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»

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

по дисциплине

«Разработка безопасного программного обеспечения»

на тему

«Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Объявление и определение переменных, функций и пространств имён. Линейные алгоритмы»

Выполнил:

студент группы БПЗ1901

Неживлева К.И.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# Цель работы

Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2015 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.

# Задание

Индивидуальный вариант:

#### Задание 1

Составить программу, имеющую линейный алгоритм и состоящую из одной функции void main(). Программу записать в файл с именем task1.cpp. Скомпилировать, скомпоновать и выполнить.

В функции main организовать вычисление задачи вашего варианта дважды:

* для исходных данных, значения которых задать в виде литералов в тексте функции main;
* для исходных произвольных данных, значения которых пользователь Вашей программы должен ввести с клавиатуры в процессе выполнения программы.

Проанализировать результаты работы и сделать выводы. Перенести полученные результаты в отчет.

Вывод действительных чисел осуществлять с точностью до 0.0001

|  |
| --- |
| Task1.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  int main()  {  double x = 10;    double f = (x\*x+2\*x-3+(x+1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  cout << "x = " << x << endl;  cout << "f = " << setprecision(4) << f << endl;  cout << "x = ";  cin >> x;  f = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  cout << "f = " << setprecision(4) << f << endl;  } |

#### 

Рис.1 - результат работы программы

#### Задание 2

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через **параметры функции по значению**.

Выходные данные необходимо возвращать через **возвращаемое значение функции**.

Прототип функции:

double f(double x) // Если функция имеет один параметр

double f(double x, double y) // Если функция имеет два параметра

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций файл с именем task2.сpp в следующем порядке: функция f, функция main.

Cкомпилировать, скомпоновать и выполнить.

|  |
| --- |
| Task2.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double f(double x) {  return (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  }  int main()  {  double x = 10;  cout << "x = " << x << endl;  cout << "f = " << setprecision(4) << f(x) << endl;  cout << "x = ";  cin >> x;  cout << "f = " << setprecision(4) << f(x) << endl;  } |

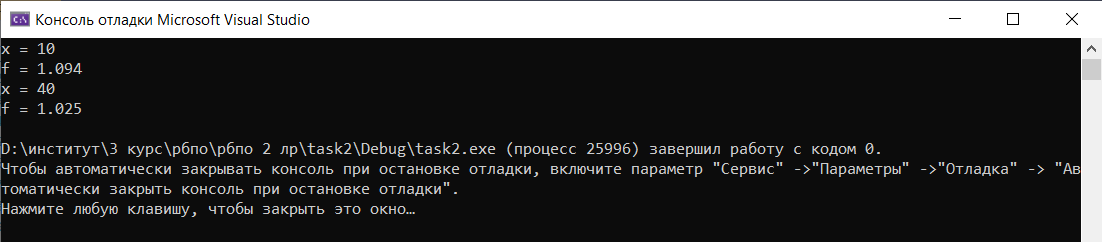


Рис.2 - результат работы программы

#### Задание 3

Создать файл task3.сpp, в котором изменить порядок записи текстов функций, созданных в задании 2. Функции записать в следующем порядке: функция main, функция f.

Внести требуемые дополнения, добиться успешной компиляции, скомпоновать и выполнить.

|  |
| --- |
| Task3.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double f(double);  int main()  {  double x = 10;  cout << "x = " << x << endl;  cout << "f = " << setprecision(4) << f(x) << endl;  cout << "x = ";  cin >> x;  cout << "f = " << setprecision(4) << f(x) << endl;  }  double f(double x) {  return (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

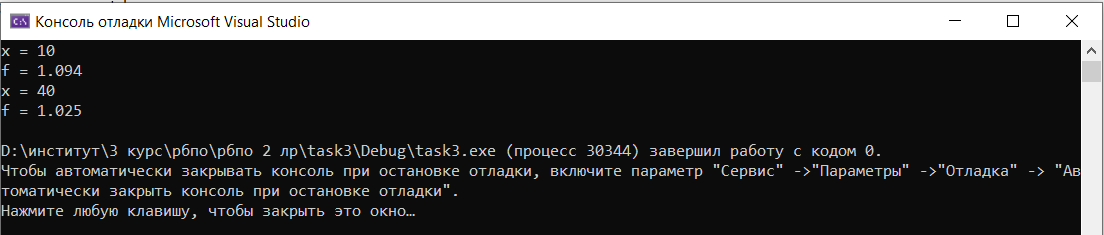


Рис.3 - результат работы программы

#### Задание 4

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через **параметры функции по ссылке на константный объект**.

