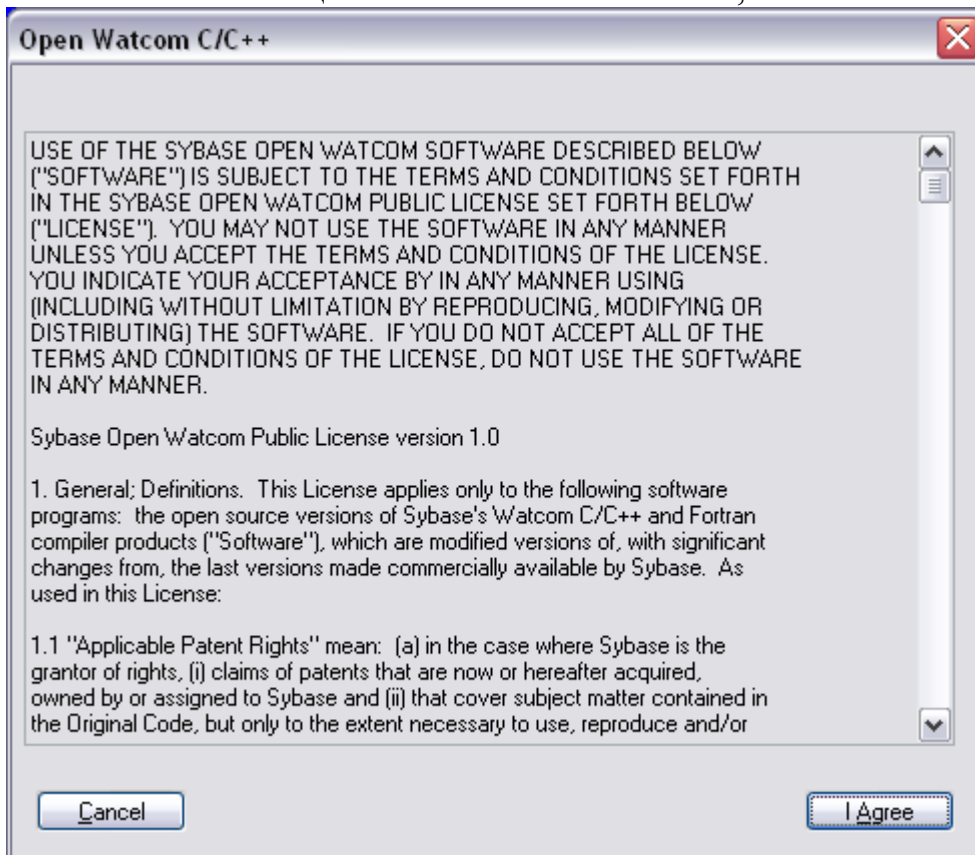
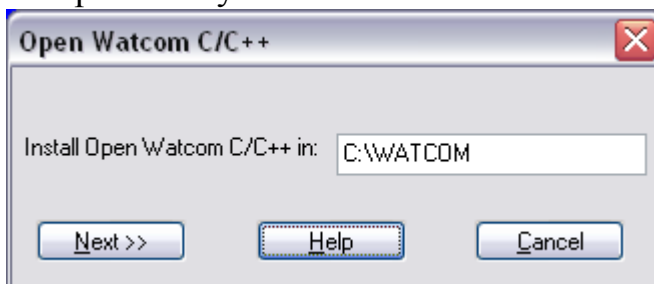


## Установка компилятора Open Watcom C/C++ 1.6

1. Запустите файл `open-watcom-c-win32-1.6.exe`
2. Согласитесь с лицензионным соглашением, нажав кнопку **I Agree**

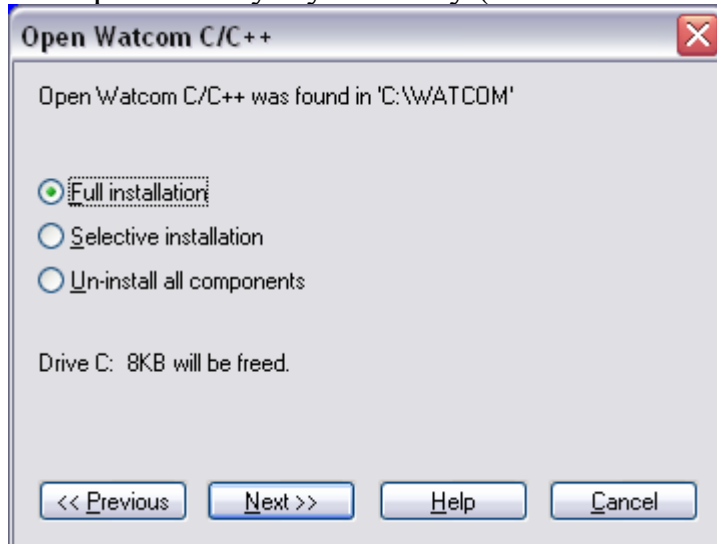


3. Введите полное имя папки, в которую хотите установить компилятор и нажмите кнопку **Next >>** (можно использовать имя по умолчанию, которое уже есть в окошке – `C:\WATCOM`).

