Лабораторная работа 1 (4 часа)
Языки программирования

Этапы обработки исходного кода

Цель работы: Создание исходного файла на языке программирования C++, изучение этапов обработки исходного кода, исследование свойств проекта в интегрированной среде разработки (IDE) Visual Studio. Компиляция и компоновка файлов в командной строке.

Введение.

Создание приложения на языке программирования C++ в интегрированной среде разработки Visual Studio проходит в несколько этапов:

- компиляция исходного кода – трансляция исходного кода, написанного на одном языке программирования, в исходный код на другом языке. В результате компиляции создается файл с расширением **obj** – объектный модуль программы. В IDE Visual Studio 20xx компиляцию модуля с исходным кодом можно осуществить, выбрав **Компилировать** в контекстном меню обозревателя решений (рисунок 1);

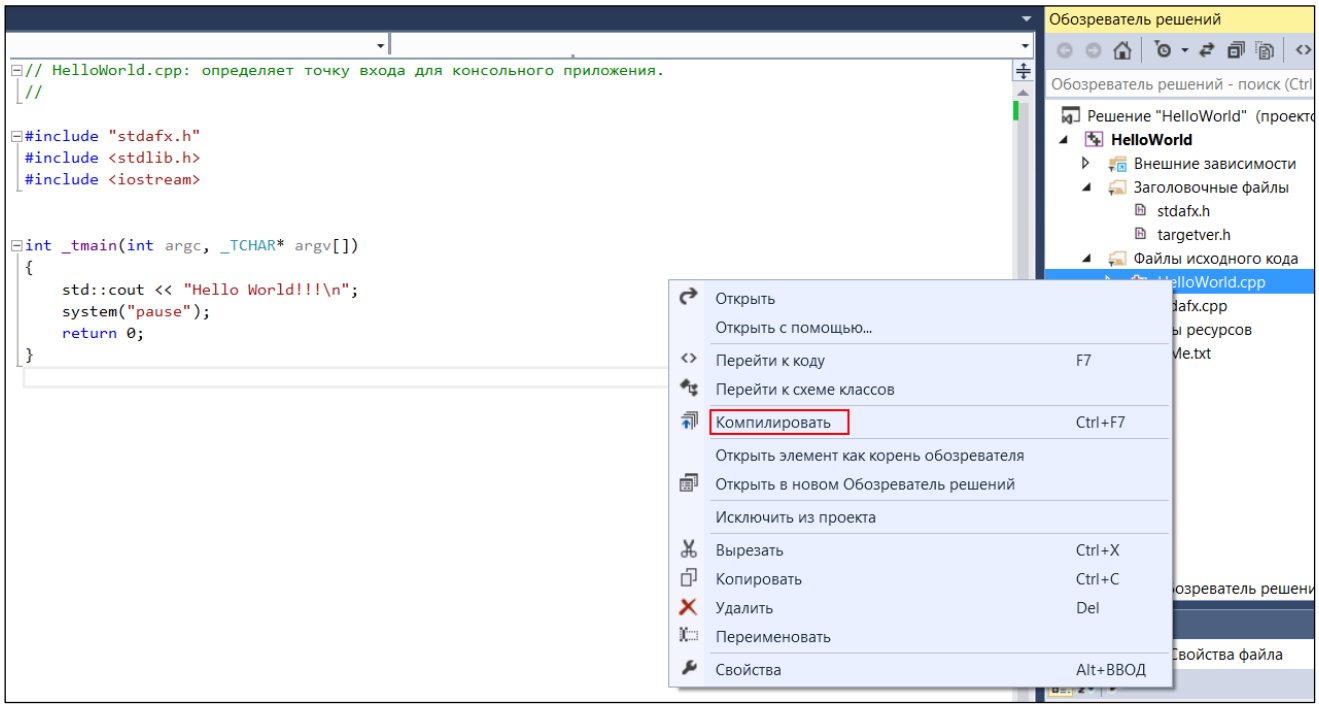


Рисунок 1. Компиляция исходного кода

- компоновка проекта – процесс связывания совокупности объектных файлов и формирование на их основе загрузочного модуля (пункт меню **Сборка -> Собрать решение**).

Выполнить полную сборку проекта можно, используя пункт меню **Сборка ->Построить *имя_проекта.***

Проект консольного приложения на языке C++ в Visual Studio, который при создании был назван HelloWorld, включает файлы и папки решения.

В папке **HelloWorld** (рисунок 2) хранятся файлы, относящиеся к проекту:

- файл HelloWorld.sln – файл решения для созданной программы. Решение может состоять из нескольких проектов, в том числе на разных языках программирования;
- файл HelloWorld.suo – файл параметров решения. Сохраняет настройки для решения (каждый раз при открытии проекта или файла в решении он имеет требуемый внешний вид и поведение);
- HelloWorld.cpp – файл исходного кода;
- HelloWorld.vcproj – файл с настойками проекта.

