

Лабораторная работа № 1

Тема: Знакомство со средой разработки программ Visual C++ .NET на языке C++

Цель: Изучить среду разработки программного обеспечения Visual C++ .NET, освоить настройки данной среды, научиться создавать проекты и набирать программы, освоить методы компиляции и запуска программ, а также их отладки посредством встроенного в среду отладчика.

Краткая теория

Интегрированная среда разработки «MS Visual Studio .NET» предназначена для создания программного обеспечения на основе технологии .NET Framework, которая применяется в настоящее время для разработки приложений под ОС Windows XP/Vista/7. Одной из отличительных особенностей этой технологии разработки ПО является возможность написания программ на различных языках программирования. Основными из поддерживаемых языков программирования в .NET являются: C++, C# и Visual Basic. Поддержку того или иного языка программирования в данной IDE можно как включать так и не включать, поэтому довольно часто при ссылке на эту среду программирования ее название пишут с указанием языка программирования, на котором будет вестись разработка (например, MS Visual C++ или MS Visual C#).

В рамках изучения дисциплины «Программирование» во втором семестре выполняется освоение языка программирования C++ (как объектно-ориентированного языка), а также разработка приложений с графическим интерфейсом. Для этих целей при выполнении всех лабораторных работ в данном семестре будет использоваться IDE Visual C++ .NET 2008.

Запуск среды «MS Visual C++» осуществляется путем запуска файла **devenv.exe** из рабочего каталога среды: **C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE**. Как правило на рабочем столе или меню программ ОС Windows располагается ярлык для запуска данной среды разработки (Microsoft Visual Studio 2008).

После запуска программы на экране появится главное окно программы, которое (в зависимости от настроек) может выглядеть так, как показано на рисунке 1.

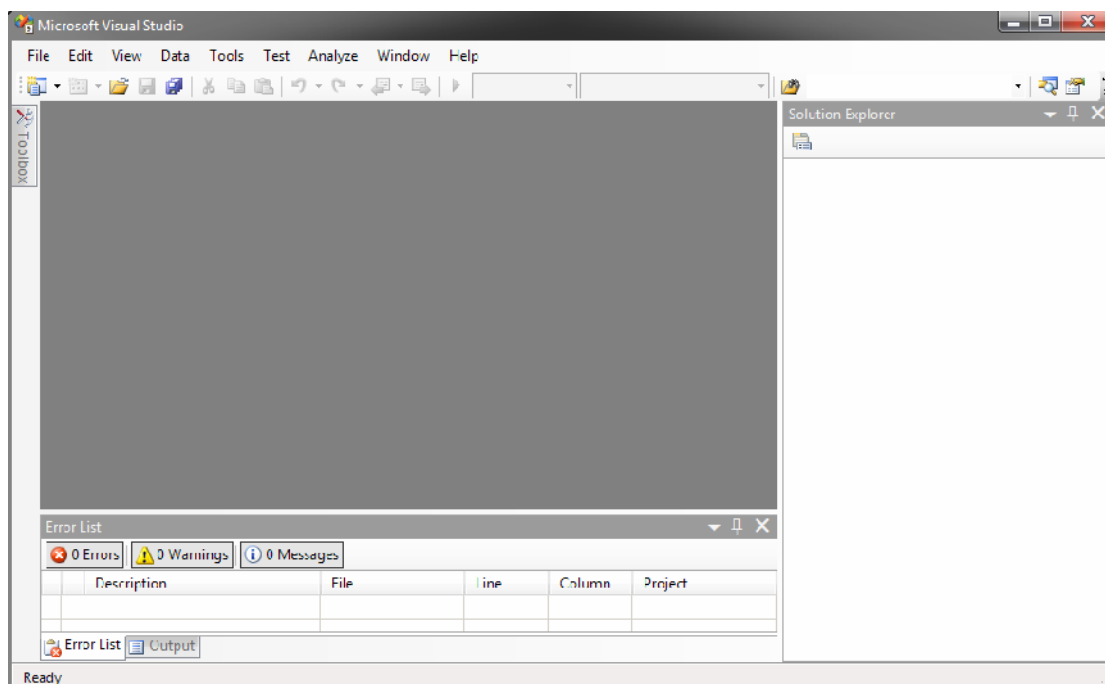


Рисунок 1 – Окно среды разработки «Visual Studio .NET 2008»