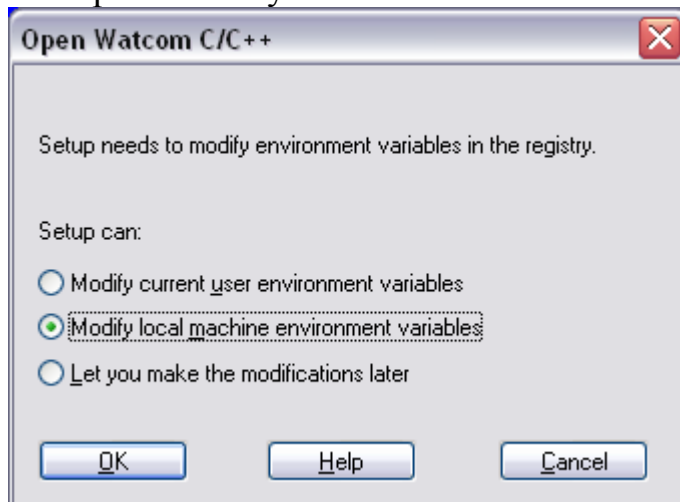


**ВНИМАНИЕ!** Не используйте русские буквы и пробелы в имени папки!

4. Выберите полную установку (Full installation), нажмите Next>>



5. Выберите Modify local machine environment variables, нажмите ОК



6. После завершения установки перезагрузите компьютер  
7. Выведите ярлык интегрированной среды разработки на рабочий стол.  
Оригинал находится в *Пуск* → *Программы* → *Open Watcom C\_C++* → *IDE*. Среду разработки можно запустить и из командной строки, выполнив команду `ide`

**Примечание:** для установки компилятора требуются права администратора. Для того, чтобы можно было работать обычным пользователям, создайте рабочую папку, полное название которой не содержит пробелов и русских букв и в которую можно писать (например, `C:\C_WORK`. Возможна работа с сетевыми папками, что «отвязывает» ученика от конкретного ПК)

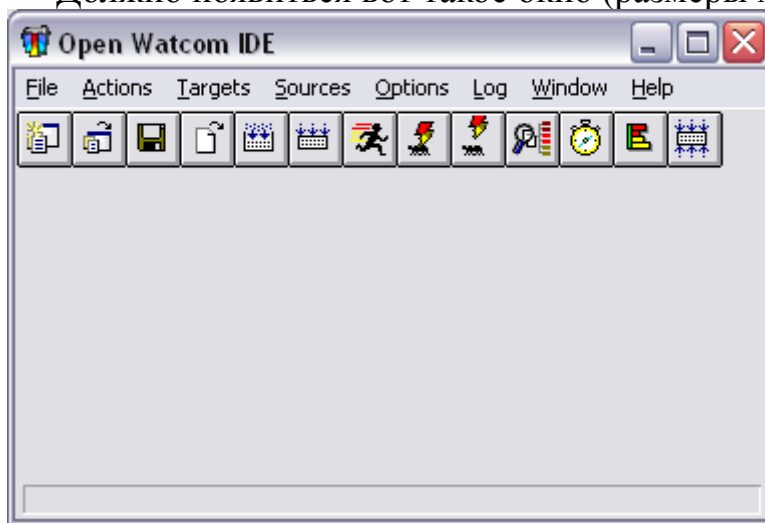
## Урок 1. Знакомство со средой и создание первой программы


### Создание личных папок

1. Откройте папку C : \C\_WORK и создайте свою личную папку (например, Ivanov\_AV; детальные указания по тому, как её назвать, получите у преподавателя).  
**Русские буквы и пробелы использовать нельзя!**
2. В личной папке создайте папку Lesson1.

