

## Лабораторная работа 1 (2 часа) Конструирование программного обеспечения

### Этапы обработки исходного кода. Командная строка разработчика. Проект, содержащий несколько cpp-файлов

#### Цель работы:

- создание исходного файла на языке программирования C++, изучение этапов обработки исходного кода;
- создание проекта, содержащего несколько cpp-файлов;
- исследование свойств проекта в интегрированной среде разработки (IDE) Visual Studio;
- компиляция и компоновка файлов в командной строке;
- компиляция и компоновка файлов в командной строке с предкомпилируемыми заголовками.

#### Предкомпилированные заголовки

Файл, который содержит предкомпилированные заголовки, имеет расширение ".pch". Файл \*.pch создается в результате выполнения компиляции с ключом "/Yc".

В файл "stdafx.h" необходимо включить заголовочные файлы, которые будут заранее предкомпилироваться.

Заголовочный файл "stdafx.h" должен включаться в файл реализации \*.cpp самым первым.

Для выполнения предкомпиляции заголовочного файла stdafx.h первый cpp-файл компилируется с ключом "/Yc".

Компиляция остальных cpp-файлов проекта выполняется с ключом "/Yu".

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).
- компоновка проекта – процесс связывания совокупности объектных файлов и формирование на их основе загрузочного модуля.

Visual Studio включает в себя командную строку разработчика для создания всех типов приложений на языке C++ непосредственно из командной

строки – от базовых консольных приложений до приложений универсальной платформы Windows, классических приложений, драйверов устройств и т.п.

Инструкции по созданию программы на языке C++ с последующей компиляцией из командной строки:

<https://docs.microsoft.com/ru-ru/cpp/build/walkthrough-compiling-a-native-cpp-program-on-the-command-line?view=msvc-160>

Вызвать **Командную строку разработчика** для **Visual Studio** можно из меню:

**Средства -> Командная строка -> Командная строка разработчика**

### **Задание 1.**

1. Создайте новый проект консольного приложения (выбрать понятное имя проекта).

Приложение по введенной дате в формате ДДММГГГГ должно:

- определить, является ли введенный год ГГГГ високосным;
- вычислить порядковый номер введенного дня в году ГГГГ;
- вычислить сколько дней *осталось* до ближайшего дня вашего рождения.

Реализовать следующие функции:

- для определения, является ли год ГГГГ високосным;
- для вычисления порядкового номера дня в году.

Результаты выполнения вывести в консоль.

2. Выполните только компиляцию программы. Найдите в папках проекта созданный объектный модуль. В какой папке проекта он находится?
3. Выполните сборку проекта. После успешной сборки найдите в папках проекта исполняемый модуль. В какой папке он находится?
4. Проанализируйте лог-файл.
5. 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 при дальнейших вычислениях программа будет использовать новое значение переменной;
  - выполните всю программу в пошаговом режиме два раза. Один раз – с трассировкой содержимого функции (с заходом в функцию), второй раз – выполнив приложение без захода в функцию.
6. Запустите программу на исполнение несколько раз с различными входными данными. Выполните тестирование приложения.

7. Введите некорректные входные данные и выполните приложение. Объясните результат выполнения программы, возникли ли ошибки на этапе выполнения программы?
8. Модифицируйте код таким образом, чтобы у пользователя не было возможности ввести некорректную дату.

### Задание 2.

1. Исследуйте *свойства проекта*, связанные с параметрами *компилятора C++*. На странице «Свойства проекта» становите параметры компилятора:

- **Местоположение листинга ASM (/Fa)** в значение **\$(IntDir)**;
- **Файл ассемблерного кода** в значение **/FAcs**.

Ознакомьтесь с разделом **Командная строка**.

2. *Перестройте* проект. Проанализируйте полученный ASM-листинг. Найдите в листинге ASM-представление операторов языка C++.
3. Ознакомьтесь с параметрами компилятора <https://docs.microsoft.com/ru-ru/cpp/build/reference/compiler-options-listed-alphabetically?view=msvc-160>
4. Исследуйте свойства проекта связанные с параметрами *компоновки*. Ознакомьтесь с разделом **Командная строка**.
5. Ознакомьтесь с параметрами компоновщика [https://msdn.microsoft.com/en-us/library/y0zzbyt4\(v=vs.110\).aspxhttps://docs.microsoft.com/ru-ru/cpp/build/reference/linker-options?view=msvc-160](https://msdn.microsoft.com/en-us/library/y0zzbyt4(v=vs.110).aspxhttps://docs.microsoft.com/ru-ru/cpp/build/reference/linker-options?view=msvc-160)
6. Проанализируйте и объясните содержимое журнала построения.
7. Запустите консоль **Командная строка разработчика MS VS**. Выполните команду **SET**. Проанализируйте значения переменных окружения.
8. Создайте в корне диска **D** (или другого диска) новый директорию. *Скопируйте в созданный директорию* файлы проекта (файлы с расширением **h** и **cpp**). Выполните компиляцию и компоновку файлов в командной строке разработчика. Убедитесь в работоспособности сформированного исполняемого файла.

### Задание 3.

1. Преобразуйте проект в многофайловый проект. Каждая функция должна быть расположена в отдельном **cpp**-файле проекта.
2. Исследуйте свойства проекта в Visual Studio и установите ключи, позволяющие выполнить предкомпиляцию заголовочного файла **stdafx.h**. Выполните сборку проекта многофайлового проекта с перекомпилированными заголовками.

3. Создайте отдельный директорию с именем SELab01 и скопируйте в него только те файлы проекта, которые содержат исходный код.
4. В *Командной строке разработчика* VS отдельно откомпилируйте каждый cpp-файл проекта с помощью команды **cl**. При компиляции первого cpp-файла обеспечьте предкомпиляцию заголовочного файла `stdafx.h` (используйте параметр компилятора `/Yc`). При компиляции остальных cpp-файлов обеспечьте использование результата предкомпиляции (параметр `/Yu`).
5. В Командной строке разработчика VS на основе полученных результатов компиляции скомпонуйте exe-файл с помощью команды **link**.
6. Убедитесь в работоспособности exe-файла.
7. Проанализируйте и объясните содержимое журнала построения.

### Вопросы:

- перечислите расширения исходных файлов проекта C++;
- назовите этапы обработки исходного кода программы;
- в результате какого этапа образуются файлы с расширением **obj**? Что в этих файлах содержится?
- в результате какого этапа образуется файл с расширением **exe**?
- укажите файлы, созданные после каждого вызова команды **cl**; какой из этих файлов содержит предкомпилированный **stdafx.h**?
- укажите файлы, созданные в результате выполнения команды **link**;
- определите понятие **прототип функции**, поясните назначение прототипа функций.
- объясните, как выполнить компиляцию в командной строке разработчика;
- как выполнить компоновку в командной строке разработчика;
- как запустить приложение в командной строке разработчика.