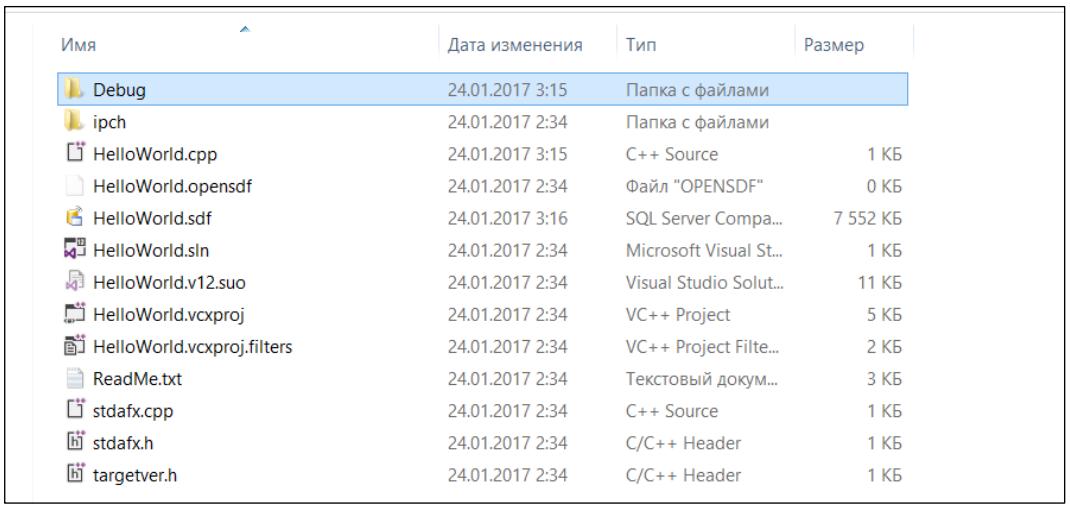
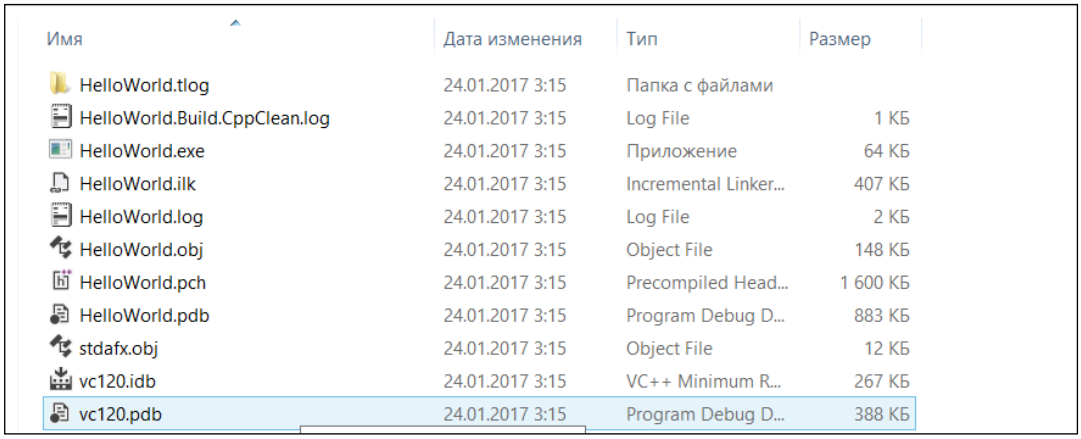


Рисунок 2. Файлы и папки проекта консольного приложения в среде VisualStudio 2013

В папке **Debug** (рисунок 3) хранятся:

- HelloWorld.exe – исполняемый файл проекта;
- HelloWorld.ilk – файл инкрементальной линковки (incremental linker), используемый компоновщиком для ускорения процесса повторной компоновки проекта;
- HelloWorld.pdb – отладочная информация/информация об именах в исполняемых файлах, используемая отладчиком.



Имя	Дата изменения	Тип	Размер
HelloWorld.tlog	24.01.2017 3:15	Папка с файлами	
HelloWorld.Build.CppClean.log	24.01.2017 3:15	Log File	1 КБ
HelloWorld.exe	24.01.2017 3:15	Приложение	64 КБ
HelloWorld.ilk	24.01.2017 3:15	Incremental Linker...	407 КБ
HelloWorld.log	24.01.2017 3:15	Log File	2 КБ
HelloWorld.obj	24.01.2017 3:15	Object File	148 КБ
HelloWorld.pch	24.01.2017 3:15	Precompiled Head...	1 600 КБ
HelloWorld.pdb	24.01.2017 3:15	Program Debug D...	883 КБ
stdafx.obj	24.01.2017 3:15	Object File	12 КБ
vc120.idb	24.01.2017 3:15	VC++ Minimum R...	267 КБ
vc120.pdb	24.01.2017 3:15	Program Debug D...	388 КБ

Рисунок 3. Содержимое папки Debug проекта консольного приложения в среде VisualStudio 2013

Вызов **Командной строки разработчика для Visual Studio** (типовое расположение в Visual Studio 12: C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\Tools\Shortcuts).

