**ЛАБОРАТОРНАЯ РАБОТА № 7**

**СОЗДАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ НА С++ В MS VISUAL STUDIO**

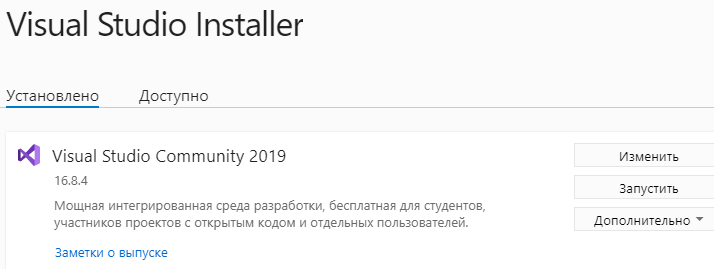
## ТЕОРИЯ И ПОЯСНЕНИЯ

### *Загрузка Visual Studio 2019 Community*

Находим в поиске среду разработки Visual Studio 2019 Community и устанавливаем ее. При установке и загрузке может понадобиться Microsoft account.

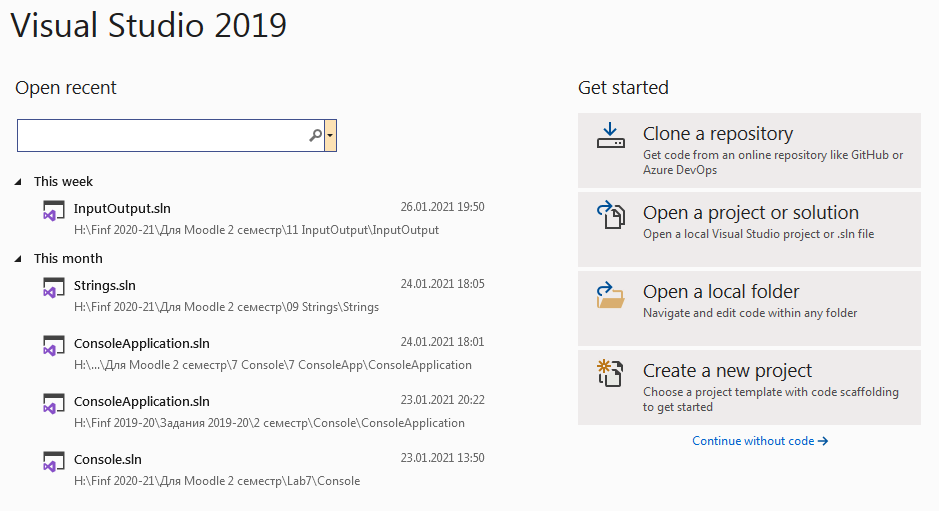
Для разработки будем использовать версию с англоязычным интерфейсом. Если после установки оказался русскоязычный интерфейс, то выполним дополнительные настройки.

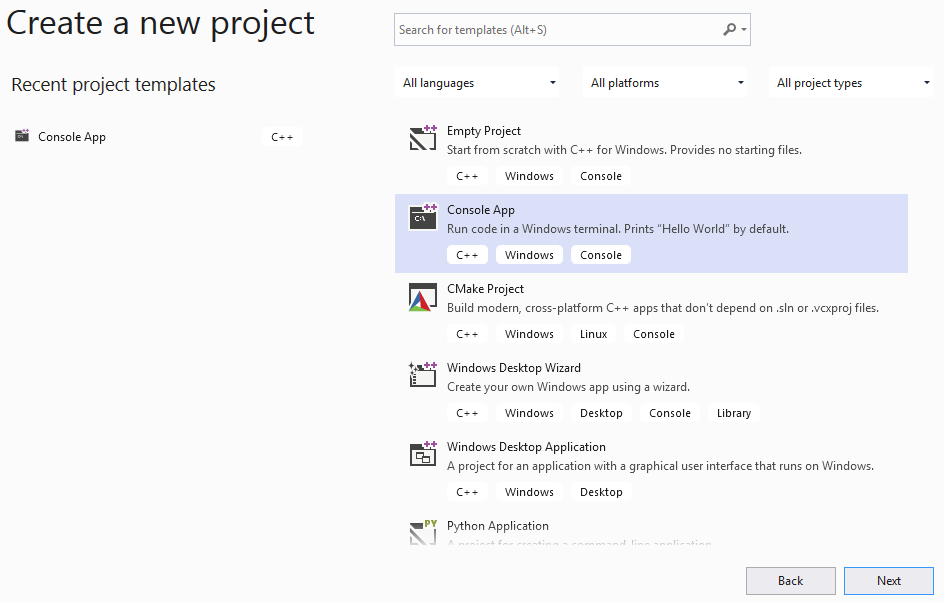
Загружаем приложение Visual Studio Installer.