Выходные данные необходимо возвращать через **дополнительный параметр функции по ссылке**. Функция не имеет возвращаемого значения.

Прототип функции:

void f(const double& x, double& result) // или

void f(const double& x, const double& y, double& result)

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций в файл с именем task4.сpp в следующем порядке: функция main, затем функция f. Cкомпилировать, скомпоновать и выполнить.

|  |
| --- |
| Task4.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  void f(const double&, double&);  int main()  {  double x = 10, result;  cout << "x = " << x << endl;  f(x, result);  cout << "f = " << setprecision(4) << result << endl;  cout << "x = ";  cin >> x;  f(x, result);  cout << "f = " << setprecision(4) << result << endl;  }  void f(const double& x, double& result) {  result = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

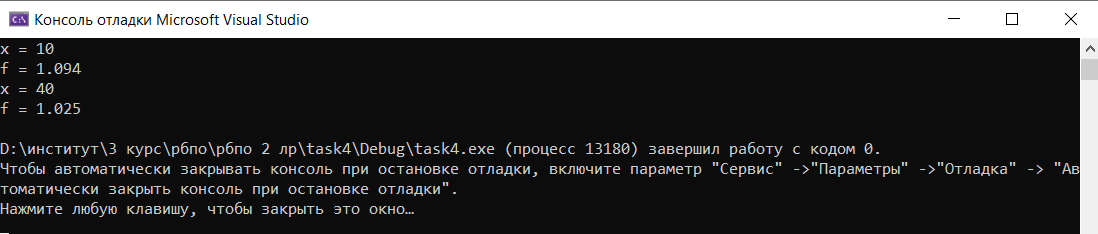


Рис.4 - результат работы программы

#### Задание 5

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через **глобальные объекты x и y**.

Выходные данные необходимо возвращать через **глобальный объект result**. Функция не имеет возвращаемого значения.

Прототип функции:

void f();

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций в файл с именем task5.сpp в следующем порядке: функция main, затем функция f. Cкомпилировать, скомпоновать и выполнить.

|  |
| --- |
| Task5.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double x, result;  void f();  int main()  {  x = 10;  cout << "x = " << x << endl;  f();  cout << "f = " << setprecision(4) << result << endl;  cout << "x = ";  cin >> x;  f();  cout << "f = " << setprecision(4) << result << endl;  }  void f() {  result = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

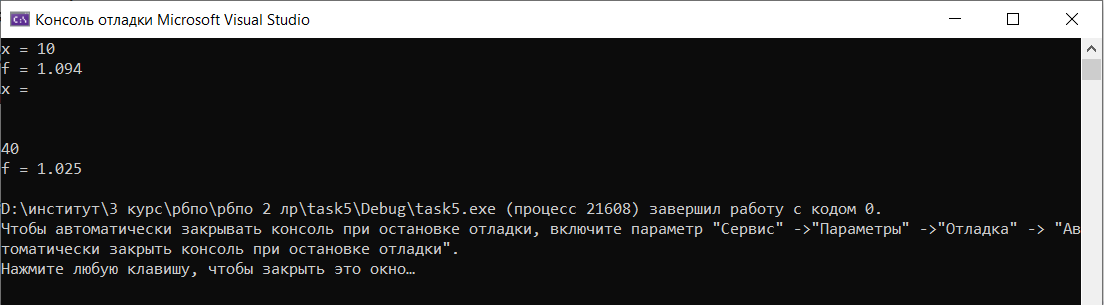


Рис.5 - результат работы программы

#### Задание 6

В этом задании необходимо разделить текст файла task2.сpp на два файла. В первый файл с именем task6\_main.cpp поместить текст функции main. Скомпилировать только файл task6\_main.cpp.

Во второй файл с именем task6\_func.сpp поместить текст функции f. Скомпилировать только файл task6\_func.cpp.

После раздельной компиляции осуществить совместную компоновку. Полученный исполняемый файл выполнить.

