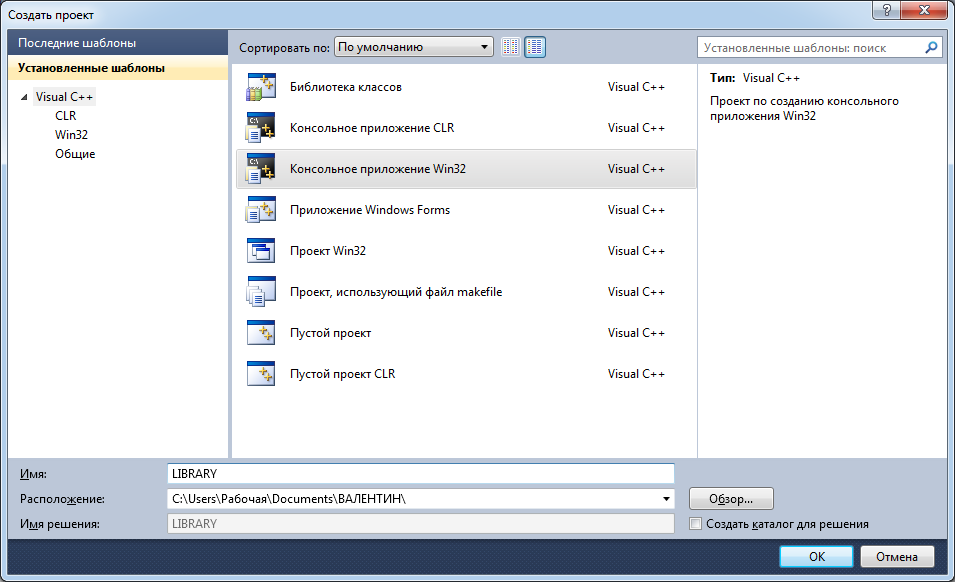
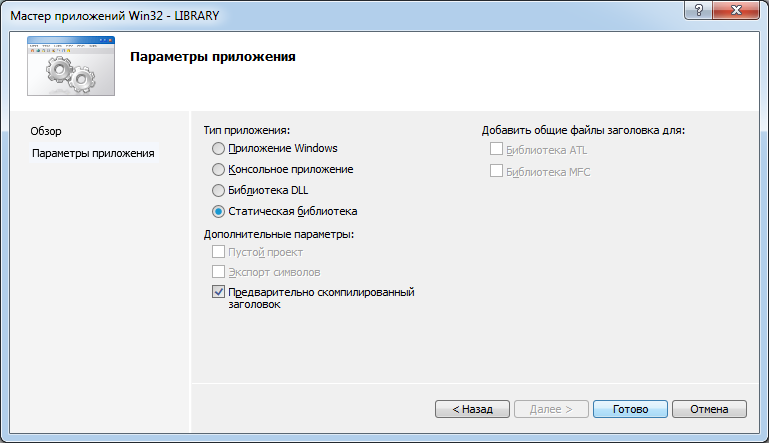
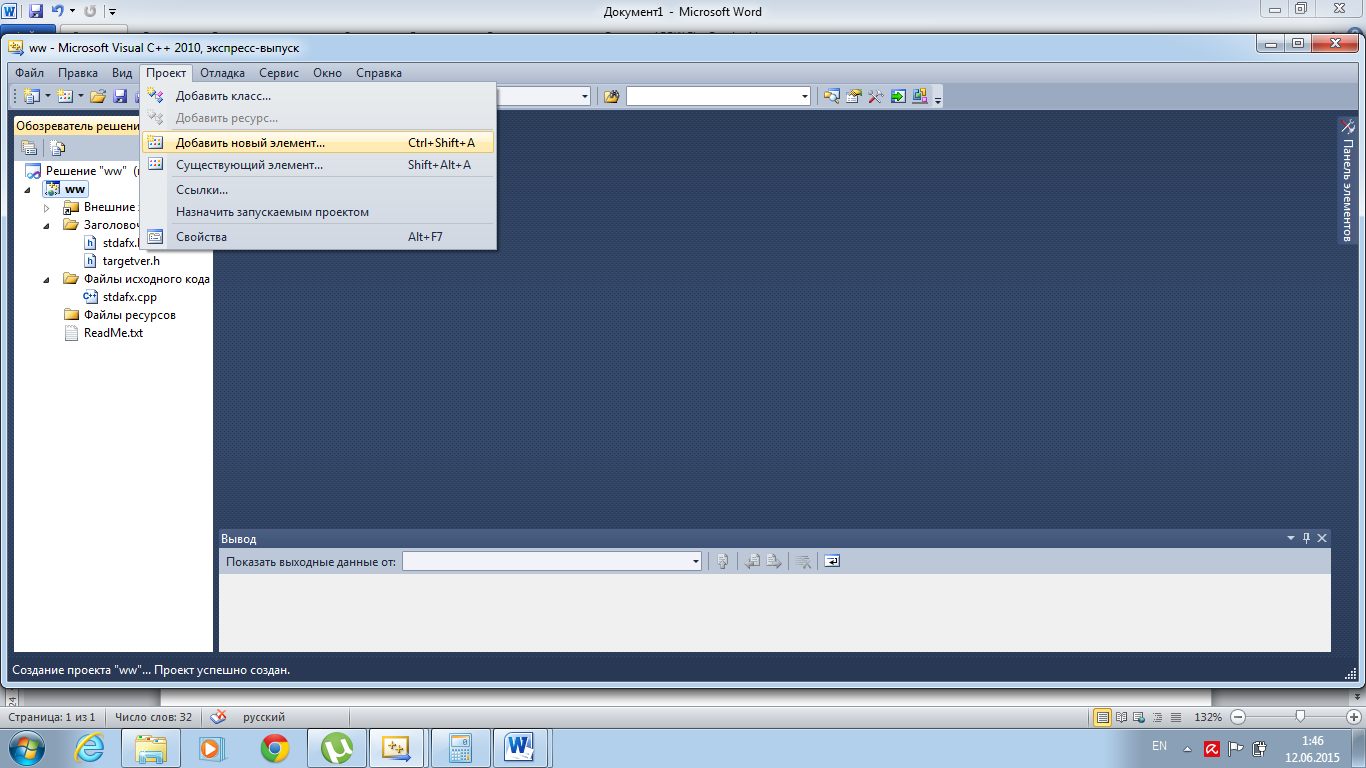
**Тема:** Создание библиотеки программиста в среде Visual Studio 2013

