

Лабораторная работа № 4:

Передача в функции параметров по значению, по ссылке и по указателю

Оглавление

Цель	1
Инструкция:	2
Задания для самостоятельного выполнения:	4
Домашнее задание:	4
БЛОК А	5
БЛОК В	5
БЛОК С	5
Требования к оформлению программ:	5
Контрольные вопросы:	5

Цель

Научиться выполнять элементарные вычисления с массивами. Научиться передавать в функцию параметры не только по значению, но и по указателю и по ссылке.

Инструкция:

Напишем три варианта функций для вычисления скалярного произведения двух векторов.

В первой функции результат будем возвращать через параметр-значение, во второй – через параметр-ссылку и в третьей – через параметр-указатель. В четвертую функцию результирующую переменную будем передавать по значению. Проанализируем работу всех функций.

```
//-----main.cpp-----  
  
//Лабораторная работа №3  
//Параметры-ссылки и параметры-указатели в функциях  
  
#include <iostream>  
#include "vect.h"  
#include "calc_vect.h"  
#define n 3  
  
using namespace std;  
  
int main()  
{  
    double res=0;  
  
    double *v1 = new double[n];  
    double *v2 = new double[n];  
  
    InputDescVect (v1,n);  
    InputDescVect (v2,n);  
  
    cout<<"Vector v1:\t";  
    OutputDescVect (v1,n);  
  
    cout<<"Vector v2:\t";  
    OutputDescVect (v2,n);  
  
    cout<<"\n-----\n";
```

```

    res=0;
    res=ScalarProduct_1(v1,v2,n);
    cout<<"Result of ScalarProduct_1: \t"<< res<<endl;

    cout<<"\n-----\n";

    res=0;
    ScalarProduct_2(v1,v2,n,res);
    cout<<"Result of ScalarProduct_2: \t"<< res<<endl;

    cout<<"\n-----\n";

    res=0;
    ScalarProduct_3(v1,v2,n,&res);
    cout<<"Result of ScalarProduct_3: \t"<< res<<endl;

    cout<<"\n-----\n";

    res=0;
    ScalarProduct_4(v1,v2,n,res);
    cout<<"Result of ScalarProduct_4: \t"<< res<<endl;

    cout<<"\n-----\n";

    delete []v1; v1=NULL;
    delete []v2; v2=NULL;
}

//-----calc_vect.h-----

#ifndef CALC_VECT_H_INCLUDED
#define CALC_VECT_H_INCLUDED

double ScalarProduct_1(double* v1, double* v2,int n);
void ScalarProduct_2(double* v1, double* v2, int n, double& res);
void ScalarProduct_3(double* v1, double* v2, int n, double* res);
void ScalarProduct_4(double* v1, double* v2, int n, double res);

#endif // CALC_VECT_H_INCLUDED

```

```

//-----calc_vect.cpp-----

#include "vect.h"
#include "calc_vect.h"

double ScalarProduct_1(double* v1, double* v2, int n)
{
    double res=0;

    for(int i=0;i<n;i++)
        res+=v1[i]*v2[i];
    return res;
}

void ScalarProduct_2(double* v1, double* v2, int n, double& res)
{
    res=0;

    for(int i=0;i<n;i++)
        res+=v1[i]*v2[i];
}

void ScalarProduct_3(double* v1, double* v2, int n, double* res)
{
    *res=0;

    for(int i=0;i<n;i++)
        (*res)+=v1[i]*v2[i];
}

void ScalarProduct_4(double* v1, double* v2, int n, double res)
{
    res=0;

    for(int i=0;i<n;i++)
        res+=v1[i]*v2[i];
}

```

Задания для самостоятельного выполнения:

1. Написать четыре функции для вычисления скалярного произведения двух векторов размерности **n** (n считывать с экрана и выполнять проверку корректности ввода):
 - 1.1. Функция, возвращающая результат через параметр-значение.
 - 1.2. Функция, в которую приходит результирующая переменная по ссылке
 - 1.3. Функция, в которую приходит результирующая переменная по указателю
 - 1.4. Функция, в которую приходит результирующая переменная по значению

Проанализировать полученный результат.

Домашнее задание:

Выполнить задание и оформить код программы в соответствии с требованиями.

БЛОК А

Задание на отметку «удовлетворительно»

Использовать «заготовку» проекта из лабораторного задания №3.

Добавить функции:

1. Для вычисления векторного произведения двух векторов размерности 3.
2. Для вычисления смешанного произведения векторов размерности 3. Результат возвращать через параметр-указатель.
3. Для вычисления длины вектора произвольного размера **n**. Результат возвращать через параметр-значение.
4. Для вычисления косинуса угла между двумя векторами и нахождения этого угла. Косинус угла возвращать через параметр-значение, а угол – через параметр-ссылку.

Продемонстрировать работу всех функций.

БЛОК В

Задания на отметку «хорошо»

Выполнить задание из *Лабораторной 1* (Блок В), используя параметры-указатели и параметры-ссылки.

БЛОК С

Задания на отметку «отлично»

Выполнить задание из *Лабораторной 3* (Блок С) и внести следующие изменения:

1. Написать функцию для определения размерности массива. Вызывать ее перед выделением памяти. В функции использовать параметры-ссылки.
2. Написать функцию для выделения памяти под одномерный массив и вызывать ее в главной программе.

Требования к оформлению программ:

1. **Содержание.** Программа должна делать то, что предусмотрено заданием. Не надо выполнять лишних действий, заданием не предусмотренных.
2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
 - Кто выполнил.
 - Что делает программа (кратко).
 - Что на входе (имена входных файлов и т.д.).
 - Что на выходе (что является результатов работы программы?).
3. **Ввод и вывод**
 - Приглашения к вводу (например, сколько чисел, какого типа и через какой разделитель нужно вводить).

- Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)
 - «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.
4. **Структура кода.** Набираемый код должен быть хорошо структурированным. Использовать:
- Отступы.
 - Комментарии – поясняют решение программы.
 - Осмысленные названия переменных.
 - Пояснения о назначении переменных в комментариях (кроме счетчиков).
5. **Декомпозиция кода**
- Функциональная. Программу оформлять с помощью функций.
6. **Многофайловые проекты**
- Все проекты должны состоять минимум из двух модулей: главного и подключаемого.

Главный модуль **main.cpp**. В нем оставить только функцию `main`, в которой вызывать функции, описанные в других модулях.

Модуль с описанием пользовательских функций **file.cpp** и **file.h**. В этот модуль перенести определение всех функций, необходимых для выполнения задания.

Контрольные вопросы:

1. Напишите пример функции для вычисления суммы двух чисел. Результат возвращать через параметр-ссылку. Написать пример вызова этой функции. Выписать прототип функции.
2. Напишите пример функции для вычисления суммы двух чисел. Результат возвращать через параметр-указатель. Написать пример вызова этой функции. Выписать прототип функции.
3. Напишите пример функции для вычисления суммы двух чисел. Результат возвращать через параметр-значение. Написать пример вызова этой функции. Выписать прототип функции.