Прежде чем перейти непосредственно к разработке программ необходимо произвести некоторые настройки среды разработки (в принципе настройки могут быть проведены и

Программирование

раньше, но проверить все же стоит). Окно настроек (рисунок 2) среды разработки вызывается из главного меню: **Tools** → **Options...** Данная среда разработки является одной из наиболее мощных и развитых систем программирования, поэтому она имеет довольно большое количество параметров, изучение которых может потребовать длительного времени. Поэтому в рамках данной лабораторной работы мы ограничимся рассмотрением лишь базовых параметров, а изучение остальных параметров настройки данной среды разработки оставляется на усмотрение студента.

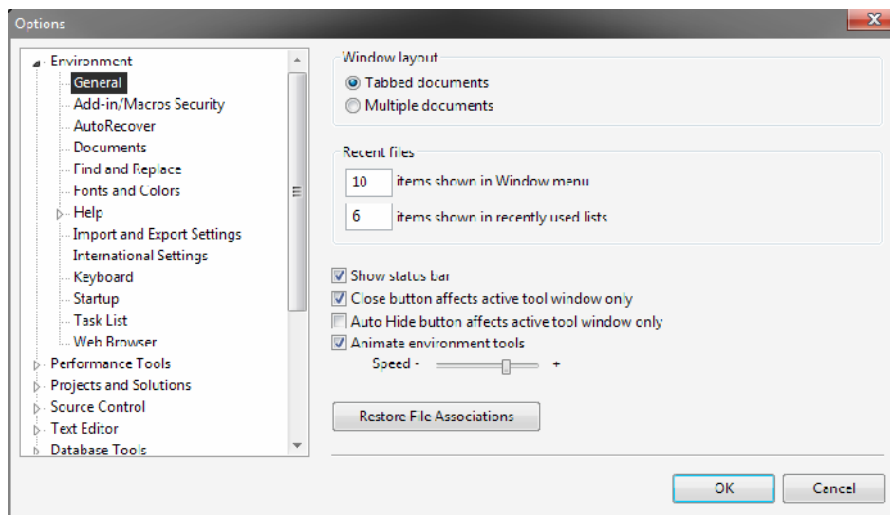


Рисунок 2 – Окно настроек среды разработки

Сначала выполним настройку шрифта, которым будет отображаться текст программы. Для этого в дереве параметров (левая часть окна) выберем пункт «Fonts and Colors» в ветви «Environment» и на экране получим окно настроек в виде, приведенном на рисунке 3.

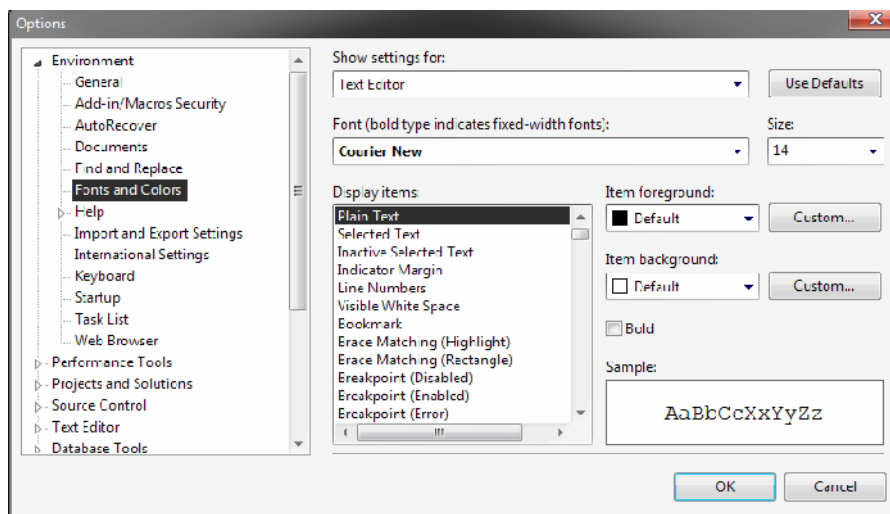


Рисунок 3 – Окно настройки шрифта среды разработки

В этом окне необходимо установить шрифт Courier New, а размер шрифта рекомендуется выбрать в пределах 12 – 14 пт (см. рисунок 3). Остальные параметры можно оставить без изменений.