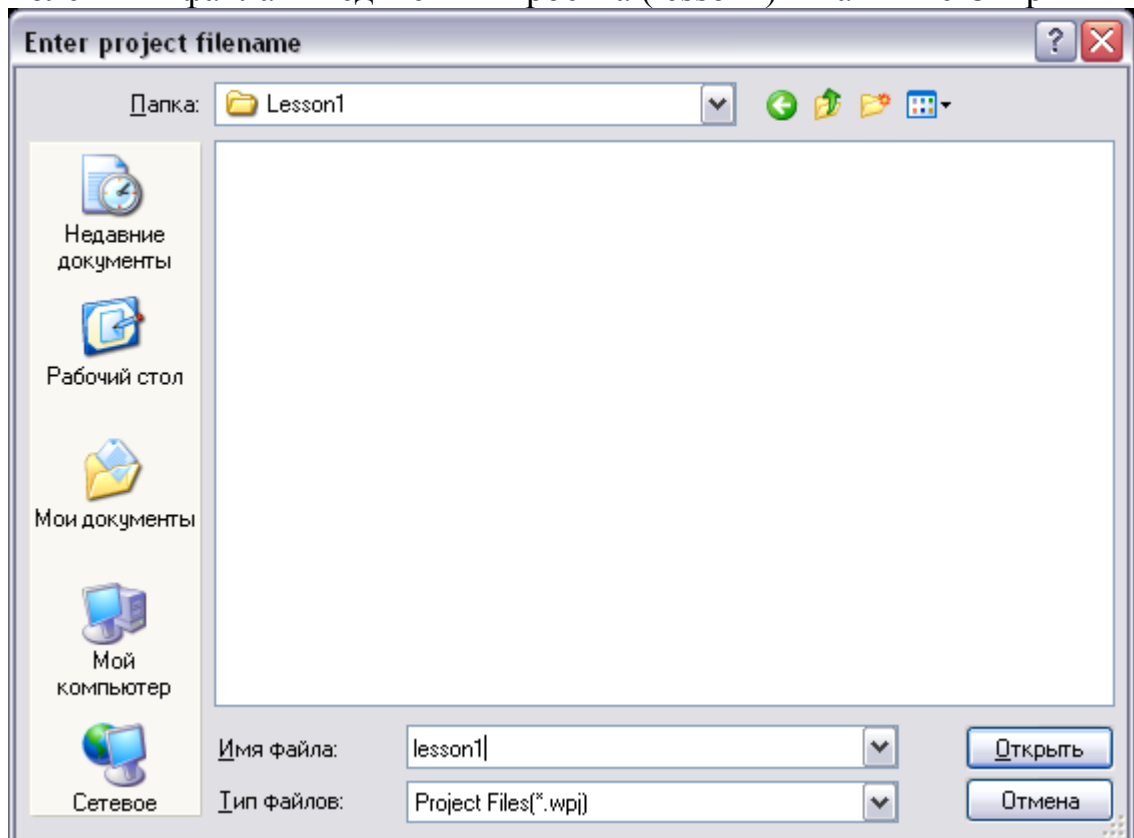
### Создание проекта

1. Запустите среду разработки (пиктограмма с молотком и подписью IDE).  
Должно появиться вот такое окно (размеры могут быть другими)

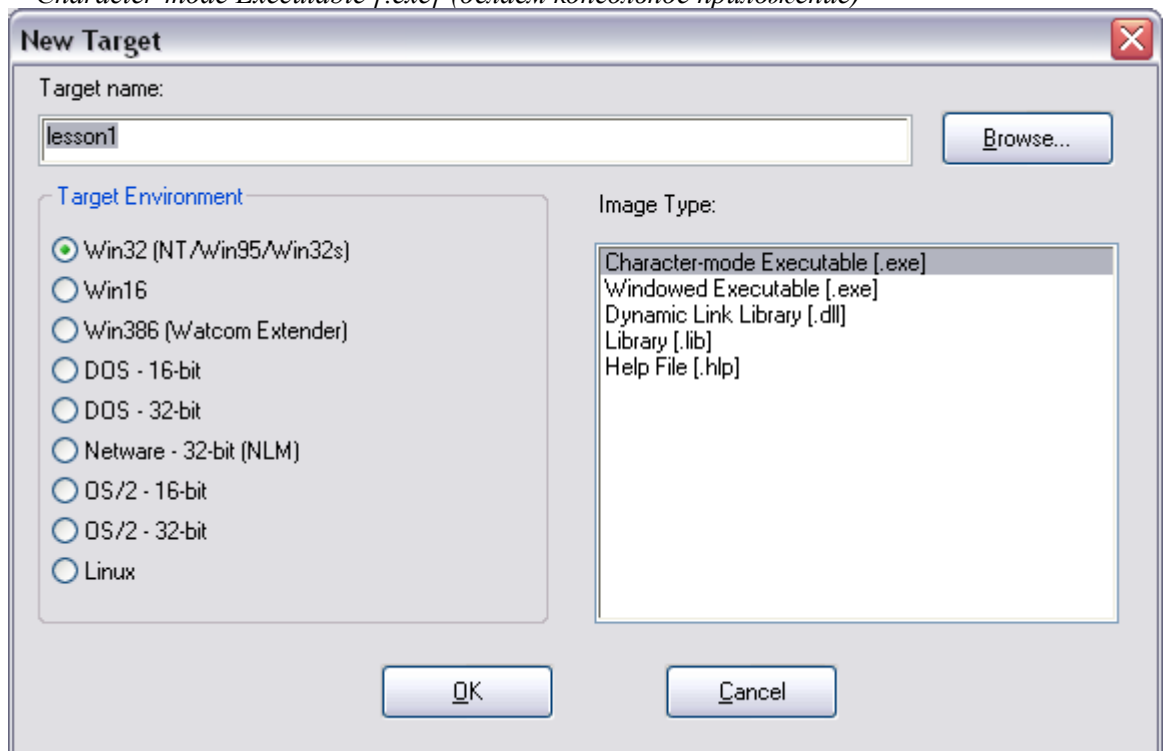


2. Начните создание нового проекта - *File*→*New Project* (или щелчок по кнопке  или «горячая» клавиша **CTRL+N**)

3. В открывшемся диалоге откройте папку C:\C\_WORK\Lesson1, в поле «имя файла» введите имя проекта (lesson1) и нажмите Открыть