Проанализировать результаты работы и сделать выводы.

|  |
| --- |
| Task6\_main.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double f(double x);  int main()  {  double x = 10;    cout << "x = " << x << endl;  cout << "f = " << setprecision(4) << f(x) << endl;  cout << "x = ";  cin >> x;  cout << "f = " << setprecision(4) << f(x) << endl;  } |

|  |
| --- |
| Task6\_func.сpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double f(double x) {  return (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

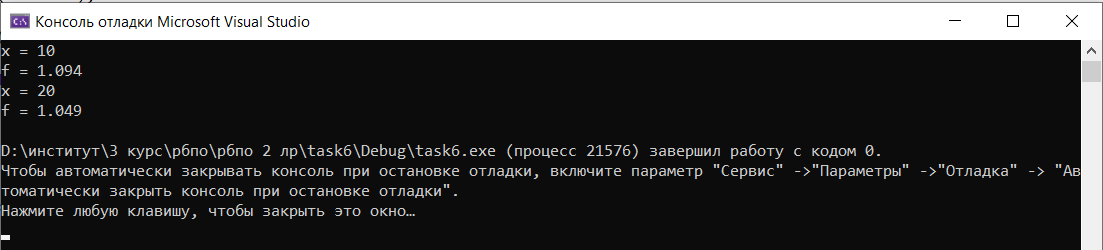


Рис.6 - результат работы программы

#### Задание 7

В этом задании необходимо разделить текст файла task5.сpp на два файла.

В первый файл с именем task7\_main.cpp поместить текст функции main. Скомпилировать только файл task7\_main.cpp.

Во второй файл с именем task7\_func.сpp поместить определения глобальных объектов x, y и result и текст функции f.

Скомпилировать только файл task7\_func.cpp.

После раздельной компиляции осуществить совместную компоновку. Разобраться в проблемах, возникающих при совместной компиляции и компоновки. Полученный исполняемый файл выполнить.

Проанализировать результаты работы и сделать выводы.

|  |
| --- |
| task7\_main.cpp |
| #include <iostream>  #include <iomanip>  using namespace std;  extern double x;  extern double result;  void f();  int main()  {  x = 10;  cout << "x = " << x << endl;  f();  cout << "f = " << setprecision(4) << result << endl;  cout << "x = ";  cin >> x;  f();  cout << "f = " << setprecision(4) << result << endl;  } |

|  |
| --- |
| task7\_func.сpp |
| #include <iostream>  #include <iomanip>  using namespace std;  double x, result;  void f() {  result = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

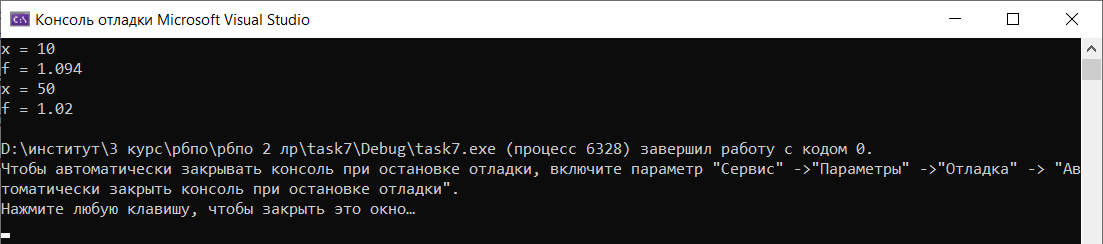


Рис.7 - результат работы программы

#### Задание 8

В этом задании необходимо модифицировать тексты файлов из задания 7.

Описание функции и внешних переменных выделить в отдельный заголовочный файл func.h, включить его содержимое в файлы task8\_main.cpp и task8\_func.cpp. Определение внешних переменных произвести в файле task8\_func.cpp.

Скомпилировать по отдельности файлы task8\_main.cpp и task8\_func.cpp. Произвести компоновку, выполнить полученный исполняемый файл.

|  |
| --- |
| func.h |
| #pragma once  extern double x, result;  void f(); |

|  |
| --- |
| task8\_func.cpp. |
| #include <iostream>  #include "func.h"  using namespace std;  double x = 10;  double result = 0;  void f() {  result = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  } |

|  |
| --- |
| task8\_main.cpp |
| #include <iostream>  #include <iomanip>  #include "func.h";  using namespace std;  int main()  {  cout << "x = " << x << endl;  f();  cout << "f = " << setprecision(4) << result << endl;  cout << "x = ";  cin >> x;  f();  cout << "f = " << setprecision(4) << result << endl;  } |