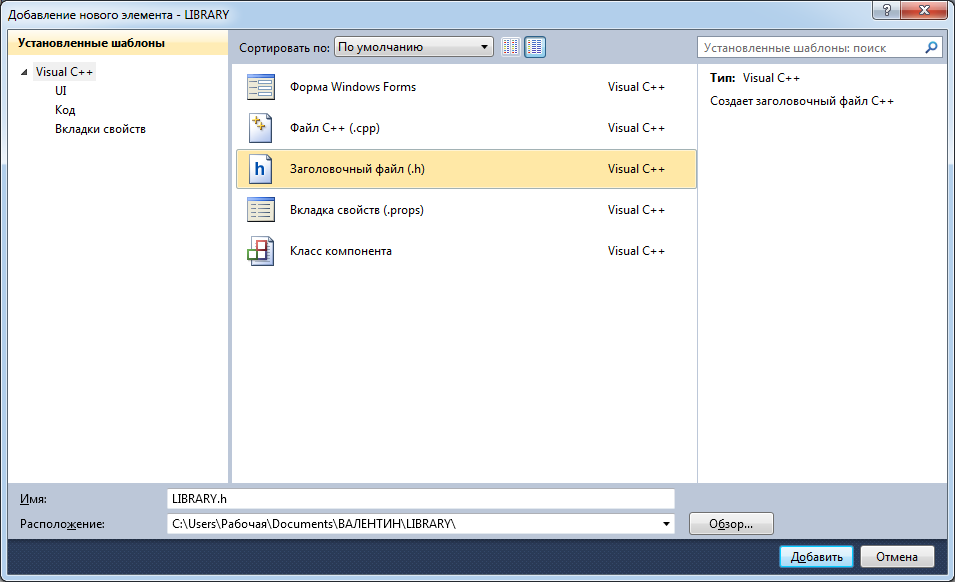
1. Создайте новый проект в среде Visual Studio. Выберите тип проекта – консольное приложение Win32. Введите имя проекта.



1. Затем настройте параметры проекта: статическая библиотека.



