МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Лабораторная работа №3

**«Разработка консольных проектов Visual Studio   
с использованием функций VC»**

по теме

**«Функции VС++ и консольные проекты Visual Studio»**

**по дисциплине**

**«Введение в Информационные Технологии»**

Выполнил: студент гр. БИБ2305 Пуховский С.Д.

Вариант №18

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_.

Москва, 2023 г

**1) Общее задание на разработку программного проекта**

1. Изучите структуру программного кода консольных проектов Visual Studio и правила определения, объявления и вызова функций VC++.
2. Выберите индивидуальный вариант задания из таблицы 3.1.
3. Решите задачу вычисления заданного арифметического выражения с использованием функций VC++ (без использования функций она уже решена в предыдущей работе 4).

Для этого разработайте три варианта схем алгоритмов и соответствующих функций, реализующих решения задачи:

* схемы алгоритмов для вычисления заданного арифметического выражения:
* схему алгоритма процедуры с входными параметрами и возвращаемым значением;
* схему алгоритма процедуры с входными и выходными параметрами и без возвращаемого значения;
* схему алгоритма без параметров и без возвращаемого значения;
* программные коды трех функций и функции **main** в соответствии со схемами алгоритмов.

1. Создайте консольное решение, содержащее пять проектов, каждый из которых содержит одну из разработанных функций п.3 и главную функцию **main**, в которой осуществляется ввод исходных данных, вызов соответствующей функции п.3 и вывод результата:
2. Функция с параметрами и возвращаемым значением, причем определение функции должно быть записано перед функцией **main**.
3. Функция с параметрами и возвращаемым значением, причем определение функции должно быть записано после функции **main**.
4. Функция с параметрами и без возвращаемого значения.
5. Функция без параметров и без возвращаемого значения (с глобальными переменными).
6. Функция с параметрами и возвращаемым значением, причем определения функции и **main** должны находится в разных файлах.

Каждый способ должен быть реализован в отдельном проекте, а все пять проектов должны быть объединены в одном решении.

1. Выполните созданные проекты и получите результаты.
2. Докажите правильность результатов.
3. Выполните созданные проекты по шагам, с помощью отладчика, установив точку останова после ввода данных.
4. Ответьте на вопросы, поставленные в примере выполнения задания. Внесение изменений в программный код выполняйте путем комментирования исходного кода с последующим удалением комментария для возврата к исходному состоянию.
5. **Общее и индивидуальное задания на разработку программного проекта:**

Создать решение (например, с именем ***lab5*)**, состоящее из пяти программных проектов, для вычисления арифметического выражения:

при значениях исходных данныхx=8.8 и y=5.8 с использованием раз-  
личных способов обмена данными и местоположением функций в соответствии с общим заданием.

1. **Формализация и уточнение задания:**

Для формализации и уточнения задания определим, что исходные данные **x**, **y** – вещественного типа **double**. Результаты вычислений – переменная **z** также должна быть вещественного типа **double**. Операция вычисления **z** будет записываться следующим оператором VC++:

|  |
| --- |
| *double* buf\_1 = (3 \* *y*) - *x*;  *double* buf\_2 = *x* \* sin(*y* \* *y*);  *z* = (M\_PI/3) + log(pow(*x*, 3));  *z* = (*z*/buf\_1) + buf\_2; |

Вычисление **z** реализуем в функциях VC++ тремя различными способами в соответствии с общим заданием.

1. **Разработка пяти программных проектов в одном решении и получение  
    результатов их работы**.