Задание 1.

1. Разработайте в Visual Studio программу **HelloWorld**, убедитесь в ее работоспособности.

```
// LP_Lab01.cpp: определяет точку входа для консольного приложения.
//

#include "stdafx.h"
#include <stdlib.h>
#include <iostream>


int _tmain(int argc, _TCHAR* argv[])
{
    std::cout << "Hello World!!!\n";
    system("pause");
    return 0;
}
```

2. Найдите в папке проекта созданный объектный модуль. В какой папке он находится?
3. Внесите изменения в текст программы, чтобы в нем содержались ошибки. Как система программирования сообщает об ошибках?
4. Выполните сборку проекта. После успешной сборки найдите в папке проекта исполняемый модуль.
5. Создайте новый проект консольного приложения.
   Приложение по введенной дате в формате ДДММГГГГ должно вызывать функции:
   - для определения, является ли год ГГГГ високосным;
   - для вычисления порядкового номера дня в году.

Результаты выполнения вывести в консоль.

6. Скомпилируйте текст программы. Найдите в папке проекта созданный объектный модуль. В какой папке он находится?
7. Внесите изменения в текст программы, чтобы в нем содержались ошибки. Посмотрите, как система программирования сообщает об ошибках.
8. Выполните сборку проекта. После успешной сборки найдите в папке проекта исполняемый модуль.
9. Запустите программу на исполнение несколько раз с различными входными (введенными) данными.
10. Установите конфигурацию проекта Release и снова выполните полную сборку проекта. Откройте папку проекта. Какие изменения в ней произошли? Сравните размер отладочной и конечной версии исполняемого модуля. Объясните их различия.
11. Переключитесь в отладочную конфигурацию, установите точки останова и выполните отладочный запуск программы.

    Просмотрите значение локальных переменных на момент останова.

    Измените значение какой-либо переменной, присвоив ей другое корректное значение. Убедитесь, что программа будет при вычислениях использовать новое значение.

    Выполните всю программу в пошаговом режиме два раза. Один раз – с трассировкой содержимого функции, второй раз – выполнив функцию в автоматическом режиме.
12. Модифицируйте код функции таким образом, чтобы у пользователя не было возможности ввести некорректную дату.

Задание 2.

1. Используйте при выполнении лабораторной работы материал лекции 1.
2. Разработайте программу HelloWorld, убедитесь в ее работоспособности.
3. Перестройте проект. Проанализируйте раздел проекта **Внешние зависимости**. Объясните содержимое этой папки.
4. Перестройте проект. Проанализируйте директории проекта. В поддиректории **Debug** найдите файлы с расширением **obj**.
5. Исследуйте свойства проекта связанные с параметрами компилятора C++. Установите параметры компилятора:
   - **Местоположение листинга ASM (/Fa)** в значение **$(IntDir);**
   - **Файл ассемблерного кода** в значение **/FAcs**.

   Ознакомьтесь с разделом **Командная строка**.

6. Перестройте проект. Проанализируйте ASM-листинг. Найдите в листинге ASM-представление операторов C++.
7. Ознакомьтесь с параметрами компилятора https://msdn.microsoft.com/ru-ru/library/fwkeyyhe(v=vs.110).aspx
8. Исследуйте свойства проекта связанные с параметрами компоновки. Ознакомьтесь с разделом **Командная строка**.
9. Ознакомьтесь с параметрами компоновщика https://msdn.microsoft.com/en-us/library/y0zzbyt4(v=vs.110).aspx
10. В директориях проекта найдите файл с расширением **log** (файл журнала построения). Проанализируйте и объясните его содержимое.
11. Запустите консоль **Командная строка разработчика VS201x**. Выполните команду **SET**. Проанализируйте значения переменных окружения.
12. Создайте в корне диска **D** (или другого диска) директорий. Скопируйте из директориев проекта **HelloWorld** в созданный директорий файлы с расширением **h** и **cpp**. Выполните компиляцию и компоновку файлов в командной строке. Убедитесь в работоспособности сформированного исполняемого файла при компоновке.

Вопросы:

- перечислите расширения исходных файлов проекта C++;
- перечислите этапы обработки исходного кода программы;
- в результате какого этапа образуются файлы с расширением **obj**? Что в этих файлах содержится?
- в результате какого этапа образуются файл с расширением **exe**?
- объясните, каким образом программа **cl** определяет местоположение файлов, указанных в папке **Внешние зависимости** (с расширением **h**) проекта при сокращенной записи команды **cl**?
- объясните, каким образом программа **link** определяет местоположение файлов с расширением **lib** при сокращенной записи команды **link**?