1. Нажмите кнопку «Готово». В открывшемся окне разработки выберите пункт меню   
   Проект–>Добавить новый элемент
2. Необходимо добавить в проект библиотеки заголовочный файл (.h), а затем аналогично добавить файл С++ (.cpp)



1. В заголовочном файле необходимо прописать заголовки библиотечных функций.
2. В файле .cpp нужно описать реализацию каждой функции (тело функций).
3. Сохраните проект библиотеки и выполните построение (компиляцию). На выполнение файл библиотеки не посылается (для проверки работоспособности функций библиотеки их нужно вызвать из другой программы). В окне вывода результатов построения появятся примерно следующие строки:

1>------ Построение начато: проект: LIBRARY, Конфигурация: Debug Win32 ------

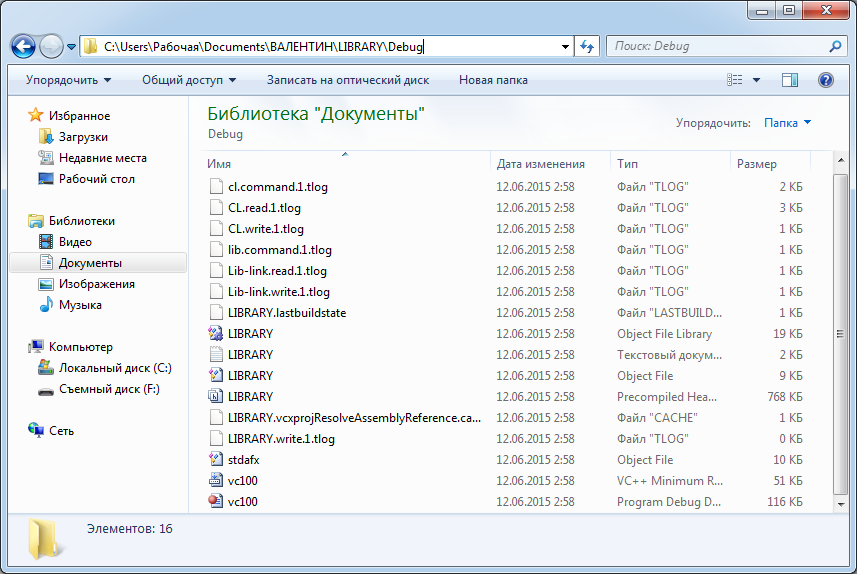
1> stdafx.cpp

1> LIBRARY.cpp

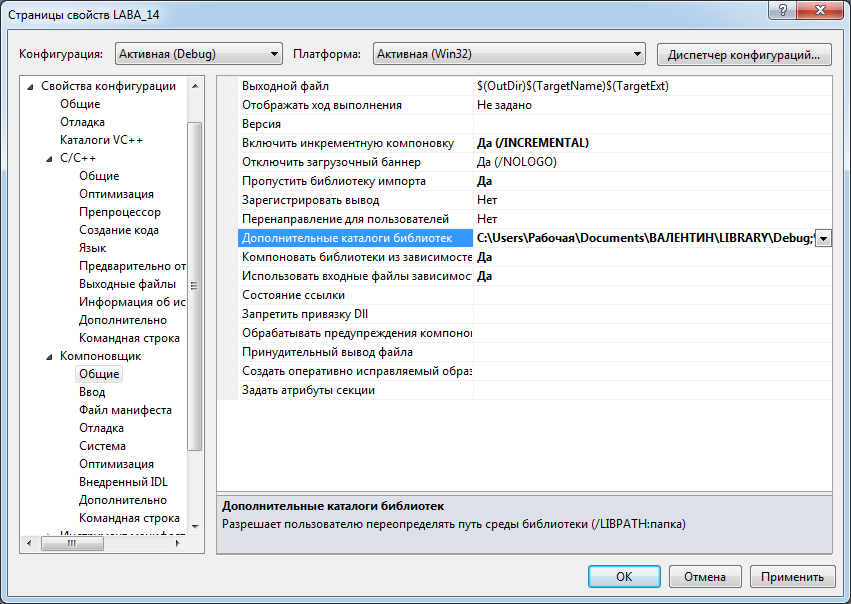
1> LIBRARY.vcxproj -> C:\Users\Рабочая\Documents\ВАЛЕНТИН\LIBRARY\Debug\LIBRARY.lib

========== Построение: успешно: 1, с ошибками: 0, без изменений: 0, пропущено: 0 ==========

