

Этапы обработки исходного кода

Цель работы: создание приложения на языке программирования C++ в интегрированной среде разработки Visual Studio 201x; исследование свойств проекта; размещение проекта на жестком диске.

Введение.

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

- 1) компиляция исходного кода – трансляция исходного кода, написанного на одном языке программирования, в исходный код на другом языке (в машинный код). В результате компиляции создается файл с расширением **obj** – объектный модуль программы.

В IDE Visual Studio компиляцию модуля с исходным кодом (файл с расширением **cpp**) можно осуществить, выбрав:

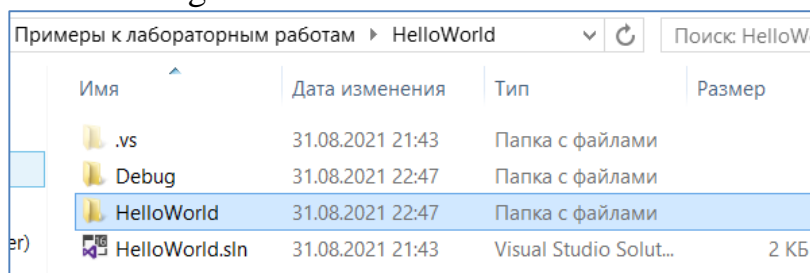
- **Компилировать** в контекстном меню обозревателя решений для нужного файла;
- с помощью горячих клавиш **Ctrl+F7**;
- через пункт главного меню **Сборка** → **Компилировать**;

- 2) компоновка проекта – процесс связывания одного или 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, размещается в папке решения **HelloWorld** (рисунок 1). В папке проекта хранятся папки и файлы, относящиеся непосредственно к проекту:

- файл HelloWorld.sln – файл решения для созданной программы. Решение может состоять из одного или нескольких проектов, в том числе, написанных на разных языках программирования;
- папка проекта HelloWorld;
- папка Debug.



Имя	Дата изменения	Тип	Размер
.vs	31.08.2021 21:43	Папка с файлами	
Debug	31.08.2021 22:47	Папка с файлами	
HelloWorld	31.08.2021 22:47	Папка с файлами	
er) HelloWorld.sln	31.08.2021 21:43	Visual Studio Solut...	2 КБ

Рисунок 1. Файлы и папки проекта консольного приложения в интегрированной среде Visual Studio 2019.

В папке Debug:

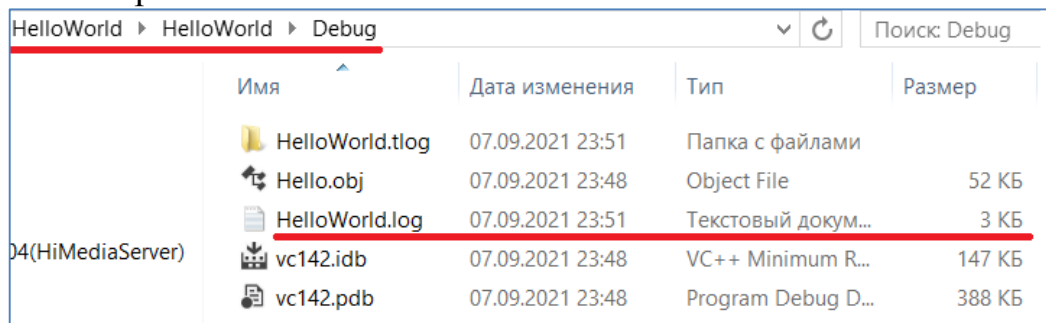
- **HelloWorld.exe** – исполняемый файл проекта;
- HelloWorld.ilc – файл инкрементальной линковки (incremental linker), используемый компоновщиком для ускорения процесса повторной компоновки проекта;
- HelloWorld.pdb – отладочная информация/информация об именах в исполняемых файлах, используемая отладчиком.

В папке HelloWorld:

- **HelloWorld.cpp** – файл исходного кода;
- HelloWorld.vcxproj – файл с настройками проекта;
- и другие.

В папке Debug проекта HelloWorld (рисунок 2):