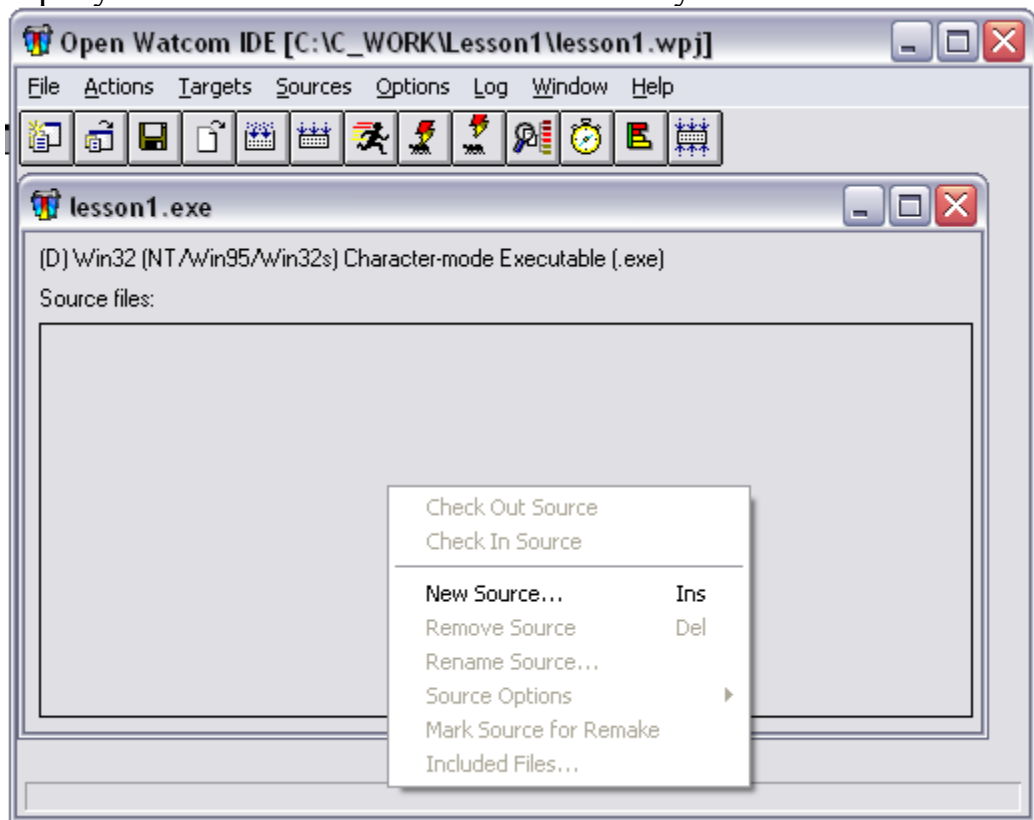


4. В появившемся диалоге ничего не меняйте и нажмите **ОК**.  
*Примечание: Target Environment = Win32 (программируем под Windows 95/NT), Image Type = Character-mode Executable [.exe] (делаем консольное приложение)*

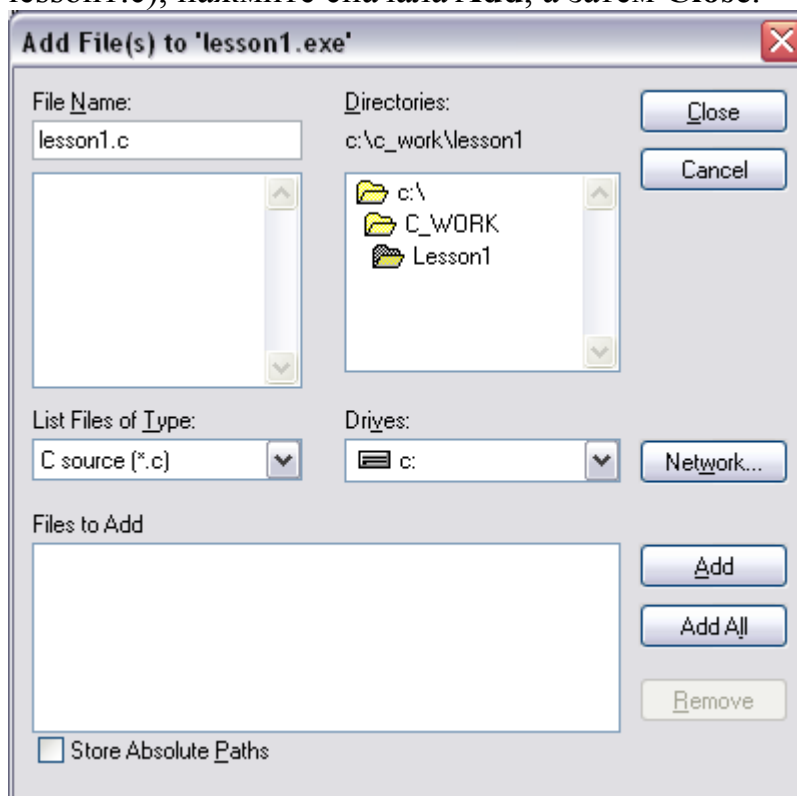


5. Появился проект, который пока не содержит ни одного файла с программным кодом. Для добавления файла с кодом щёлкните правой кнопкой мыши по пустой чёрной рамочке под надписью *Source Files* и

выберите *New Source* (см. рисунок). Вместо мыши можно нажать горячую клавишу **Ins**