Далее перейдем к параметрам настройки расположения программных проектов. Для этого выберем пункт «General» в ветви «Projects and Solutions» (рисунок 4). В этом окне необходимо указать место расположения по умолчанию создаваемых проектов. В качестве такого места целесообразно указать каталог, расположенные на диске D (например, D:\09KB) в поле Projects Location. В полях User project template locations и User item template locations указываются месторасположения временных файлов, создаваемых для нужд среды разработки. Значения этих полей можно оставить по умолчанию.

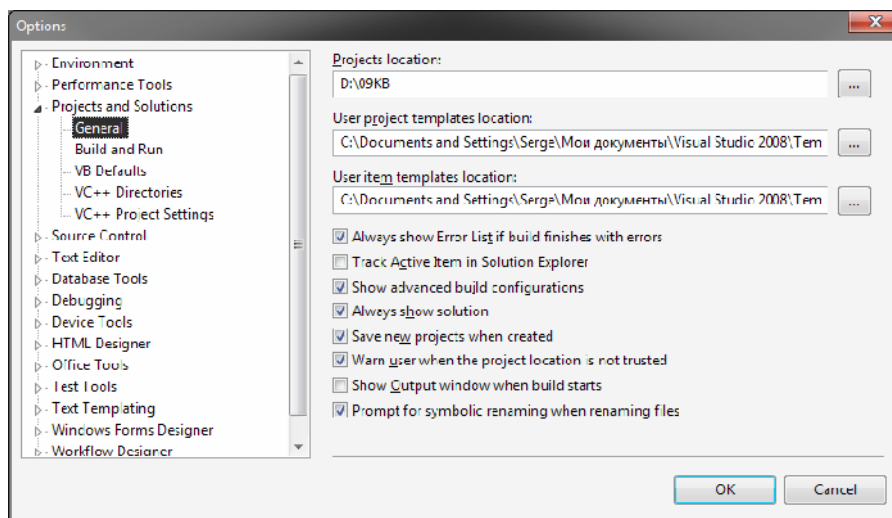


Рисунок 4 – Окно настройки расположения проектов

Дополнительно можно выполнить настройку запуска среды разработки. Для этого необходимо выбрать пункт «Startup» в ветви «Environment» (рисунок 5). В появившемся окне в поле «At startup» необходимо выбрать «Show empty environment», а также убрать флажок «Download content every:».

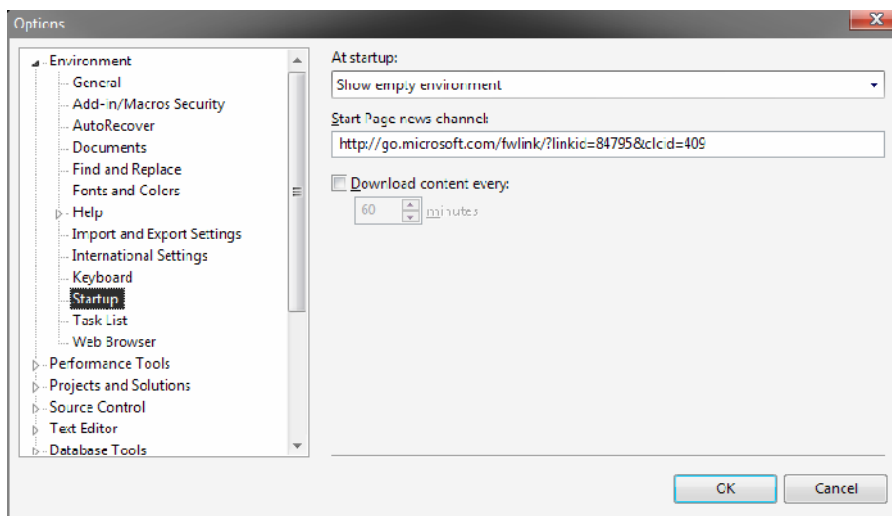


Рисунок 5 – Окно настройки запуска среды разработки

Сейчас мы подошли к моменту когда можно создавать проект. Для начала разработки программы в среде «Visual C++» **необходимо** создать проект. Это выполняется в окне создания проекта, которое вызывается из главного меню программы: **File → New → Project**. На открывшемся окне (рисунок 6) необходимо выбрать тип создаваемого проекта.