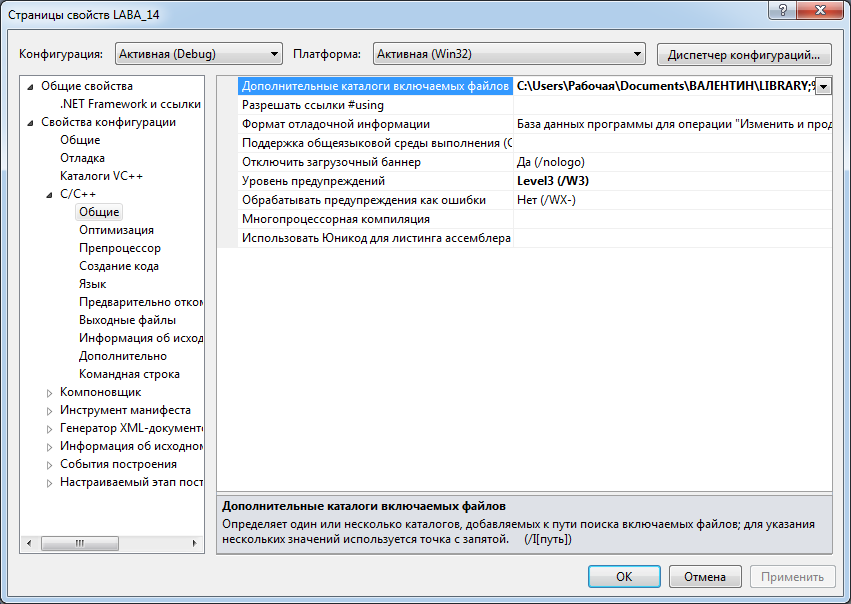
1. После компиляции во вложенной папке Debug появиться файл библиотеки с тем же именем и расширением .lib (тип файла Object File Library). Если компиляция прошла успешно, то библиотека готова. Теперь её может подключить любая программа.

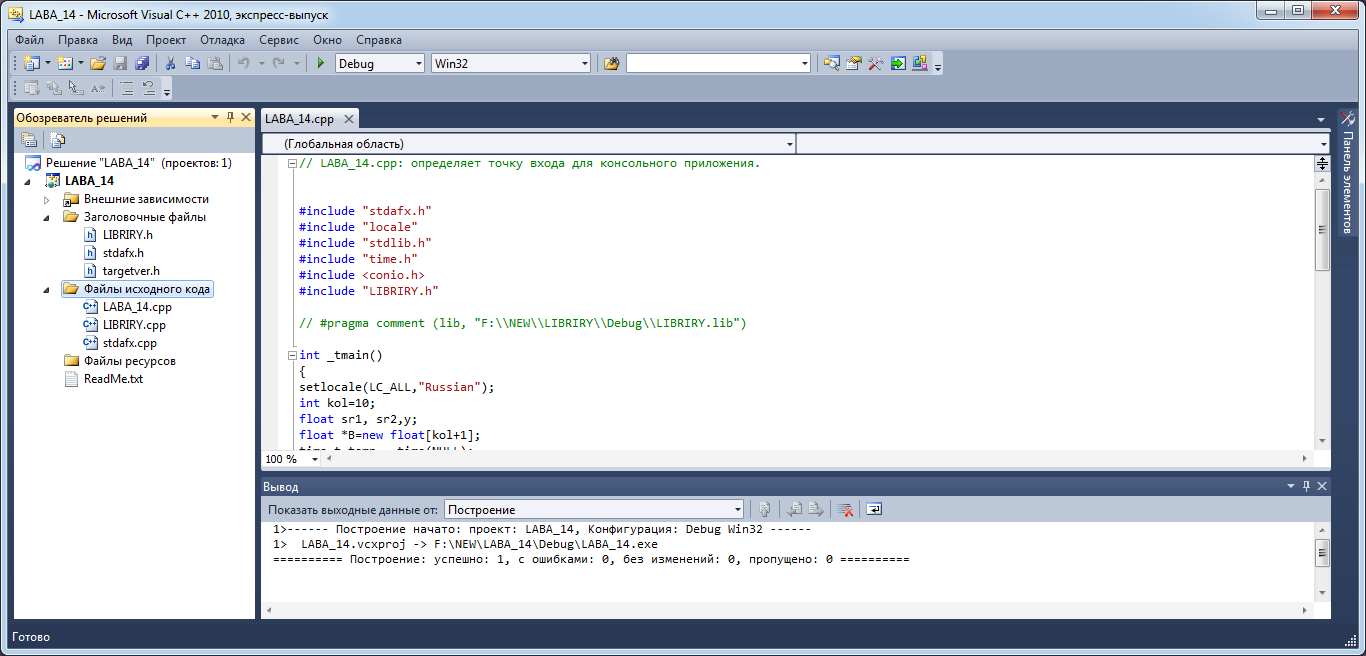


1. Теперь нужно создать отдельный проект для проверки работоспособности функций библиотеки. Создайте новый проект в среде Visual Studio. Выберите тип проекта – консольное приложение Win32. Введите имя проекта.
2. Настройте параметры проекта. На вкладке Компоновщик –>Общие необходимо указать путь к библиотеке (в строке «Дополнительные каталоги библиотек»)



1. На вкладке C/C++ –>Общие необходимо указать основной каталог, где находятся файлы .h и .cpp вашей библиотеки (в строке «Дополнительные каталоги включаемых файлов»)



1. Теперь нужно добавить в проект файлы LIBRARY.h и LIBRARY.cpp при помощи меню   
   Проект–> Добавить существующий элемент
2. В основной программе подключите заголовочный файл собственной библиотеки с помощью директивы #include "LIBRARY.h"
3. Затем в основной программе пропишите вызов функций библиотеки. При наборе имени функции редактор кода должен выводить всплывающую строку с подсказками о количестве и типах параметров этой функции.