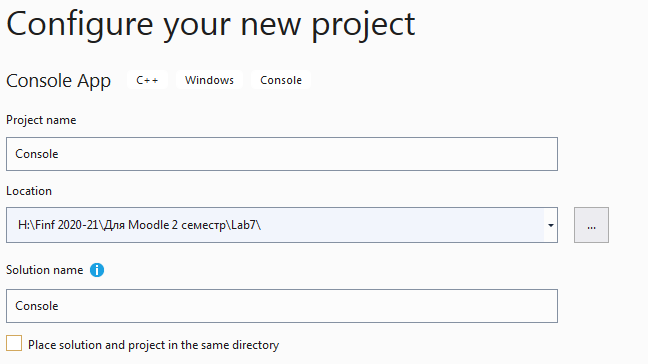


Выбираем действие Изменить. Далее пункт Языковые пакеты – отмечаем Английский – кнопка Изменить.

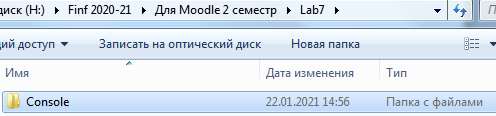
Несколько следующих скриншотов объясняют, какие настройки и действия рекомендуется выполнить в Visual Studio 2019 для перехода к созданию консольного приложения.



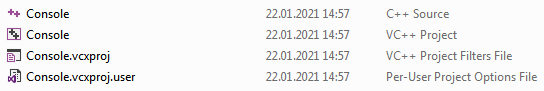




Содержимое папки Lab7 с проектом Console:

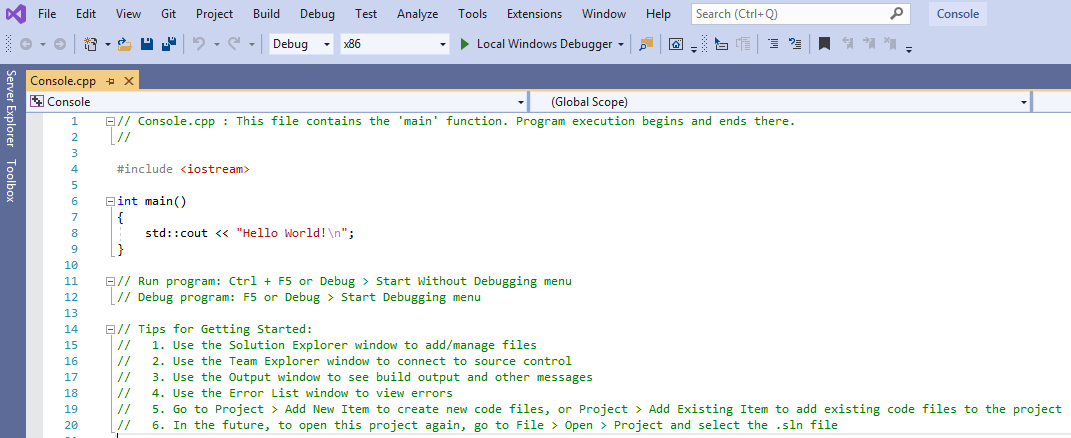








### *Шаблон для консольного приложения*



**#include** – директива препроцессора для подключения к проекту содержимого заголовочных файлов (.h), содержащих интерфейсную информацию о стандартных функциях и константах, которые вы хотите использовать в вашей программе. Эти директивы помещаются в начале исходного файла. #include <iostream> – подключение к проекту стандартного файла для организации потокового ввода-вывода. При выполнении лабораторной работы могут понадобиться также следующие файлы:

#include <math.h> – для использования математических функций и констант;

#include <stdlib.h> – для генерации случайных значений и ряда других полезных функций.

Имена файлов в директивах #include задаются по-разному:

* в двойных кавычках (“stdafx.h”), если файл находится непосредственно в проекте;
* в угловых скобках (<math.h>), если это стандартный заголовочный файл C, который находится в специальных каталогах Visual C++.

В соответствии со стандартом C++ имена стандартных файлов в директивах #include можно задавать без расширения: <iostream>.

После директив include следует указать использование пространства стандартных имен:

**using namespace std;**

Программа на языке C представляет собой некоторый набор функций. Все функции расположены на одном уровне, т.е. вложенных функций нет. Одна из функций определяет точку входа в программу (место, с которого программа начнет выполняться). В стандартном C она должна иметь имя main, но в системе Visual C++ возможны и другие варианты, в которых к имени добавляется префикс.

