.

1. С помощью команды Исходный файл подменю команды **Создать в меню Файл**;

·  нажав кнопку Исходный файл ()на панели инструментов;



·  нажав клавиши Ctrl+N.

При этом в области редактора появляется новая вкладка с пустым содержимым – Безымянный1, Безымянный2 и т. д.

Набор и редактирование текста программы в Dev-C++ выполняется так же, как и в приложении Блокнот. Для вырезания, копирования и вставки фрагментов программы можно использовать команды Вырезать, Копировать и Вставить меню Правка, либо клавиши Ctrl+X, Ctrl+C Ctrl+V. Если текст программы был изменен, но изменения не сохранены, перед именем файла в названии вкладки выводятся символы "[\*]".

При сохранении вновь созданного файла (с помощью команды Сохранить как меню Файл) рекомендуется задавать имя файла латинскими буквами. При сохранении файла в окне Сохранить файл следует в раскрывающемся списке Тип файла задать тип C source files (\*.c).

Повторно сохранить отредактированный файл можно с помощью одной из следующих операций:

·  выполнить команды Сохранить меню Файл;

·  нажать кнопку Сохранить () на панели инструментов;



·  нажать клавиши Ctrl+S.

**Открытие** **файла** с программой выполняется одним из следующих способов:

·  с помощью команды Открыть проект или файл меню Файл;

·  при нажатии кнопки Открыть проект или файл () на панели инструментов;



·  при нажатии клавиш Ctrl+O.

**Закрыть файл** с программой можно:

·  с помощью команды Закрыть меню Файл;

·  нажатием клавиш Ctrl+F4.

Простую одномодульную программу в Dev-C++ можно сразу откомпилировать, скомпоновать и выполнить одним из следующих способов:

помощью команды **Скомпилировать и выполнить** меню Выполнить;

·  нажав кнопку Скомпилировать и выполнить () на панели инструментов;



·  нажав клавишу F9.

**Отдельно компиляция** программы выполняется:

·  с помощью команды Скомпилировать меню Выполнить;

·  при нажатии кнопки Скомпилировать () на панели инструментов;



·  при нажатии клавиш Ctrl+F9,

а **выполнение программы**:

·  с помощью команды Выполнить меню Выполнить;

·  при нажатии кнопки Выполнить () на панели инструментов;



·  при нажатии клавиш Ctrl+F10.

Если при компиляции обнаружены ошибки, в окне Компилятор выводятся соответствующие сообщения:



Если при компиляции и компоновке не обнаружено ошибок, в Dev-C++ открывается окно выполнения программы как окно приложения MS DOS:

После выполнения оператора return окно выполнения программы автоматически закрывается и часто результат выполнения программы увидеть нельзя.

Чтобы этого не происходило, можно перед оператором return поставить оператор

**1-й способ**; getchar();,

который вызывает функцию ввода символа. Эта функция, так же как и функция printf(), находится в разделе stdio. h стандартной библиотеки C. В этом случае окно выполнения приложения MS DOS закрывается при вводе любого символа (или просто нажатия клавиши Enter).

**2-й способ** : system("pause")

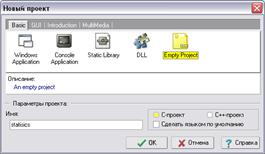
**2. Проект** – это набор файлов, содержащих программные модули, которые рассматриваются как одна программа.

Создать новый проект можно либо с помощью команды Проект подменю команды Создать меню Файл, либо нажав кнопку Проект () на панели инструментов.



После этого открывается диалоговое окно нового проекта

После этого открывается диалоговое окно нового проекта:



Во вкладке Basic этого окна необходимо выбрать вид проекта: приложение Windows (Windows Application), консольное приложение (Console Application), статическая библиотека (Static Library), библиотека динамических связей (DLL) для Windows или пустой проект (Empty Project). Для создания многомодульных программ с выводом в окно MS DOS лучше выбрать вид Empty Project. Затем в области Параметры проекта задается имя проекта (латинскими буквами) и тип проекта (C-проект). После нажатия кнопки OK выбирается папка, в которой будет сохранен файл с данными о проекте (с расширением. dev).

После этого в левом окне IDE во вкладке Проект появляется имя проекта. С помощью контекстного меню для проекта можно выполнить следующие операции:

·  создать новый файл (команда Создать файл);

·  добавить существующий файл программного модуля к проекту (команда Добавить к проекту);

·  удалить файл из проекта – файл при этом не уничтожается (команда Удалить из проекта);

·  добавить папку в проект (команда Добавить папку);

·  настроить параметры проекта (команда Параметры проекта).

В проекте должен быть хотя бы один файл. После создания или добавления всех модулей проекта:



можно **запускать проект на компиляцию и выполнение, используя те же команды и кнопки, что и для одномодульных программ.** При этом будут повторно компилироваться только те файлы проекта, в которых проведены изменения. Загрузочный файл для проекта имеет то же имя, что и имя проекта, но с расширением. exe.