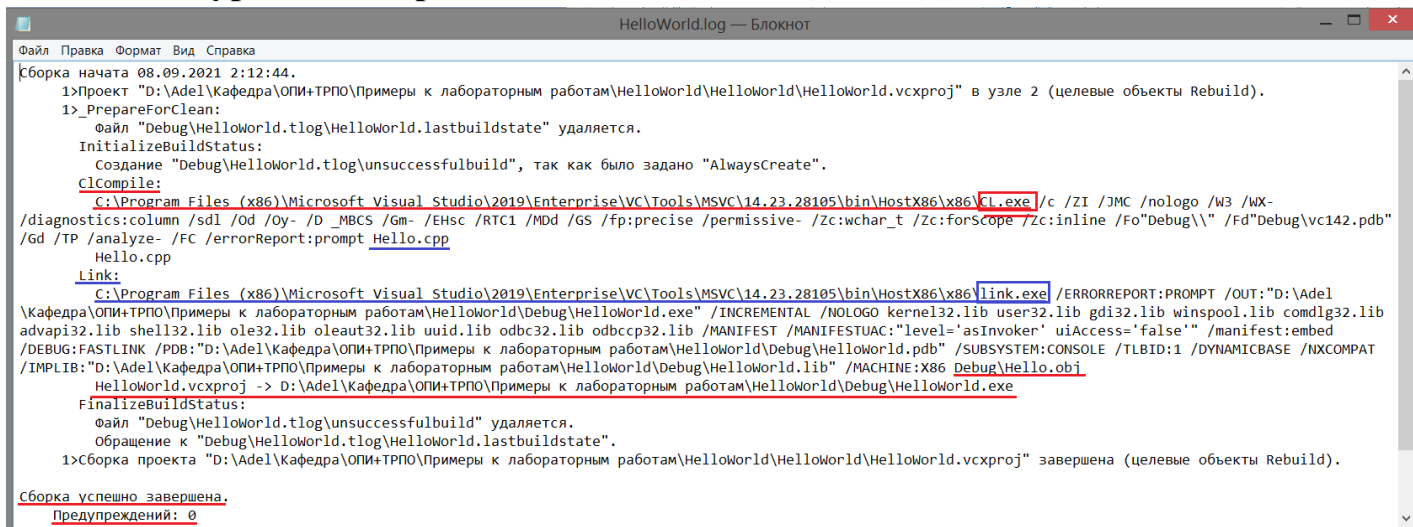
- **Hello.obj** – объектный модуль (результат компиляции);
- **HelloWorld.log** – текстовый документ, содержащий журнал построения.



Имя	Дата изменения	Тип	Размер
HelloWorld.tlog	07.09.2021 23:51	Папка с файлами	
Hello.obj	07.09.2021 23:48	Object File	52 КБ
HelloWorld.log	07.09.2021 23:51	Текстовый докум...	3 КБ
vc142.idb	07.09.2021 23:48	VC++ Minimum R...	147 КБ
vc142.pdb	07.09.2021 23:48	Program Debug D...	388 КБ

Рисунок 2. Файлы в папке проекта Debug.

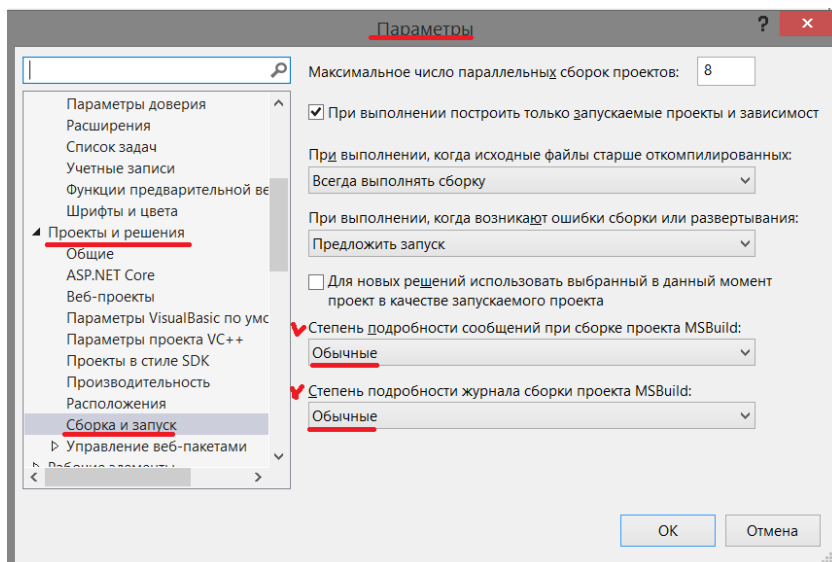
Файл журнала построения



В VS2017 и выше:

для изменения объема сведений, включаемых в журнал сборки необходимо:

- в меню **Сервис (Средства) → Параметры**
- в разделе **Проекты и решения** выбрать **Сборка и запуск**
- в списках **Степень подробности сообщений** и **Степень подробности журнала сборки** при построении проекта MSBuild выбрать **Обычный**



Иерархическая структура компонентов в Visual C++.

Глобальным контейнером, то есть компонентом, включающим в себя другие компоненты, является Решение. Решение может содержать один или несколько проектов.

Проекты являются независимыми компонентами. Они имеют собственную структуру, состоящую из четырех основных каталогов:

Внешние зависимости – содержит ссылки на все модули, которые использует программа.

Файлы заголовков – содержит файлы кода C++ с расширением h.

