного массива <math>X=(x_1, x_2, \dots, x_n)</math> строго упорядочены по возрастанию, т.е.</p> $x_1 < x_2 < x_3 < \dots < x_n,$ <p>при этом <math>x[i+1]-x[i] &gt; \epsilon</math> (<math>i = 1, \dots, n-1</math>) <math>\epsilon</math> – малое число, например, 0.001).</p> <p>Элементы вещественного массива <math>Y=(y_1, y_2, \dots, y_m)</math>, <math>m \leq n</math> расположены в произвольном порядке. Включить в состав массива <math>X</math> те элементы <math>y[j]</math>, <math>j = 1..m</math>, которые отличаются от элементов</p>

№	Задание
	<p><math>x[i]</math> не менее чем на <math>eps</math>, сохранив при этом упорядоченность массива <math>X</math>. Буферный массив не использовать.</p> <p>подсчитать количество положительных <math>k_1</math> и количество отрицательных <math>k_2</math> элементов массива <math>X</math>.</p> <p>Если <math>k = k_1 - k_2 &gt; 0</math>, то изменить знаки столько же положительных или отрицательных элементов, чтобы выполнялось условие <math>abs(k) \leq 1</math>.</p>
22	<p>Задан целочисленный массив <math>X=(x_1, x_2, \dots, x_n)</math>, в котором могут быть одинаковые числа.</p> <p>Определить есть ли повторяющиеся числа.</p> <p>Найти максимальный и минимальный элементы среди повторяющихся чисел и обменять их местами. Учесть частные случаи (в массиве нет повторяющихся чисел, максимальный и минимальный элементы равны друг другу). Буферный массив не использовать.</p>
23	<p>Выполнить циклический сдвиг массива <math>X=(x_1, x_2, \dots, x_n)</math> на <math>k</math> элементов (<math>0 \leq k \leq n</math>) по направлению, которое определяется значением переменной <math>s</math> (<math>s = 0</math> - влево, <math>s = 1</math> - вправо). Например, при циклическом сдвиге массива</p> <p>4 -8 6 12 1 0 7 9    влево на 3 элемента получим    12 1 0 7 9 4 -8 6 .</p> <p>Значения переменных <math>k</math> и <math>s</math> ввести с клавиатуры.</p> <p>Найти серии отрицательных элементов.</p> <p>Элементы каждой серии отрицательных элементов вещественного массива <math>X=(x_1, x_2, \dots, x_n)</math> переставить в обратном порядке.</p>
24	<p>Элементы массивов <math>X=(x_1, x_2, \dots, x_n)</math> и <math>Y=(y_1, y_2, \dots, y_n)</math> определяют координаты точек ломаной линии. Удалить из состава ломаной отрезок минимальной длины и отрезок максимальной длины.</p> <p>Определить, как при этом изменилась общая длина ломаной линии</p> <p>и средняя длина ее отрезков.</p>
25	<p>Значение целой части неотрицательного вещественного числа задано в виде массива двоичных цифр <math>a[n], a[n-1], a[n-2], \dots, a[1], a[0]</math>, значение его дробной части - в виде массива двоичных цифр <math>b[1], b[2], \dots, b[m]</math>.</p> <p>По отношению к этому числу выполнить следующее:</p> <ul style="list-style-type: none"> <li>– удалить незначащие нули в целой и дробной части, если они имеются;</li> <li>– округлить дробную часть до четырех двоичных цифр.</li> </ul> <p>Учесть частные случаи:</p> <ul style="list-style-type: none"> <li>– все элементы <math>a[i] = 0</math>;</li> <li>– все элементы <math>b[j] = 0</math>;</li> <li>– все элементы <math>a[i]</math> и <math>b[j]</math> равны нулю.</li> </ul> <p>Отпечатать полученное десятичное значение.</p>

#### 4. Требования к отчету

1. Обоснование выбора структуры памяти класса и его интерфейса.
2. Блок-схемы алгоритмов функций обработки (ПОЗ).
3. Описание класса + схемы (ПО2, ПО4).
4. Результаты тестирования класса.

#### 5. Контрольные вопросы

1. Пояснить назначение конструктора копий.
2. Опишите механизм вызова конструктора и деструктора.
3. Как Вы понимаете термин «перегрузка функций»?
4. Классы полезны потому, что:
  - не занимают памяти, если не используются;
  - защищают свои данные от доступа со стороны других классов;
  - собирают вместе все аспекты, касающиеся отдельной вещи;
  - адекватно моделируют объекты реального мира.