Практическое занятие 1

Тема: Написание и компиляция консольного приложения в среде разработки

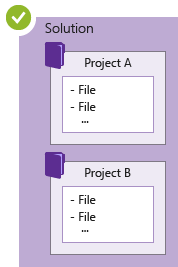
Цель работы: Знакомство со средой разработки, создание проекта и написание консольного приложения.

**Теоретическая справка. Инструмент разработчика**

1. Visual Studio – Это комплексная интегрированная среда разработки (IDE), которую можно использовать для записи, редактирования, отладки и сборки кода, а затем развертывания приложения. Помимо редактирования и отладки кода Visual Studio включает компиляторы, средства завершения кода, управление версиями, расширения и многое другое, чтобы улучшить каждый этап процесса разработки программного обеспечения.
2. Редактор кода
3. Обозреватель решений. Решения – это просто контейнеры для упорядочения одного проекта или нескольких связанных. Проект содержит программный код, ресурсы, ссылки на включаемые библиотеки и компоненты, необходимые для создания программы. Каждый проект содержит несколько файлов и папок. Решения и проекты можно просматривать в окне инструментов, которое называется ***обозревателем решений***. Полужирным шрифтом выделено имя запускаемого проекта. Этот проект выполняется при запуске приложения.

У проектов есть свойства, которые определяют различные аспекты компиляции, отладки, тестирования и развертывания. Некоторые свойства являются общими для всех типов проектов, а некоторые – уникальными для конкретных языков или платформ.

Доступ к свойствам проекта можно получить, щелкнув правой кнопкой мыши узел проекта в обозревателе решений и выбрав "Свойства" или введя "свойства" в поле поиска в строке меню и выбрав окно свойств из результатов.