.7. Справочная система Dev-C++

Справочная система вызывается с помощью команд меню Справка и организована по тому же принципу, что и все справочные системы для приложений Windows. С помощью команды Справка Dev-C++ можно получить сведения о работе в среде IDE (основных выполняемых операциях и командах меню) в разделе Dev-C++ 5, а в разделе An Introduction to C Programming содержатся краткие сведения о языке C.

Так как мы программируем под Windows, то нас будут интересовать такие кодировочные таблицы: **cp866, cp1251** и **utf-8**(стандарт Unicode).  Хотя уже давно разработан единый стандарт кодирования символов — **Unicode**, в Windows до сих пор используются несколько кодировочных таблиц, а именно — **cp866**, **cp1251.**Использование нескольких таблиц кодирования символов и является причиной появления козябликов, вместо сообщения Кириллица в консоли.

**Unicode** — единый стандарт кодирования символов, позволяющий представить знаки всех письменных языков.

Таким образом стандарт **Unicode**присваивает каждому символу уникальный код, независимо от языка. Сейчас **Unicode** считается лучшим стандартом кодирования символов. Вернёмся к нашей проблеме — вывод кириллицы в консоль.

|  |
| --- |
| // rus\_text.cpp: определяет точку входа для консольного приложения.   #include "stdafx.h"  #include <iostream>  #include <clocale>  using namespace std;    int main(int argc, char\* argv[])  {      setlocale(LC\_CTYPE, "rus"); // вызов функции настройки локали      cout << "Кириллица в консоли\n";      system("pause");      return 0;  } |

Функция setlocale() имеет два параметра, первый параметр — тип категории локали, в нашем случае LC\_TYPE — набор символов, второй параметр — значение локали. Таким образом, сначала устанавливаем нужную локаль, в нашем случае — "rus", после чего, можно использовать кириллицу, для вывода сообщений в консоль.

Вместо аргумента "rus" можно также писать "Russian", или оставлять пустые двойные кавычки, тогда набор символов будет такой же как и в ОС.

На первый взгляд функция setlocale()делает своё дело, но это только на первый взгляд. Суть в том, что данная функция setlocale()работает только с потоком вывода. Но как же быть с потоком ввода??? Есть немало способов решать данную проблему, мы рассмотрим лишь один из них. Для этого подключаем к проекту заголовочный файл <windows.h>. В файле содержатся прототипы функций SetConsoleCP() и SetConsoleOutputCP(), они-то нам и нужны. Аргументом этих функций является идентификатор кодовой страницы, нужная нам страница **win-cp 1251**. Функция SetConsoleCP()устанавливает нужную кодовую таблицу, на поток ввода, тогда как функция SetConsoleOutputCP()устанавливает нужную кодовую таблицу, на поток вывода.

Есть один весомый недостаток у данных функций — они работают со шрифтом **Lucida Console**. По умолчанию в консоли стоит шрифт **Consolas**. Поэтому в командной строке необходимо настроить используемый шрифт. Для этого открываем**cmd**, заходим в свойства, в закладке **Шрифт**выбираем **Lucida Console**, нажимаем **ОК**и на этом настройка завершена (см. Рисунок 5).

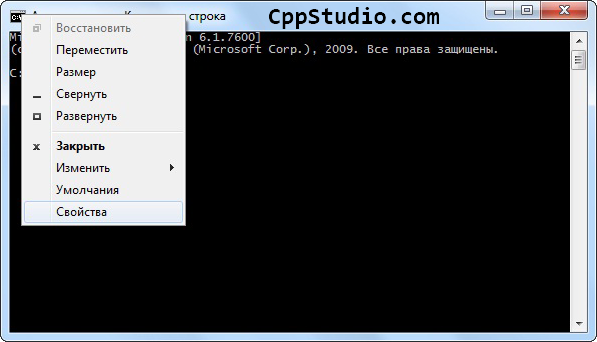


Рисунок 5 — Кириллица в консоли

Если все сделали правильно, то в консоли корректно будут отображаться символы Кириллицы. Теперь вместо функции setlocale()воспользуемся этими функциями.

[?](http://cppstudio.com/post/435/)

|  |  |
| --- | --- |
|  |  |

// код Dev-C++

[?](http://cppstudio.com/post/435/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | // rus\_text.cpp: определяет точку входа для консольного приложения.    #include <iostream>  #include <Windows.h>  using namespace std;    int main(int argc, char\* argv[])  {      SetConsoleCP(1251);// установка кодовой страницы win-cp 1251 в поток ввода      SetConsoleOutputCP(1251); // установка кодовой страницы win-cp 1251 в поток вывода        char string[20];      cin >> string; // вводим строку, используя Кириллицу      cout << "\nвывод: "<< string << endl; // ввывод строки      return 0;  } |

**Локаль** — это набор параметров: набор символов, язык пользователя, страна, часовой пояс и др. Локаль необходима для быстрой настройки пользовательского интерфейса, в зависимости от географического положения. В С++ есть функция setlocale(), которая выполняет перекодировку символов в соответствии с требуемым языком. Эта функция определена в заголовочном файле <clocale>. Переделаем программу, которая передает сообщение Кириллица в консоли в командную строку windows.