**Лабораторная работа 1 (2 часа)**

**Конструирование программного обеспечения Этапы обработки исходного кода. Командная строка разработчика. Проект, содержащий несколько cpp-файлов**

Задание 1

#include <iostream>

#include <locale>

using namespace std;

//проверка на корректность

bool correctness(int day, int mounth, int year) {

if (year < 0) {

return false;

}

if (mounth < 1 || mounth > 12) {

return false;

}

int Inmonth[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

int maxDay = Inmonth[mounth - 1];

if (day < 1 || day > maxDay) {

return false;

}

return true;

}

//является, ли год високосным

bool LeapYear(int year) {

return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}

//вычисление порядкового номера дня в веденном году

int Day\_that\_was\_entered(int day, int mounth, int year) {

int o = 0;

int Inmonth[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

for (int i = 1; i < mounth; i++) {

o += Inmonth[i - 1];

}

if (LeapYear(year) && mounth > 2) {

o++;

}

o += day;

return o;

}

int UBirthday(int cd, int cm, int cy, int bd, int bm) {

int co = Day\_that\_was\_entered(cd, cm, cy);

int bo = Day\_that\_was\_entered(bd, bm, cy);

if (bo > co) {

bo += LeapYear(cy + 1) ? 366 : 365;

}

return co - bo;

}

int main() {

setlocale(LC\_ALL, "Russian");

int day, mounth, year;

cout << "Введите дату в формате ДДММГГГГ: ";

cin >> day >> mounth >> year;

if (!correctness(day, mounth, year)) {

cout << "Некорректная дата!" << endl;

return 1;

}

cout << "Введенный год " << (LeapYear(year) ? "високосный" : "не високосный") << endl;

int o = Day\_that\_was\_entered(day, mounth, year);

cout << "Порядковый номер введенного дня в году: " << o << endl;

int cd = 25; // Введите текущий день вашего рождения

int cm = 5; // Введите текущий месяц вашего рождения

int cy = 2024; // Введите текущий год вашего рождения

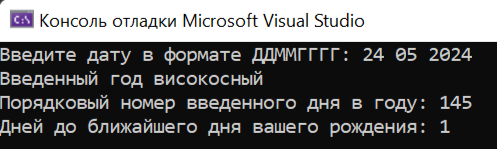
int du = UBirthday(cd, cm, cy, day, mounth);

cout << "Дней до ближайшего дня вашего рождения: " << du << endl;

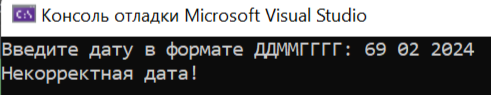
return 0;