В первой половине данного семестра мы будем осваивать объектно-ориентированное программирование на примере создания консольных приложений. Поэтому в качестве типа проекта выберем в левой части окна пункт «Win32» в ветви «Visual C++», которая может располагаться в ветви «Other Languages». В окне справа необходимо выбрать вид приложения «Win32 Console Application». В нижней части окна необходимо указать название проекта (в поле Name), а также, при необходимости, изменить расположение проекта (поле Location). Дополнительно здесь можно указать необходимость создания отдельной директории для проекта и ее имя.

После нажатия на кнопку «ОК» появится следующее окно (рисунок 7) в котором можно открыть вкладку «Application Settings». В этом окне необходимо убрать флажок «Precompiled header», как это показано на рисунке 7. Для завершения мастера просто нажмите «Finish».

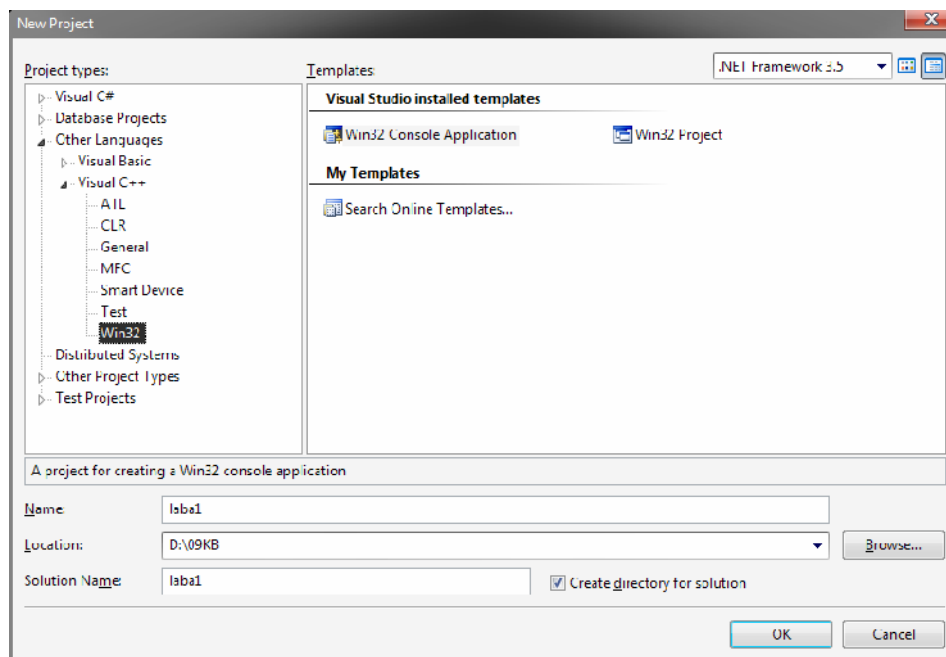


Рисунок 6 – Окно создания нового проекта

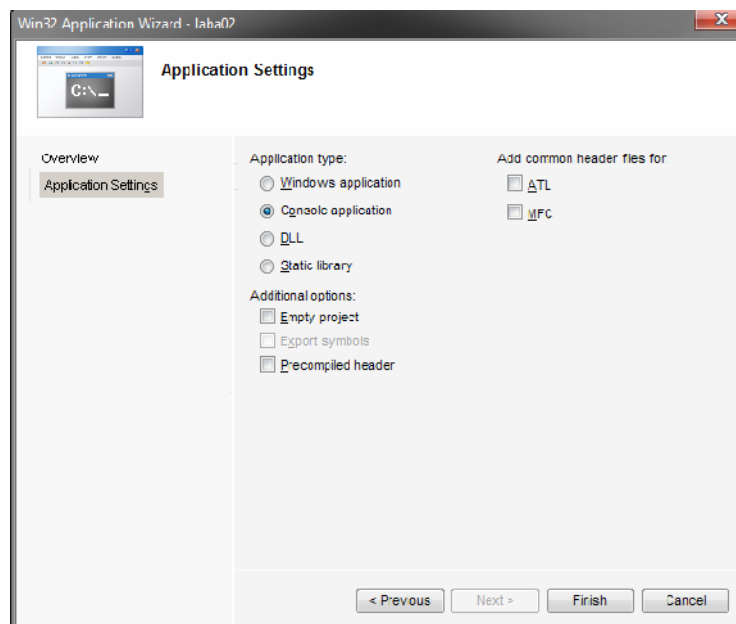


Рисунок 7 – Окно параметров нового проекта

После того, как проект создан в главном окне среды разработки «Visual C++» (рисунок 8) появится окно инспектора проекта, в котором будут отображены все файлы проекта. В проекте могут присутствовать заголовочные файлы (ветвь Header Files), файлы ресурсов (ветвь Resource Files) и файлы исходного кода (ветвь Source Files). Двойной клик левой кнопкой мыши по имени файла приведет к открытию редактора текста программы для этого файла.

Среда Visual C++ генерирует исходный код, содержащий описание функции `_tmain`, которая является точкой входа в консольное приложение. Это описание не соответствует стандарту языка C++, поэтому его необходимо изменить. Для этого сначала необходимо закомментировать или удалить подключение библиотеки `stdafx.h`. Затем объявление функции `_tmain` следует исправить на стандартное, как и в языке C:

```
int main(int argc, char *argv[])
```

Для осуществления ввода и вывода данных в языке C++ используется библиотека `iostream`. Подключение библиотеки осуществляется также как и в языке C: используя директиву `#include`. Дополнительно необходимо определить пространство имен, которое будет

Программирование

использоваться в программе. Для этого сразу после подключения библиотеки `iostream` необходимо вставить строку `using namespace std`. После этих преобразований можно приступить к написанию остальной программы. Окончательный вид заготовки программы приведен на рисунке 9.