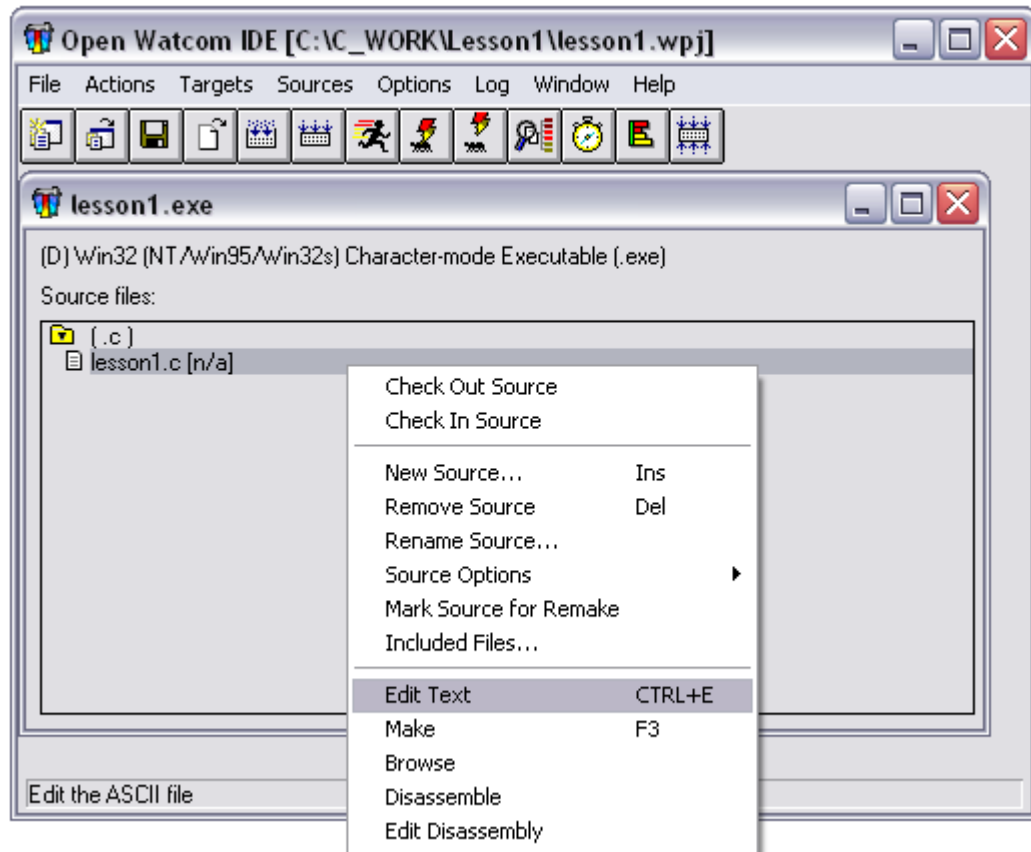
6. Введите имя нового файла вместе с расширением (в нашем случае это lesson1.c), нажмите сначала **Add**, а затем **Close**.




## Ввод первой программы

Проект создан, теперь перейдём к созданию исходного текста простейшей программы – вывод строки текста на экран.

1. Сделайте двойной щелчок по файлу lesson1.c. Можно также воспользоваться правым щелчком мыши и контекстным меню (см. рисунок)

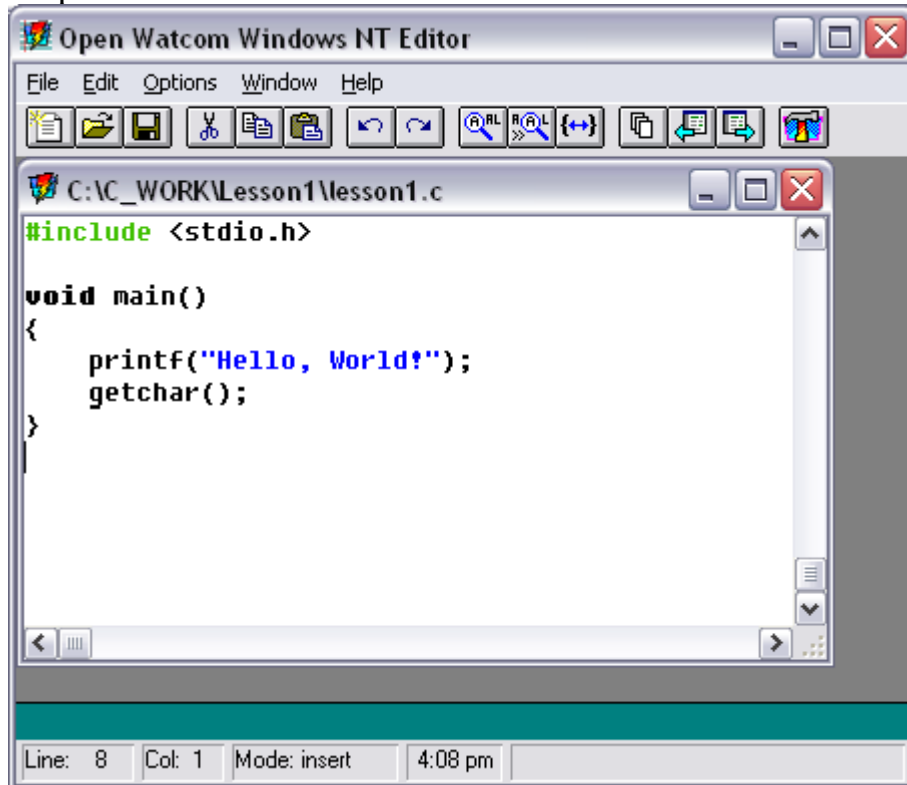


2. В окне появившегося редактора набейте текст программы. При наборе помните, что в языке Си **строчная и прописная буква – это разные символы**. Сохраните его на диск (*File*→*Save*,  или **CTRL+S**) и