}

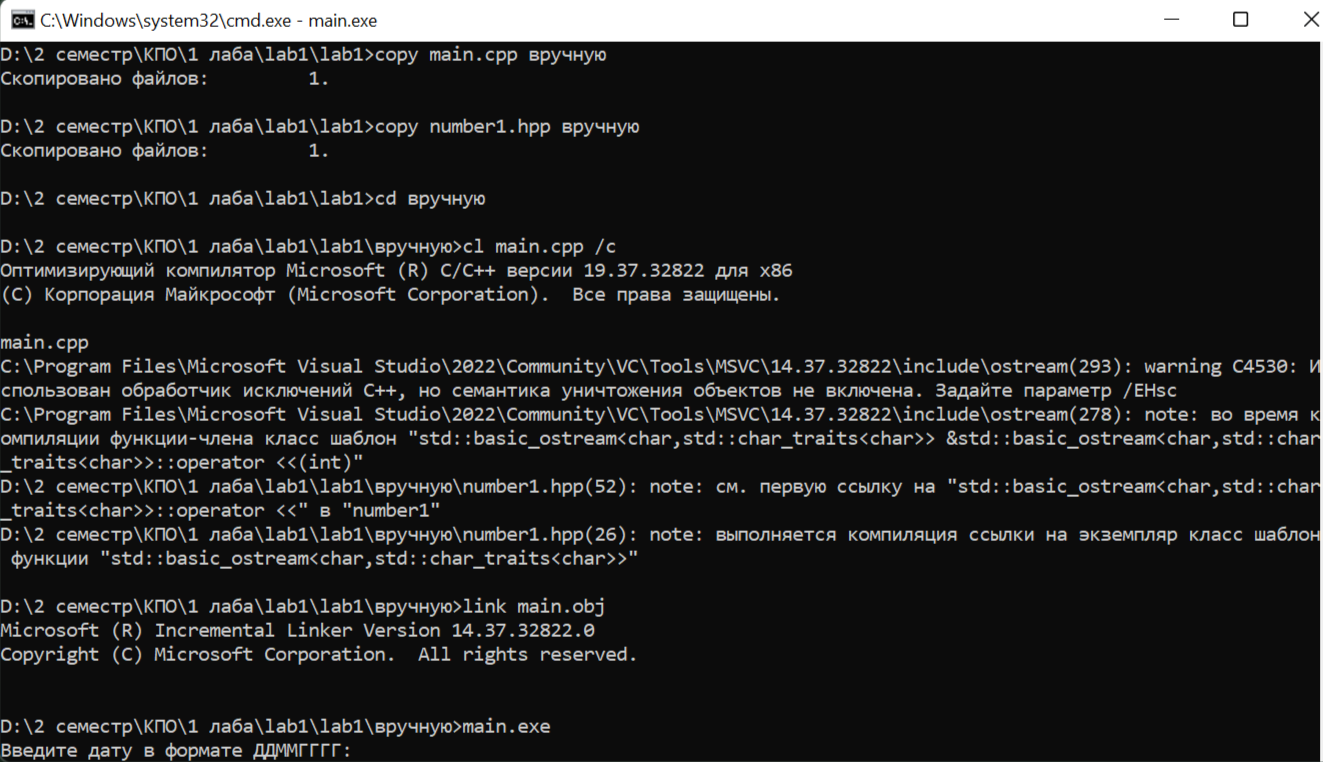
Успешное завершение программы



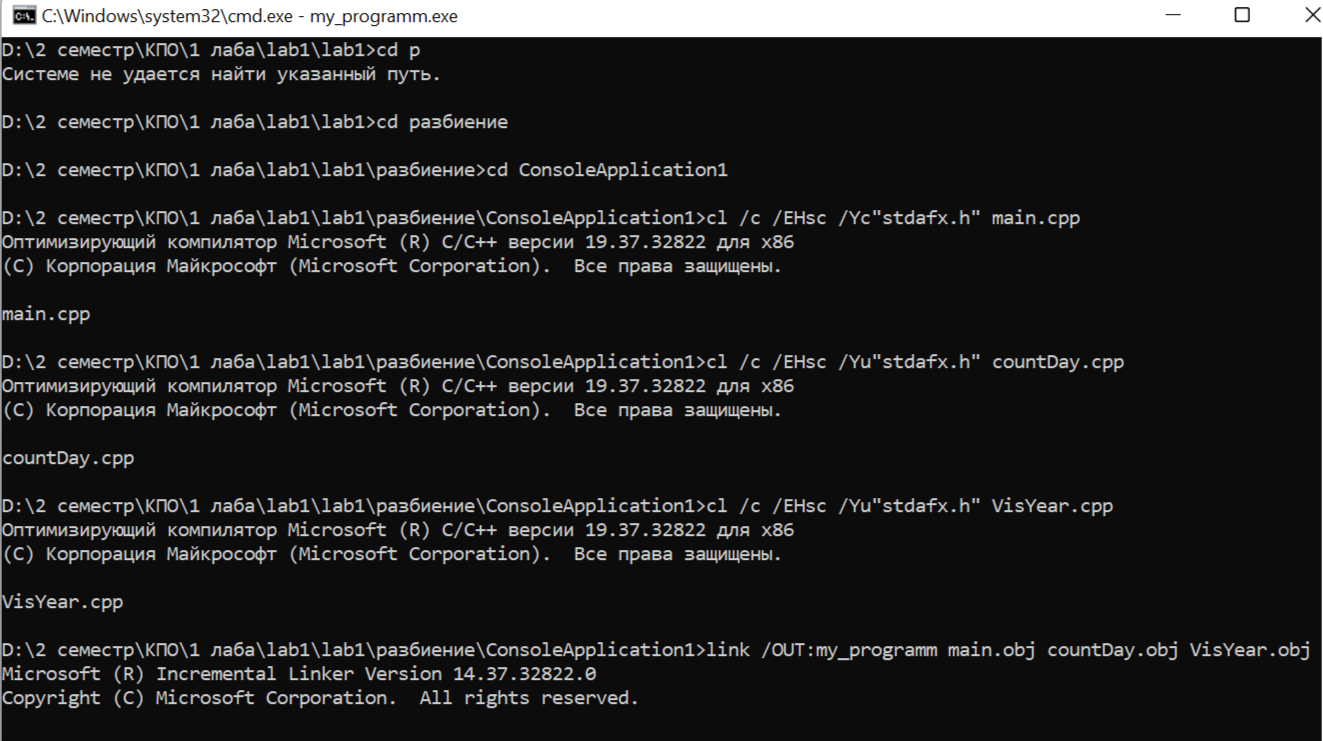
Завершение программы с негативным исходом