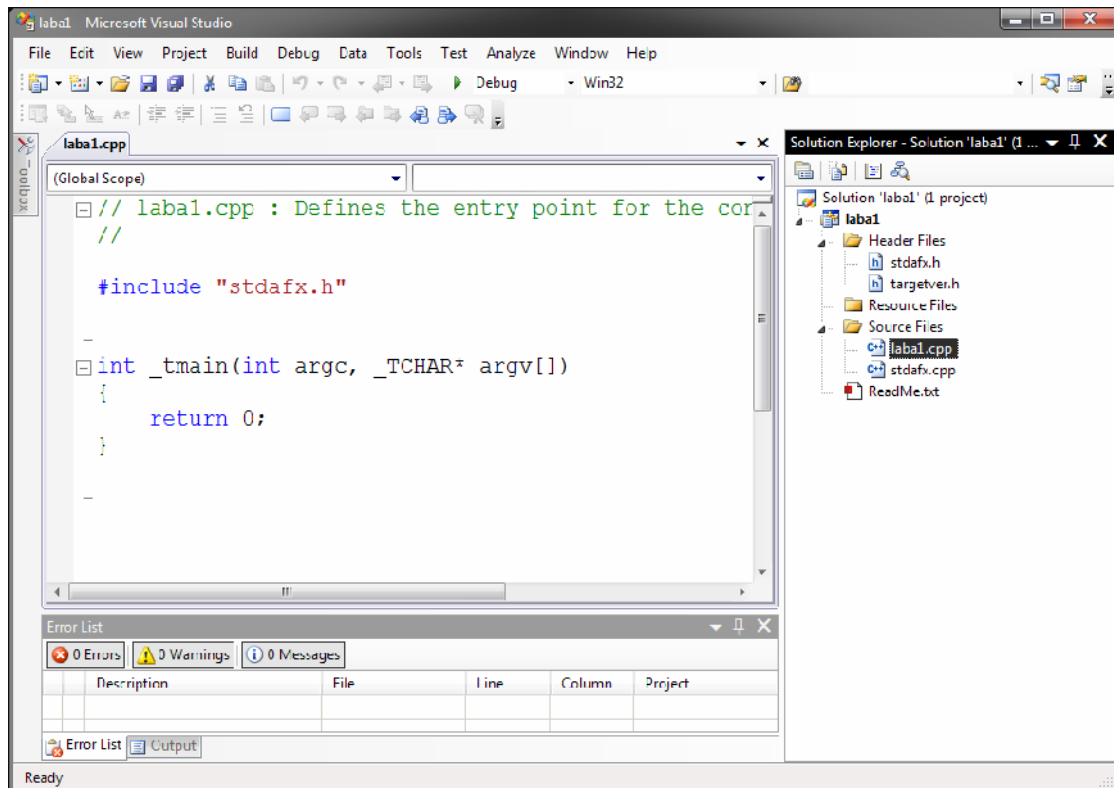


Рисунок 8 – Окно среды разработки после создания проекта

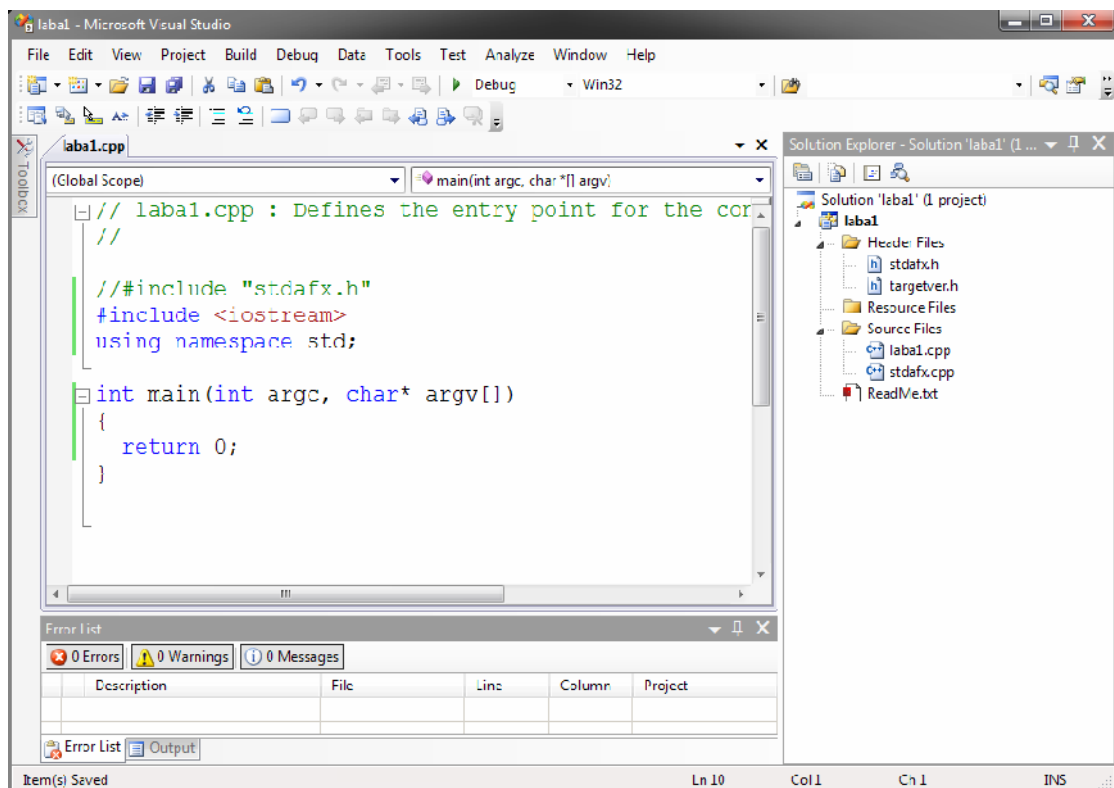


Рисунок 9 – Исправленный вид программы

После написания самой программы ее необходимо скомпилировать и запустить. Построение программы осуществляется с помощью элементов меню **Debug** главного меню программы. В этом меню содержатся следующие пункты:

- **Build Solution (F6)** – компиляция и сборка решения;
- **Rebuild Solution** – перекомпиляция и сборка решения;
- **Clean Solution** – очистка временных, объектных и т.п. файлов для данного решения;
- **Build <имя проекта> (Shift+F6)** – компиляция и сборка проекта;
- **Rebuild <имя проекта>** – перекомпиляция и сборка проекта;
- **Clean <имя проекта>** – очистка временных, объектных и т.п. файлов для проекта.

В среде разработки Visual C++ в рамках одного решения (solution) может быть несколько проектов.