Решения и проекты

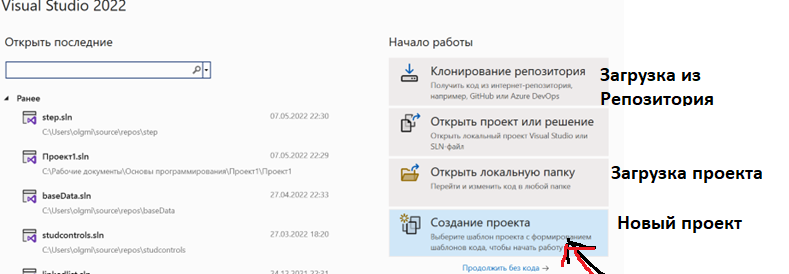
**Задания и указания по их выполнению**

**Задание 1. Создание проекта в среде MVS**

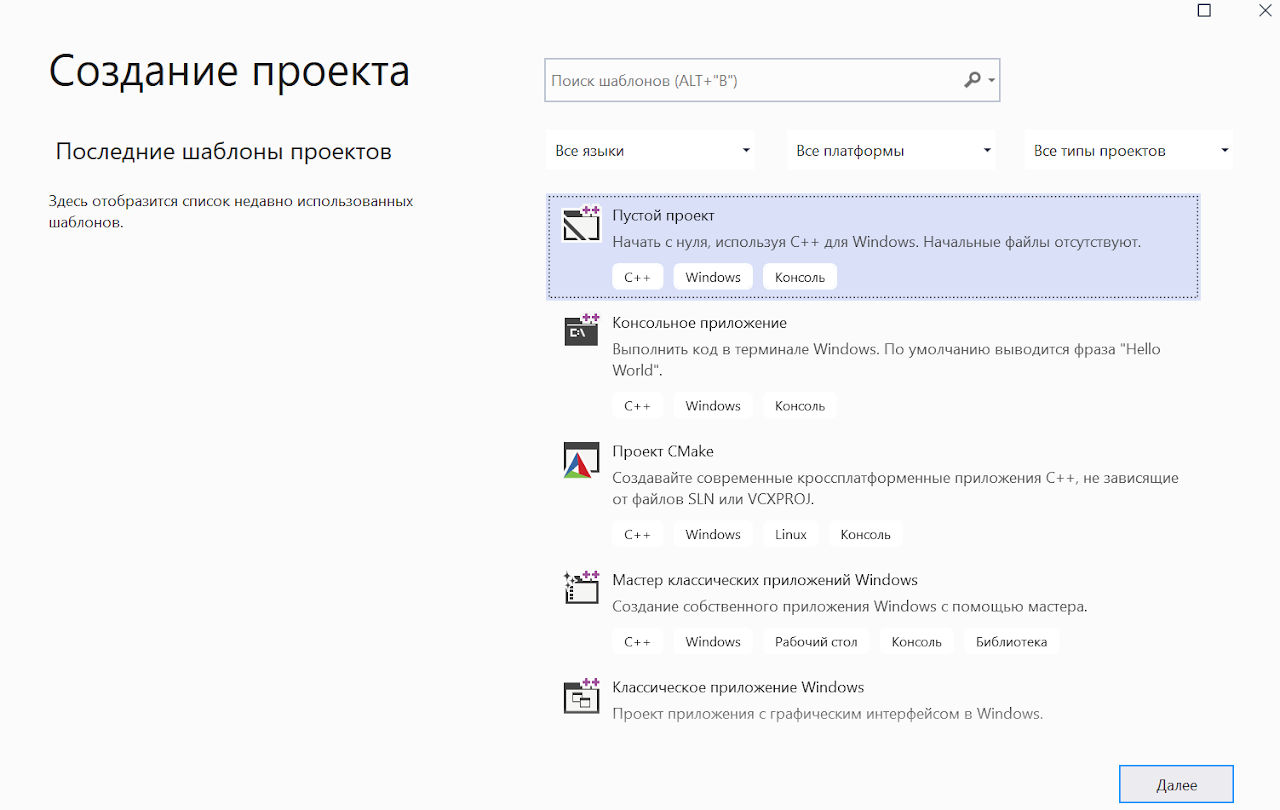
1. Запустите среду разработки.
2. Создайте проект Lab0, следуя инструкции. И запустите свое первое приложение.

## 1. Создание проекта

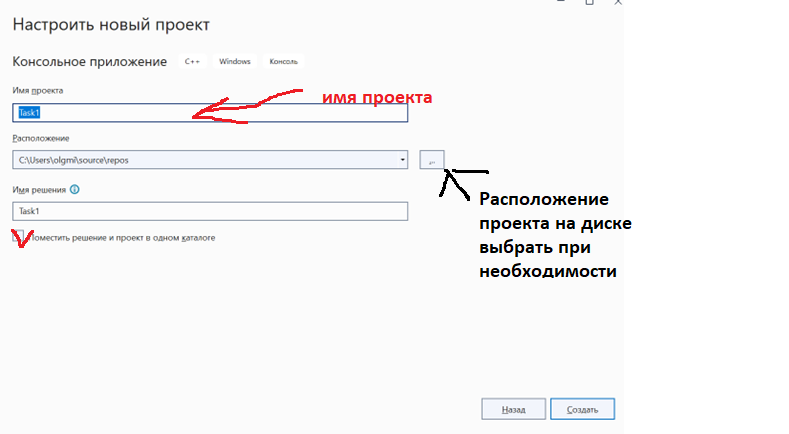
Для создания проекта выберите в меню **Файл** пункты **Создать** и **Проект**.



В области типов проектов **Visual C++** выберите группу **Общие** и щелкните элемент **Пустой проект**.



### Настройка проекта

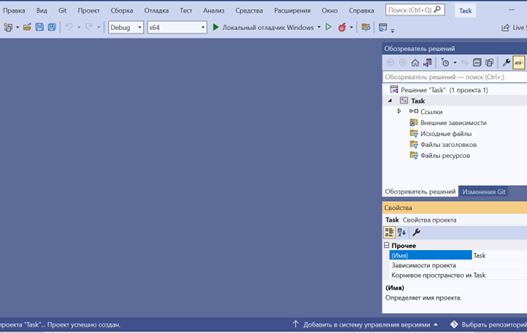


Введите имя проекта.

*(По умолчанию имя решения, содержащего проект, совпадает с именем проекта, однако можно ввести другое имя. Также можно указать другое расположение для проекта)*

И нажмите кнопку **Создать**.