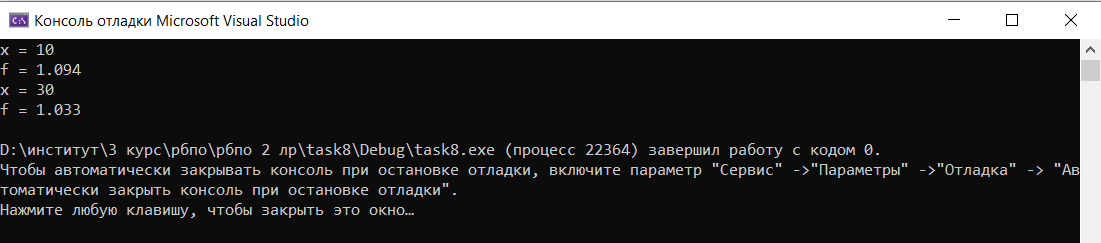


Рис.8 - результат работы программы

#### Задание 9

В этом задании необходимо модифицировать тексты файлов из задания 8.

Перенести все объявления и определения функции f в пространство имен, содержащее Вашу фамилию.

Скомпилировать по отдельности файлы task9\_main.cpp и task9\_func.cpp. Произвести компоновку, выполнить полученный исполняемый файл.

|  |
| --- |
| func.h |
| #pragma once  extern double x, result;  namespace Nezhivleva {  void f();  } |

|  |
| --- |
| task9\_main.cpp |
| #include <iostream>  #include <iomanip>  #include "func.h";  using namespace std;  int main()  {  cout << "x = " << x << endl;  Nezhivleva::f();  cout << "f = " << setprecision(4) << result << endl;  cout << "x = ";  cin >> x;  Nezhivleva::f();  cout << "f = " << setprecision(4) << result << endl;  } |

|  |
| --- |
| task9\_func.cpp |
| #include <iostream>  #include "func.h"  double x = 10;  double result = 0;  namespace Nezhivleva {  void f() {  result = (x \* x + 2 \* x - 3 + (x + 1) \* sqrt(x \* x - 9)) / (x \* x + 2 \* x - 3 + (x - 1) \* sqrt(x \* x - 9));  }  } |

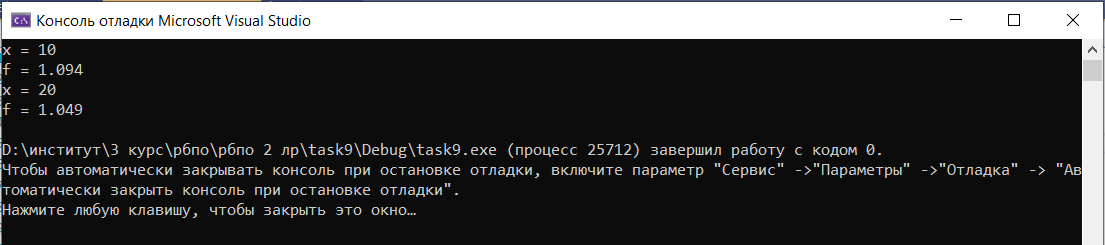


Рис.9 - результат работы программы

#### Задание 10

В этом задании необходимо собрать статическую библиотеку task10\_lib.lib. В библиотеку включить объектный файл task9\_func.obj из задания 9.

Произвести компоновку объектного файла task9\_main.obj из задания 9 и полученную статическую библиотеку task10\_lib.lib. Выполнить полученный исполняемый файл.

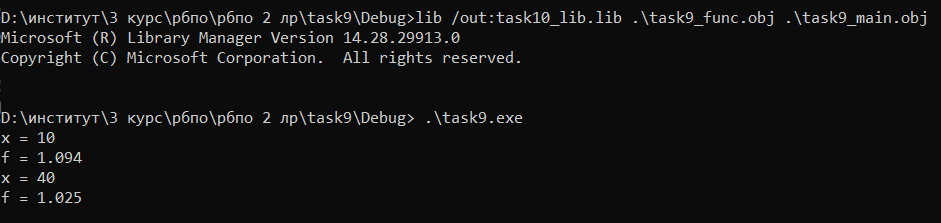


Рис.10 - результат работы программы

# Выводы

Мы овладели навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2015 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.