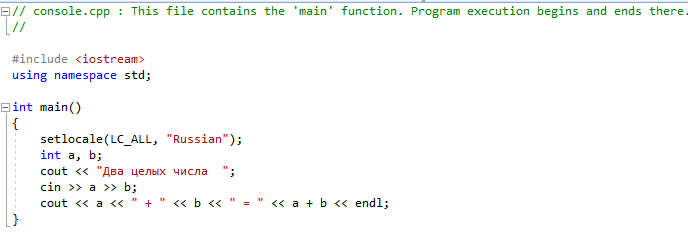
Приведенный шаблон позволяет создать простейшую пустую программу, которая может быть оттранслирована и запущена на выполнение. Для создания своей собственной программы нужно просто разместить в функции main объявления переменных и необходимые операторы – внутри фигурных скобок.

### *Таблица сопоставления синтаксиса языков Паскаль и Си*

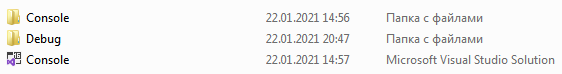
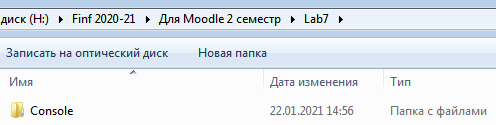
|  |  |  |
| --- | --- | --- |
| Описание, оператор | Паскаль | Си |
| Общие соглашения | Заглавные и строчные буквы не различаются (Case Insensitive) | Заглавные и строчные буквы различаются (**Case Sensitive**) |
| Объявление типов | Раздел var в начале программы  имя\_переменной : ТИП;  integer real | В любом месте программы до первого использования переменной  ТИП имя переменной = значение;  **int** (4 байта) **float** (4) **double** (8) |
| Присваивание | имя\_переменной := выражение | имя\_переменной = выражение; |
| Составной оператор | begin … end | { … } |
| Ветвление | if УСЛОВИЕ then УСЛ\_TRUE else УСЛ\_FALSE | **if** **(**УСЛОВИЕ**)** УСЛ\_TRUE; **else** УСЛ\_FALSE; |
| Цикл с известным числом шагов | for параметр := НачЗн to КонЗн do  тело\_цикла  параметр – целое | **for** **(**параметр**=**НачЗн; параметрКонЗн;  изменение параметра оператором присваивания**)**  тело\_цикла; |
| Цикл с предусловием | while УСЛОВИЕ do тело\_цикла | **while (**УСЛОВИЕ**)** тело\_цикла; |
| Потоковый ввод данных | read(имя\_переменной, …) | **cin >>** имя\_переменной **>>** … ; |
| Потоковый вывод данных | write(имя\_переменной, ‘текст’…) | **cout <<** имя\_переменной << “текст” **<<** … ; |
| Потоковый вывод данных с форматированием | write(целая\_переменная:LL, вещественная\_переменная:LL:PP) | **printf**(“%LLd, %LL.PPf ”, целая\_переменная, вещественная\_переменная); |
| Перевод вывода на новую строку | writeln | cout << endl; printf(“\n”); |

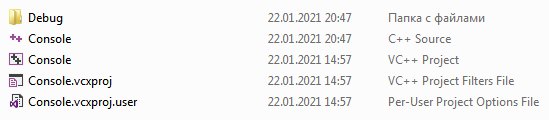
### *Отладка программы*

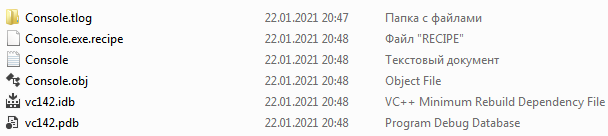
Написанную программу сначала рекомендуется откомпилировать в режиме отладчика **Debug**. Система производит построение проекта – **Build**.

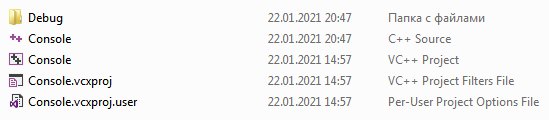


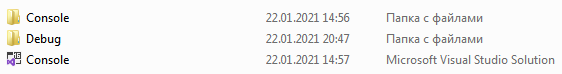
Содержимое папки Lab7 с проектом Console после отладки и построения проекта:

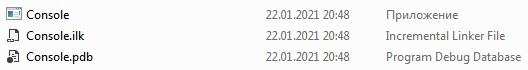


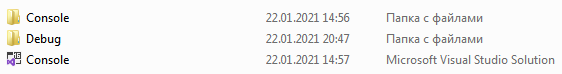






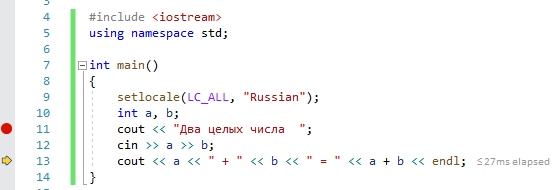






Сообщения об ошибках выводятся в отдельное окно **Output**.