Если программа написана с ошибками, то компилятор выдаст соответствующие сообщение в нижнем поле «**Error List**» главного окна на вкладке «**Errors**», предупреждения выводятся на вкладке «**Warnings**», а остальные сообщения на вкладке «**Messages**». Если компиляция программы прошла успешно, то можно осуществить запуск программы. Это действие осуществляется с помощью меню **Debug** главного окна среды разработки. В этом меню есть два пункта:

- **Start Debugging (F5)** – запустить программу в режиме отладки;
- **Start Without Debugging (Ctrl+F5)** – запустить программу без отладки.

Режим отладки обладает довольно широкой функциональностью. Все действия режима отладки доступны в меню **Debug** главного меню программы:

- **Continue (F5)** – продолжить выполнение до следующей точки останова или конца программы;
- **Stop Debugging (Shift+F5)** – завершить отладку;
- **Restart (Ctrl+Shift+F5)** – перезапустить программу;
- **Step Into (F11)** – вход внутрь функции или переход к следующей строке;
- **Step Over (F10)** – переход через функцию или переход к следующей строке;
- **Step Out (Shift+F11)** – выход из функции (без дальнейшей ее трассировки).

Установка точек останова осуществляется командой **Breakpoint → Insert Breakpoint** контекстного меню редактора текста программы, удаление точки останова осуществляется командой **Breakpoint → Delete Breakpoint**, отмена (запрет) точки останова осуществляется командой **Breakpoint → Disable Breakpoint**.

Во время отладки в нижней части экрана выводятся окна, содержащие отладочную информацию. Так по умолчанию там присутствуют вкладки:

- **Locals** – содержит список локальных переменных и их текущие значения;
- **Watch 1** – содержит список переменных (и их текущих значений), указанных пользователем.

Добавление переменной в список **Watch** окна отладочной информации осуществляется командой **Add** контекстного меню данного окна или путем непосредственного ввода имени переменной в свободную ячейку этой вкладки.

Ход работы

Запустите среду разработки «**Visual C++**» и проведите необходимые настройки, указанные в краткой теории.

Создайте проект под именем **lab1** в своем рабочем каталоге. Рабочий каталог должен располагаться на диске **D:** в каком-нибудь каталоге (его можно создать самостоятельно). В качестве имени рабочего каталога лучше всего использовать свою фамилию, записанную латинскими символами без пробелов и символов пунктуации.

Программирование

Дана программа, осуществляющая ввод целочисленного массива, поиск в нем максимума и вывод его значения и позиции на экран. Введите (предпочтительно) или скопируйте данную программу в файл main.c проекта. Откомпилируйте и запустите, просмотрите на ее работу.

```
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
    int n;
    cout << "Puts size of array: ";
    cin >> n;
    //Выделение динамической памяти под массив в стиле языка C++
    int *a = new int[n];
    cout << "Puts array: ";
    for(int i=0;i<n;i++) cin >> a[i];
    int max = a[0], pos = 0;
    for(int i=1;i<n;i++){
        if(max<a[i]){
            max = a[i];
            pos = i;
        }
    }
    cout << "Value of maximum: " << max
         << "\nhis position: " << pos << endl;
    //Освобождение динамически выделенной памяти в стиле C++
    delete [] a;
    return 0;
}
```

Запустите программу в режиме отладки. Выполните программу в режиме пошагового исполнения. Установите точку останова на начало поиска максимума в массиве и запустите программу в обычном режиме. Посмотрите на результат. После достижения точки останова программы на вкладке Watch отобразите переменные max, pos, i и a.

Модифицируйте программу для решения следующей задачи: дан массив целых чисел размера N, организовать ввод массива, найти максимум и минимум массива и поменять их местами. Полученный массив вывести на экран. Над новой программой проведите те же действия, что и для примера.

Создайте новый проект в рамках открытого решения. Напишите программу согласно варианту вашего задания. Выполните над ней все действия по построению, запуску и отладке. Продемонстрируйте полученные навыки преподавателю.

Варианты заданий

1	Переписать в обратном порядке элементы целочисленного массива, расположенные между I-ым и J-ым элементами. Значения I, J вводит пользователь.
2	Упорядочить элементы символьного массива так, чтобы с начала располагались буквы, а затем цифры. Другие символы в массиве отсутствуют.
3	Сдвинуть циклически элементы вещественного массива вправо на K позиций. Число K вводит пользователь.
4	Определить в целочисленном массиве серии минимальной и максимальной длины и поменять их местами. Если таких серий несколько, то только первые из них.
5	Поменять в символьном массиве элементы местами так, чтобы чередовались большие и маленькие буквы. Предполагается, что в массиве равное количество больших и маленьких букв, а других символов нет.