**Реализация 1-го проекта:**

Алгоритм главной процедуры не зависит от способа обмена данными и приведен на рисунке 5.2.

|  |
| --- |
|  |

Рисунок 5.2 – Схема алгоритма главной процедуры **main** для всех проектов

* ***Разработаем алгоритм процедуры с параметрами и возвращаемым значением****.*

Схема алгоритма этой процедуры **func1** представлена на рисунке 5.3.

|  |
| --- |
|  |

Рисунок 5.3 – Схема алгоритма процедуры **func1** с параметрами и возвращаемым значением для первого проекта

* ***Разработаем программные коды двух функций в соответствии со схемами алгоритмов.***

Программные коды разработанных функций запишем в файл с именем **zad1.cpp** в следующем порядке (рисунок 5.4):

* сначала определение функции **func1** с параметрами и возвращаемым значением;
* после него главную функцию **main.**

|  |
| --- |
| *double* func1(*double* *x*, *double* *y*)  {  *double* z;  *double* buf\_1 = (3 \* *y*) - *x*;  *double* buf\_2 = *x* \* sin(*y* \* *y*);      z = (M\_PI/3) + log(pow(*x*, 3));      z = (z/buf\_1) + buf\_2;      return z;  }  *int* main()  {  *double* x, y, z;     cout<<" Input x = ";     cin>>x;     cout<<endl<<"Input y = ";     cin>>y;     z = func1(x, y);     cout<<endl<<"Result z = "<<z<<endl;     return 0;  } |

Рисунок 5.4 –Программный код первого проекта

* ***Откомпилируем файл*** **zad1.cpp**, ***выполним построение*** решения **lab5** и ***выполнение*** проекта **pr1**. Получим следующие результаты при заданных значениях исходных данных (рисунок 5.5).

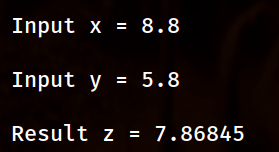


Рисунок 5.5 –Результаты выполнения проекта **pr1**

**Реализация 2-го проекта:**

* ***Создадим******второй пустой проект*** с именем **pr2** в уже существующем решении. Для этого выполним команду ***Файл/Добавить/Создать проект***. В открывшемся окне ***Добавить новый проект*** зададим имя проекта **pr2.**
* ***Запишем в файл*** с именем **zad2.cpp** программные коды разработанных в первом проекте функций (переименовав функцию **func1** в **func2)** в следующем порядке:
* сначала программный код главной функции **main**;
* после него определение функции **func2** с параметрами и возвращаемым значением.

Перед кодом главной функции запишем прототип функции **func2**.

* ***Откомпилируем файл*** **zad2.cpp** и ***выполним перестроение*** решения **lab5**.
* Чтобы выполнить второй проект, его надо назначить запускаемым   
   проектом. Для этого необходимо выделить в окне ***Обозреватель   
   решений*** имя проекта **pr2** и нажать правую кнопку мыши, а затем   
   выполнить команду ***Назначить запускаемым проектом***
* ***Выполним*** проект **pr2**. Получим следующие результаты при заданных значениях исходных данных (рисунок 5.7).

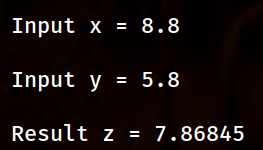


Рисунок 5.7 –Результаты выполнения проекта **pr2**

**Реализация 3-го проекта:**

* ***Создадим третий проект с именем*** **pr3** в уже имеющемся решении.
* ***Разработаем алгоритм процедуры с параметрами и без возвращаемого значения****.*

Схема алгоритма этой процедуры **func3** представлена на рисунке 5.8.

|  |
| --- |
|  |

Рисунок 5.8 – Схема алгоритма процедуры **func3** с параметрами и без возвращаемого значения для третьего проекта

* + ***Разработаем программные коды двух функций в соответствии со схемами алгоритмов***.

Программные коды разработанных функций запишем в файл с именем **zad3.cpp** в следующем порядке (рисунок 5.9):

* объявление **(**прототип**)** функции **func3**;
* определение функции **main;**
* определение функции **func3** с входными и выходными параметрами без возвращаемого значения, где входные данные передаются по значению, а результат вычислений возвращается через параметр по ссылке.

|  |
| --- |
| #include <iostream>  #include <cmath>  using *namespace* std;  *void* func3(*double* *x*, *double* *y*, *double*& *z*);  *int* main()  {  *double* x,y,z;      cout<<" Input x = ";      cin>>x;      cout<<endl<<"Input y = ";      cin>>y;        func3(x,y,z);        cout<<endl<<"Result z = "<<z<<endl;      return 0;  }  *void* func3 (*double* *x*, *double* *y*, *double*& *z*)  {  *double* buf\_1 = (3 \* *y*) - *x*;  *double* buf\_2 = *x* \* sin(*y* \* *y*);  *z* = (M\_PI/3) + log(pow(*x*, 3));  *z* = (*z*/buf\_1) + buf\_2;  } |

Рисунок 5.9 –Программный код третьего проекта.