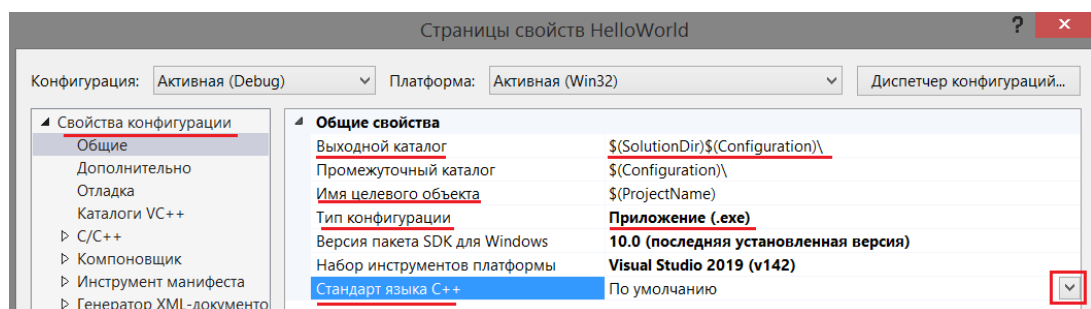
Исходные файлы – содержит файлы кода C++ с расширением cpp.

Файлы ресурсов – содержит файлы, непосредственно не относящиеся к языку C++, но необходимые для работы приложения. Например, мультимедийные файлы.

Код программного проекта может иметь сложную структуру и состоять из нескольких файлов исходного кода и конфигурационных файлов.

Страница свойств проекта.

Раздел **конфигурация**:



В **Общие** определен тип проекта (Приложение); отображается текущая версия VS; стандарт языка. Изменить свойства можно, нажав кнопку

Проект может находиться в одной из конфигураций **Debug** или **Release**.

Текущая конфигурация и целевая платформа, на которой разрабатывается проект, отображается в верхней части окна. Изменить текущие настройки можно в **Диспетчере конфигураций**.

В разделе **C/C++** отображаются текущие настройки компилятора C/C++. Приведем некоторые из них:

Общие – ключ **Уровень предупреждений** позволяет отключить все предупреждения (`/W0`), либо ужесточить уровень проверок и считать все предупреждения ошибками (`/Wall`).

Предварительно откомпилированные заголовки позволяют их включить/отключить, определить имя создаваемого предварительно откомпилированного заголовочного файла и местоположение для полученного выходного файла (с расширением `pch`).

Также можно настроить имена и папки, в которых будут размещаться различные **Выходные файлы**.

Командная строка отображает, с какими параметрами (ключами) выполняется текущая компиляция.

Раздел **Компоновщик** отображает и позволяет изменить текущие настройки и ключи компоновки.

Задание

1. Изучите основные возможности интегрированной среды разработки Visual Studio.
2. В решение **HelloWorld**, созданное в лабораторной работе 1, добавьте новый проект с именем Lab_02. Для создания проекта воспользуйтесь шаблоном **Пустой проект**.
3. В обозревателе решений в папку проекта **Исходные файлы** добавьте новый элемент с именем lab02.
4. Введите исходный текст программы:

```
1  #include <iostream>
2
3  int main()
4  {
5      char hello[] = "Hello, ";
6      char fio[] = "Ivanov Ivan Ivanovich";
7      std::cout << hello << fio << std::endl;
8      return 0;
9  }
```

В 6 строке запишите вашу фамилию, имя, отчество на английском языке.

5. Используйте контекстно-зависимое меню проекта и назначьте его автозапускаемым.
6. Выполните приложение и убедитесь в его работоспособности.

7. Скомпилируйте текст программы. Найдите в папке проекта созданный объектный модуль. В какой папке он находится, время создания/изменения, его размер?
8. Внесите изменения в исходный текст программы, чтобы в нем содержались ошибки. Посмотрите, как система программирования сообщает об ошибках.
9. Выполните сборку проекта. После успешной сборки найдите в папке проекта исполняемый модуль.
10. Перестройте проект. Проанализируйте директории проекта. Какие изменения произошли? В поддиректории **Debug** найдите файлы с расширением **obj**.
11. В контекстном меню обозревателя решений выберите «Перестроить» для активного проекта. Поясните реакцию системы программирования.
12. В директориях проекта найдите файл с расширением **log** (файл журнала построения). Проанализируйте и объясните его содержимое.
13. Установите конфигурацию проекта **Release** и снова выполните полную сборку проекта. Откройте папку решения. Какие изменения в ней произошли? Сравните размер отладочной и конечной версии исполняемого модуля. Объясните их различия.

Вопросы:

- перечислите расширения исходных файлов проекта C++;
- охарактеризуйте файловую структуру решения;
- назовите этапы обработки исходного кода программы;
- как назначить проект автозапускаемым?
- в результате какого этапа образуются файлы с расширением **obj**? Что в этих файлах содержится?
- в результате какого этапа образуются файл с расширением **exe**?
- какая информация содержится в log-файле проекта?
- как получить доступ к свойствам проекта?
- поясните назначение страниц свойств проекта;
- поясните, в чем отличие конфигурации **Debug** от **Release**.