закройте

ОКНО

редактора



Разберём подробнее текст программы

```
#include <stdio.h>
```

Эта строка с директивой `include` означает использование стандартной библиотеки ввода-вывода. В этой программе она нужна для печати строки текста и ожидания нажатия клавиши.

```
void main()
```

Выполнение программы на Си начинается с функции `main()`. `void` означает, что эта функция ничего не возвращает (ср. с процедурой в Pascal или Qbasic).

```
{ }
```

Фигурные (операторные) скобки обозначают начало и конец тела функции.

```
printf("Hello, World!");
```

Функция `printf` выводит данные на экран. В этой программе она выводит строку `Hello, World!`

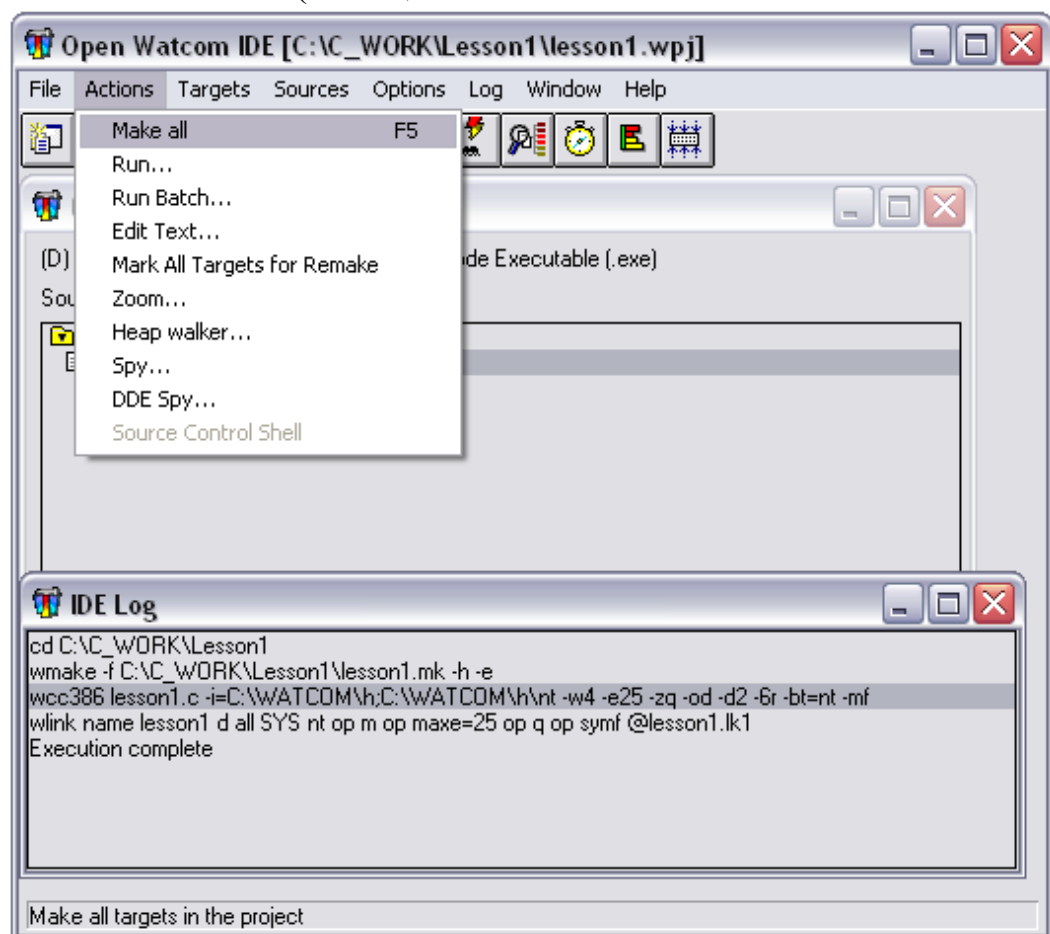
```
getchar();
```

Функция дожидается ввода символа с клавиатуры

### Компиляция и выполнение программы

Для того чтобы выполнить программу на Си, её необходимо откомпилировать – перевести набранный текст в исполняемый файл.

1. Откомпилируйте программу меню: *Actions*→*Make all* (или просто нажав **F5**). В окне IDE Log можно посмотреть результат. Если всё набрано правильно, то компиляция пройдёт без единой ошибки (см. скриншот), в противном случае появятся строки, начинающиеся с **Error** (сообщения об ошибках).