Для отладки можно использовать точки останова и режим пошагового исполнения программы (без захода в функции – F10, с заходом в функции – F11).

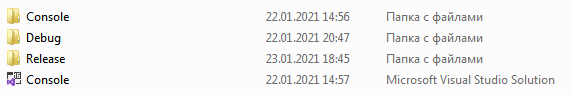


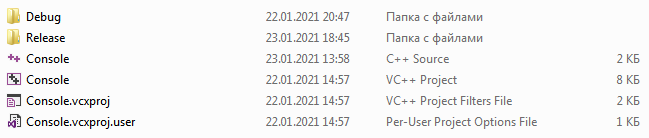
Выход из режима пошагового исполнения – Shift + F5

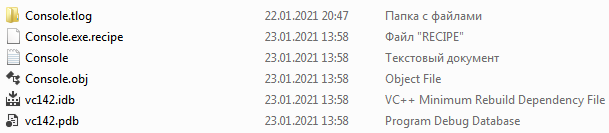
Полностью отлаженную программу нужно запустить в режиме Release. В полученном исполняемом файле не будет информации для отладчика, не важной для работы программы. За счет этого размер исполняемого файла существенно сократится.

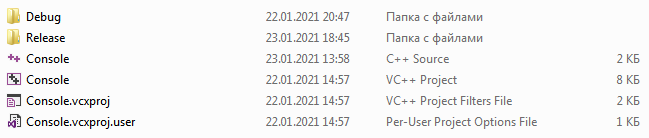
Содержимое папки Lab7 с проектом Console после формирования релиза проекта:

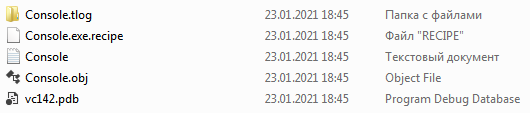


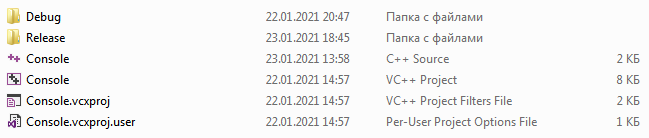


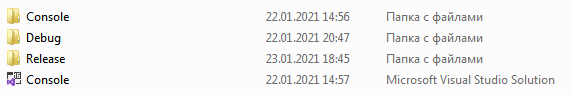


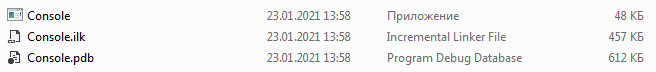


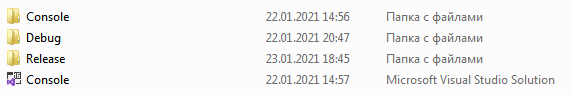


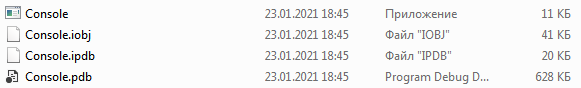


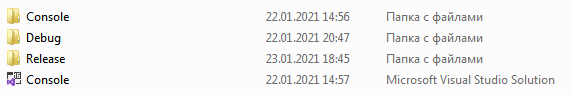












## ЗАДАНИЯ

Оба задания следует реализовать в ОДНОМ проекте в консольном режиме системы MS VS.

**Рекомендация**. Для корректного использования кириллицы добавьте в исходный код вызов функции setlocale(LC\_ALL, "Russian");

### ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИИ. Напишите программу вычисления значений функции в десяти равноотстоящих друг от друга точках интервала (0, 4] возможных значений аргумента *x*.

**На входе** программы задается начальная точка интервала возможных значений аргумента.

**Результаты вычислений** выводятся на экран в удобном для чтения виде без использования параметров форматирования.

### ЧИСЛА ФИБОНАЧЧИ. Напишите программу вычисления S - суммы N первых чисел Фибоначчи. Вычисления следует проводить, пока сумма чисел не превысит наперед заданного большого числа MAX.

**На входе** программы задается значение MAX.

**Результаты вычислений:** значения N и S.