* ***Откомпилируем*** ***файл*** **zad3.cpp**, ***выполним перестроение*** решения **lab5** и ***выполнение*** проекта **pr3**, назначив его запускаемым проектом. Получим следующие результаты при заданных значениях исходных данных (рисунок 5.10).

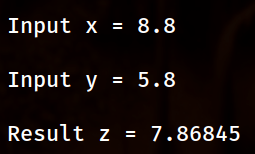


Рисунок 5.10 –Результаты выполнения проекта **pr3**

**Реализация 4-го проекта:**

* ***Создадим четвертый проект с именем*** **pr4** в уже имеющемся решении.
* ***Разработаем алгоритм процедуры без параметров и без возвращаемого значения****.*

Схема алгоритма этой процедуры **func4** представлена на рисунке 5.11.

|  |
| --- |
|  |

Рисунок 5.11 – Схема алгоритма процедуры **func4** без параметров и без возвращаемого значения для четвертого проекта

* + ***Разработаем программные коды двух функций в соответствии со схемами алгоритмов***.

Программные коды разработанных функций запишем в файл с именем **zad4.cpp** в следующем порядке (рисунок 5.12):

* объявление **(**прототип**)** функции **func4**;
* определение глобальных переменных;
* определение функции **main**;
* определение функции **func4** без параметров и без возвращаемого значения.

|  |
| --- |
| #include <iostream>  #include <cmath>  using *namespace* std;  *void* func4(*void* );  *double* x, y, z;  *int* main()  {      cout<<"Input x = ";      cin>>x;      cout<<endl<<"Input y = ";      cin>>y;      func4(); // Вызов функции      cout<<endl<<"Result z = "<<z<<endl;      return 0;  }  *void* func4()  {  *double* buf\_1 = (3 \* y) - x;  *double* buf\_2 = x \* sin(y \* y);      z = (M\_PI/3) + log(pow(x, 3));      z = (z/buf\_1) + buf\_2;  } |

Рисунок 5.12 –Программный код четвертого проекта

* ***Откомпилируем*** ***файл*** **zad4.cpp**, ***выполним перестроение*** решения **lab5** и ***выполнение*** проекта **pr4**, назначив его запускаемым проектом. Получим следующие результаты при заданных значениях исходных данных (рисунок 5.13).

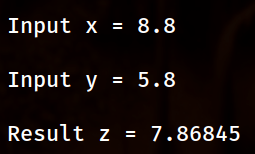


Рисунок 5.13 –Результаты выполнения проекта **pr4**

**Реализация 5-го проекта:**

Создадим в уже имеющемся решении пятый проект с именем **pr5**, состоящий из двух файлов. В первый файл с именем **zad5\_m.срр**поместим текст функции **main** из второго проекта.Во второй файл с именем **zad5\_f.срр** поместим текст функции **func2** из того же проекта.

После раздельной компиляции файлов выполним их совместную компоновку (перестроение решения) и выполнение проекта. Результаты выполнения при заданных значениях исходных данных приведены на рисунке 5.14.

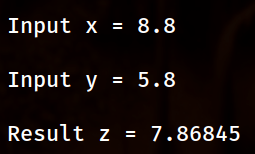


Рисунок 5.14 –Результаты выполнения проекта **pr5**

**6) Доказательство правильности результатов.**

Результат выполнения всех пяти проектов одинаков и равен **7.86845**. Выполним расчет арифметического выражения с использованием калькулятора или программы Microsoft Excel и получим совпадающий результат, что доказывает его правильность.