2. Запустите программу через меню *Targets*→*Run* (или через **CTRL+R**). Должно появиться примерно такое окно:



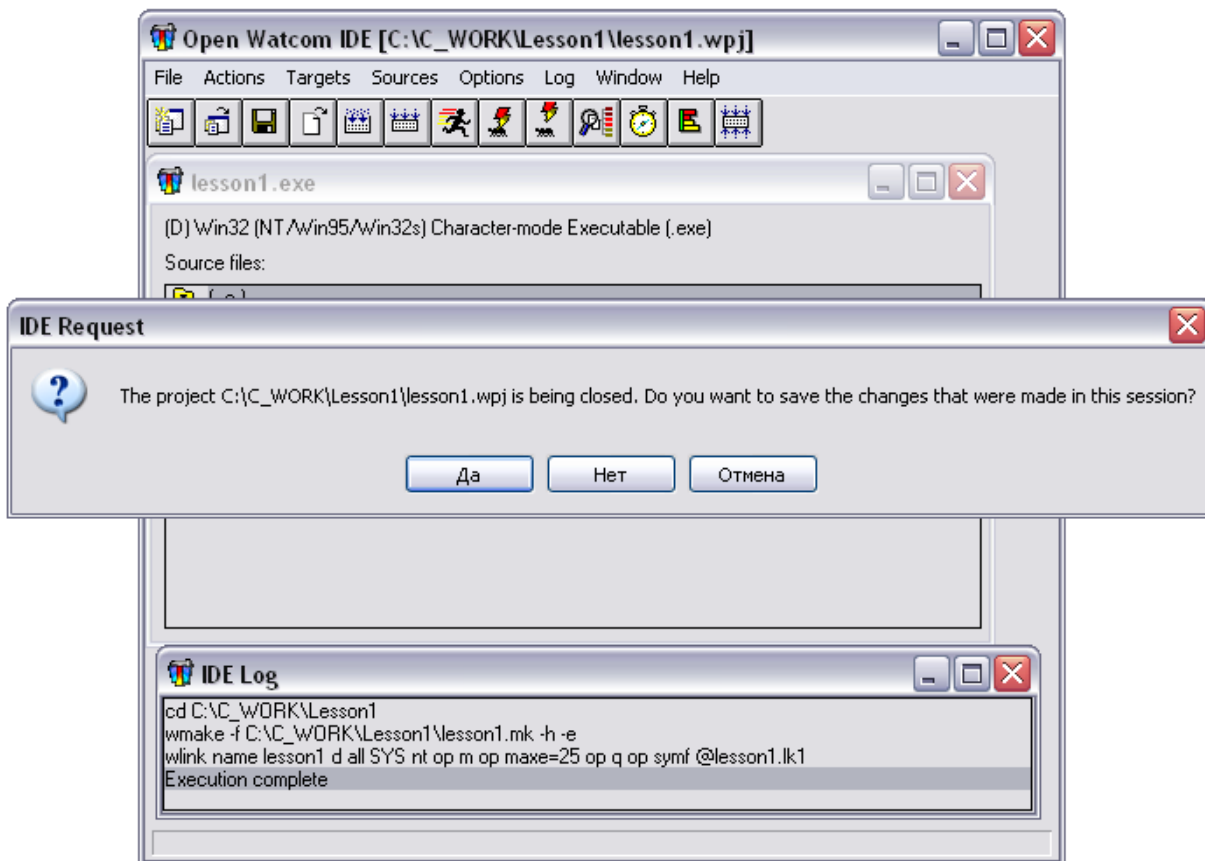


Для того чтобы оно исчезло, нажмите Enter. Можно закрыть его крестиком или по ALT+F4

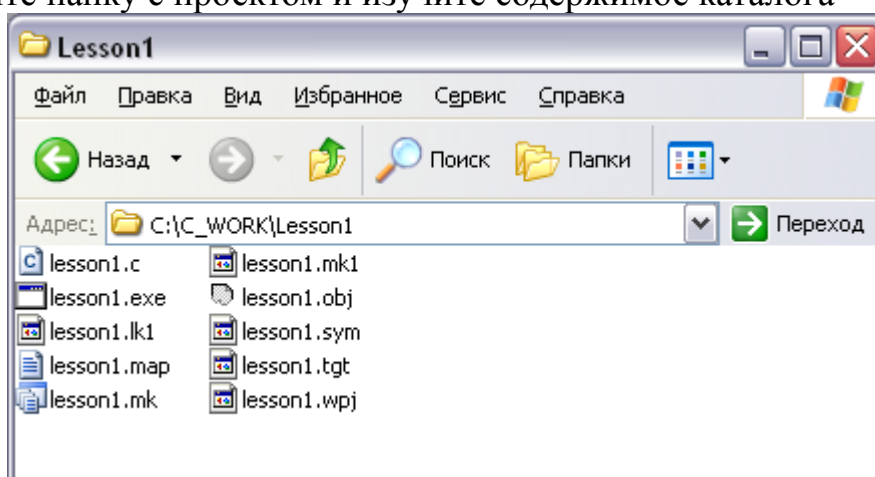
**Поздравляем с успешной компиляцией  
первой программы на Си!**

### Завершение работы и подведение итогов

Для завершения работы со средой закройте окно Open Watcom IDE. Ответьте «Да» на появившийся вопрос о том, сохранять ли проект. После этого окно среды исчезнет, а проект будет сохранён на диске.