Задание 2



Задание 3



**Вопросы**

1. Расширения исходных файлов проекта C++:

- `.cpp` - файлы исходного кода на языке C++.

- `.h` или `.hpp` - заголовочные файлы C++.

2. Этапы обработки исходного кода программы:

- Препроцессинг (Preprocessing).

- Компиляция (Compilation).

- Линковка (Linking).

3. Файлы с расширением `.obj` образуются в результате этапа компиляции. В этих файлах содержится объектный код, сгенерированный компилятором из исходного кода программы.

4. Файл с расширением `.exe` образуется в результате этапа линковки. Это исполняемый файл, который содержит скомпонованный исходный код программы, готовый к выполнению.

5. Предполагается, что вы используете предкомпилированный заголовочный файл `stdafx.h`. Файлы, созданные после каждого вызова команды `cl`:

- После компиляции с опцией `/Yc` для предкомпилирования заголовочного файла `stdafx.h` будет создан файл `stdafx.obj`, содержащий предкомпилированный код из `stdafx.h`.

- После компиляции остальных файлов с использованием результата предкомпиляции (`/Yu`) будут созданы соответствующие файлы `.obj` для каждого исходного файла C++.

6. Файлы, созданные в результате выполнения команды `link`, обычно включают:

- Исполняемый файл (`.exe`), который является результатом сборки программы.

- Вспомогательные файлы, такие как `.pdb` (файлы информации о отладке) и другие.

7. Прототип функции - это объявление функции, которое содержит информацию о ее сигнатуре (возвращаемый тип, имя функции и параметры), но не содержит тела функции. Назначение прототипа функций состоит в предоставлении компилятору информации о том, что функция существует и какие параметры она принимает, чтобы компилятор мог правильно интерпретировать вызовы этой функции до того, как будет доступно ее определение.

8. Для выполнения компиляции в командной строке разработчика используйте команду `cl`, указав исходные файлы и необходимые опции компилятора, например:

```

cl /Ycstdafx.h main.cpp

```

9. Для выполнения компоновки в командной строке разработчика используйте команду `link`, указав объектные файлы и необходимые опции линковщика, например:

```

link /OUT:my\_program.exe main.obj stdafx.obj

```

10. Чтобы запустить приложение в командной строке разработчика, просто введите имя исполняемого файла (`.exe`), например:

```

my\_program.exe

```