6	Отсортировать элементы вещественного массива в порядке возрастания их дробной части.
7	Упорядочить элементы целочисленного массива так, чтобы сначала располагались нулевые значения, затем отрицательные, а потом положительные.
8	Сдвинуть циклически влево символьный массив так, чтобы из начала в конец массива было перенесено K цифр. Значение K вводит пользователь.
9	Переписать в обратном порядке элементы вещественного массива, расположенные между минимумом и максимумом массива. Все элементы массива различны.
10	Упорядочить все серии в целочисленном массиве в порядке убывания их длины.
11	Переставить элементы символьного массива так, чтобы сначала располагались буквы и цифры, а затем знаки препинания. Порядок следования самих элементов не менять.
12	Поменять попарно отрицательные и положительные элементы в вещественном массиве местами. Количество отрицательных и положительных значений в массиве одинаково.
13	Переставить в целочисленном массиве элементы местами так, чтобы сначала располагались нечетные значения, а затем четные.
14	Упорядочить символьный массив так, чтобы сначала в нем располагались цифры в порядке убывания, а затем буквы в алфавитном порядке.
15	Отсортировать по возрастанию элементы вещественного массива, расположенные на нечетных позициях. Остальные элементы оставить без изменения.
16	Сдвинуть циклически вправо целочисленный массив так, чтобы сумма перенесенных из конца в начало элементов превысила значение K, которое вводит пользователь.
17	Упорядочить символьный массив так, чтобы сначала в нем располагались буквы в обратном алфавитном порядке, а затем цифры в порядке возрастания.
18	Заменить все отрицательные элементы вещественного массива значением минимального положительного элемента всего массива.
19	Поменять местами I-ую и J-ую серии целочисленного массива. Значения J и I вводит пользователь. Если таких серий нет, то ничего не менять.
20	В символьном массиве все элементы, расположенные на четных позициях, перенести в 1-ую половину массива, а на нечетных позициях – во 2-ую половину.
21	В вещественном массиве переписать в обратном порядке все положительные элементы. Остальные элементы массива оставить на прежних местах.
22	В целочисленном массиве отсортировать элементы в порядке убывания сумм их цифр.
23	Переставить элементы символьного массива так, чтобы сначала в нем шли знаки препинания, затем цифры и, в конце, буквы латинского алфавита.
24	Если сумма элементов вещественного массива положительна, то отсортировать массив по возрастанию, в противном случае по убыванию.
25	В целочисленном массиве переписать все четные значения в обратном порядке. Остальные элементы оставить без изменений.
26	Если в символьном массиве букв больше чем цифр, то перенести их в начало массива. В противном случае в начало массива перенести цифры. Порядок следования не менять.
27	Заменить все положительные элементы вещественного массива значением максимального отрицательного элемента.
28	Сдвинуть циклически элементы целочисленного массива влево на K позиций. Число K вводит пользователь.
29	В символьном массиве заменить все маленькие буквы на большие, а большие на маленькие. Остальные элементы оставить без изменений.
30	Переписать в обратном порядке элементы вещественного массива, расположенные между максимальным отрицательным и минимальным положительным значениями в массиве.

Контрольные вопросы

1. Для разработки каких приложений предназначена IDE Visual Studio .NET?
2. Какие языки программирования поддерживаются в Visual Studio .NET?

Программирование

3. Где осуществляется настройка месторасположения проектных файлов в Visual Studio?
4. Где осуществляется настройка параметров шрифта в Visual Studio .NET?
5. Как в MS Visual C++ осуществляется создание нового проекта?
6. Как в MS Visual C++ осуществляется построение исполняемого модуля программы?
7. Как в MS Visual C++ осуществляется запуск программы в режиме отладки?
8. Как в MS Visual C++ осуществляется запуск программы без режима отладки?
9. Какие возможности отладки присутствуют в MS Visual C++?
10. Как в MS Visual C++ осуществляется управление точками останова?