**Пример кода библиотеки программиста**

В библиотеку поместим следующие функции для работы с вещественным одномерным массивом произвольного размера: ввод массива, вывод массива на экран, вычисление среднего арифметического элементов массива от элемента с одним номером до элемента с другим номером, вставка в массив заданного значения на указанную позицию.

/\* Заголовочный файл LIBRARY.h \*/

#pragma ones

void Input\_array(float \*A,int k);

void Out\_array(float \*A,int k);

float Srednee (float \*A,int k,int n1,int n2);

void Vstavka(float \*A,int &k,float x,int n);

/\* Файл реализации LIBRARY.cpp \*/

#include "StdAfx.h"

#include <stdio.h>

#include <stdlib.h>

void Input\_array(float \*A,int k)

// функция заполняет одномерный массив случайными числами

// входные параметры: адрес начала массива, количество элементов

{

for (int i=0;i<k;i++)

A[i]=(float)(rand()%101)/(rand()%11+1)-5;

}

void Out\_array(float \*A,int k)

// функция выводит одномерный массив на экран

// входные параметры: адрес начала массива, количество элементов

{

for (int i=0;i<k;i++)

printf("%7.1f",A[i]);

printf("\n");

}

float Srednee (float \*A,int k,int n1,int n2)

// функция вычисляет среднее арифметическое части массива

// входные параметры: адрес начала массива, количество элементов массива,

// номер начала отсчёта, номер конца

// результат функции: среднее арифметическое части массива

// если номера неверны, возвращает значение ноль

{

int x; float s=0,sred=0;

if (n1<0 || n1>k-1) return 0;

if (n2<0 || n2>k-1) return 0;

if (n1>=n2) return 0;

x=0;

for(int i=n1;i<=n2;i++)

{

s=s+A[i];

x++;

}

if (x>0) sred=s/x;

return sred;

}

void Vstavka(float \*A,int &k,float x,int n)

// функция вставляет в одномерный массив заданное значение на нужную позицию

// входные параметры: адрес начала массива, количество элементов (изменяется),

// вставляемое значение, номер позиции вставки

// результат работы функции: изменённый массив

{

if (n<k)

{

for (int i=k-1; i>=n;i--)

A[i+1]=A[i];

A[n]=x;

k++;

}

}

/\* Файл программы, подключившей библиотеку LABA\_14.cpp \*/

#include "stdafx.h"

#include "locale"

#include "stdlib.h"

#include "time.h"

#include "LIBRARY.h"

void main()

{

setlocale(LC\_ALL,"Russian");

int kol=10;

float sr1, sr2,y;

float \*B=new float[kol+1];

srand(time(NULL);

Input\_array(B,kol);

Out\_array(B,kol);

sr1=Srednee(B,kol,0,kol/2-1);

sr2=Srednee(B,kol,kol/2,kol-1);

printf("\n Среднее значение первой половины массива:");

printf("\n sr1=%5.2f",sr1);

printf("\n Среднее значение второй половины массива:");

printf("\n sr2=%5.2f \n",sr2);

y=sr1+sr2;

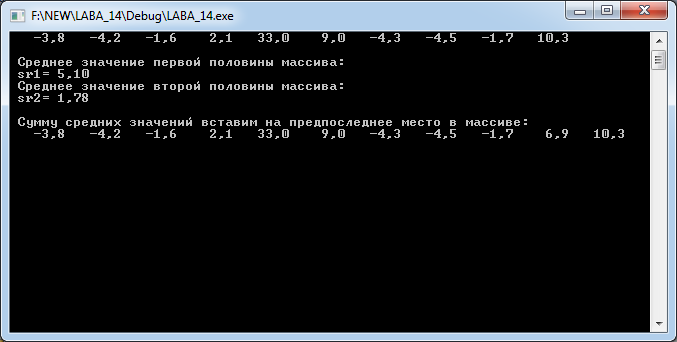
printf("\n Сумму средних значений вставим на предпоследнее место в массиве: \n");

Vstavka(B,kol,y,kol-1);

Out\_array(B,kol);

getch();

}

**Результаты работы программы:**