Теперь давайте посмотрим на результаты работы компилятора. Для этого откройте папку с проектом и изучите содержимое каталога



Основным результатом работы интегрированной среды является файл `lesson1.exe`. Это самый обычный исполняемый файл («EXE-шник»), который может быть запущен на любой машине с Windows. Он очень компактный (всего 30Кб) и не требует для своей работы компилятора или каких-то сторонних библиотек. Так что его смело можно унести домой на дискете и там запустить.

В файле `lesson1.c` находится исходный текст программы.

Проект хранится в файлах `lesson1.wpj` и `lesson1.tgt`

Все остальные файлы – промежуточные и их не обязательно сохранять.

Если Вы захотите программировать на Си дома, то компилятор Open Watcom можно найти в Интернете по адресу <http://openwatcom.org/>. Скачивание и использование – бесплатное.

### Приложение 1. Краткий словарь

<code>#include &lt;stdio.h&gt;</code>	Включить <b>Standart Input and Output</b> (стандартный ввод и вывод) .h = <b>header file</b> (заголовчный файл)
<code>void main ()</code>	пустота (т.е. ничего не возвращается) главные (т.е. главная функция, с которой начинается выполнение программы)
<code>{ }</code>	Операторные скобки
<code>printf( )</code>	<b>Formatted print</b> – форматная печать (на экране). Функция получила такое название за богатые возможности по выводу данных на экран практически в любом нужном формате. В программе Hello, World используется лишь малая доля её возможностей
<code>getchar()</code>	<b>Get character</b> – получить символ. (Ввод символов с клавиатуры)

### Приложение 2. Горячие клавиши

<b>CTRL+N</b>	Создать новый проект
<b>CTRL+O</b>	Открыть существующий проект
<b>F5</b>	Откомпилировать всю программу
<b>CTRL+S</b>	Сохранить проект или исходный текст программы
<b>CTRL+R</b>	Запустить откомпилированную программу
<b>ALT+F4</b>	Закрывать окно
<b>Insert</b>	Добавить новый файл к проекту
<b>Delete</b>	Удалить файл из проекта (но не с диска)

## Резюме

### Установка компилятора Open Watcom C/C++ 1.6

1. Запустите файл `open-watcom-c-win32-1.6.exe`
2. Согласитесь с лицензионным соглашением, нажав кнопку I Agree
3. Введите полное имя папки, в которую хотите установить компилятор и нажмите **Next >>** (можно использовать имя по умолчанию, которое уже есть в окошке)

**ВНИМАНИЕ!** Не используйте русские буквы и пробелы в имени папки!

4. Выберите полную установку (Full installation), нажмите **Next >>**
5. Выберите Modify local machine environment variables, нажмите **OK**
6. После завершения установки перезагрузите компьютер
7. Выведите ярлык интегрированной среды разработки на рабочий стол.  
Оригинал находится в Пуск → Программы →  
*Open Watcom C\_C++ → IDE.*


**Примечание:** для установки компилятора требуются права администратора. Создайте рабочую папку, полное название которой не содержит пробелов и русских букв и в которую можно писать (например, `C:\C_WORK`). Возможна работа с сетевыми папками.

## Урок 1. Знакомство со средой и создание первой программы


### Создание личных папок

1. Откройте папку C : \C \_WORK и создайте свою личную папку (например, Ivanov\_AV). **Русские буквы и пробелы использовать нельзя!**
2. В личной папке создайте папку Lesson1

### Создание проекта

1. Запустите среду разработки
2. Начните создание нового проекта - *File*→*New Project* (или щелчок по кнопке  или **CTRL+N**)
3. В открывшемся диалоге откройте папку C:\C\_WORK\Lesson1, в поле «имя файла» введите имя проекта (lesson1) и нажмите **Открыть**
4. В появившемся диалоге ничего не меняйте и нажмите **ОК**.
5. К пустому проекту добавьте файл с кодом: щёлкните правой кнопкой мыши по пустой чёрной рамочке под надписью *Source Files* и выберите *New Source* (или нажмите **Ins**).
6. Введите имя нового файла вместе с расширением (lesson1.c), нажмите сначала **Add**, а затем **Close**.

### Ввод первой программы

1. Сделайте двойной щелчок по файлу lesson1.c.
2. В окне появившегося редактора наберите текст программы. **Строчная и прописная буква – это разные символы.** Сохраните его на диск (*File*→*Save*,  или **CTRL+S**) и закройте окно редактора

```
#include <stdio.h>

void main()
{
    printf("Hello, World!");
    getchar();
}
```

### Компиляция и выполнение программы

Для того чтобы выполнить программу на Си, её необходимо откомпилировать – перевести набранный текст в исполняемый файл. Результат компиляции и сообщения об ошибках можно найти в окне IDE Log.

Откомпилируйте программу меню: *Actions*→*Make all* (или **F5**). Если откомпилировалось без ошибок, то запустите программу через меню *Targets*→*Run* (или **CTRL+R**). Появится окно с надписью “Hello, World!”.