После загрузки должно появится окно **Обозреватель решений.** В нем отражается состав проекта, включающий папки **Исходный код, Файлы заголовков и Файлы ресурсов**



**Исходный код - для файлов на языке С (\*.с)**

**Файлы заголовков - для \*.h**

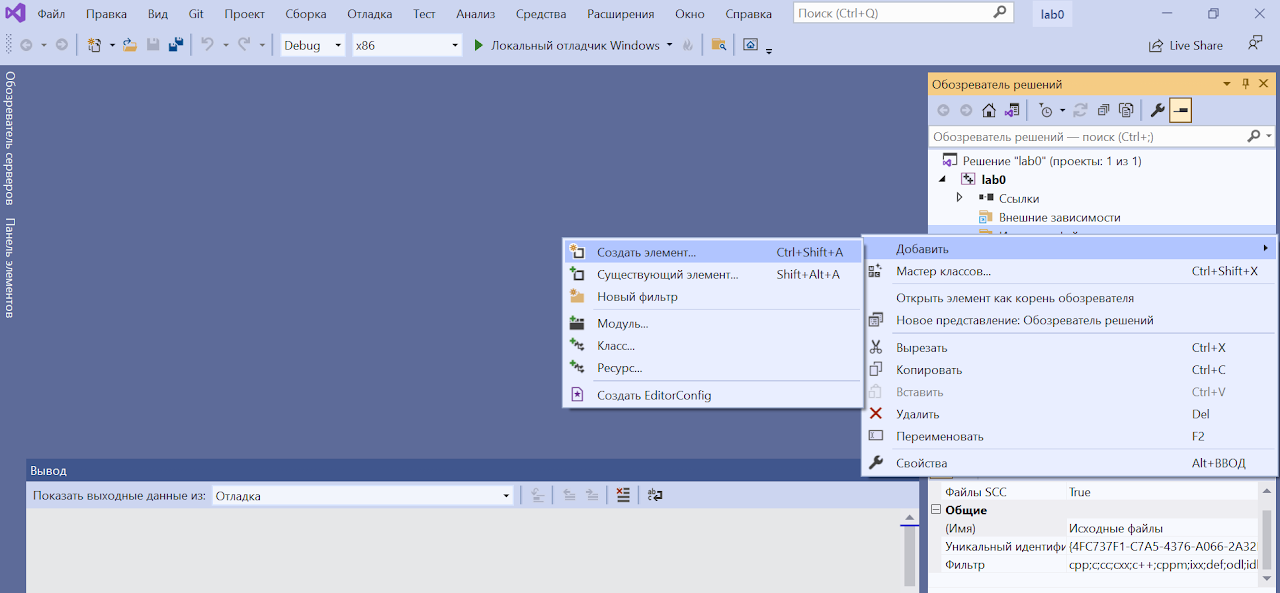
**Файлы ресурсов - для подгружаемых ресурсов**

Если окно **Обозреватель решений** не открыто, выберите в меню **Вид** пункт **Обозреватель решений**.

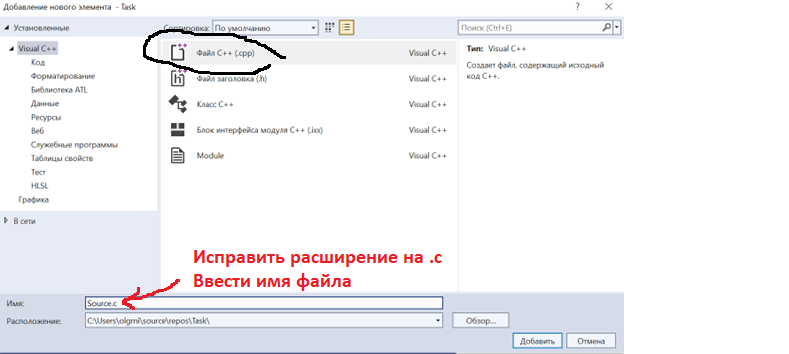
## 2. Создание файла для кода программы

Чтобы добавить новый исходный файл в проект, выполнив следующие действия.

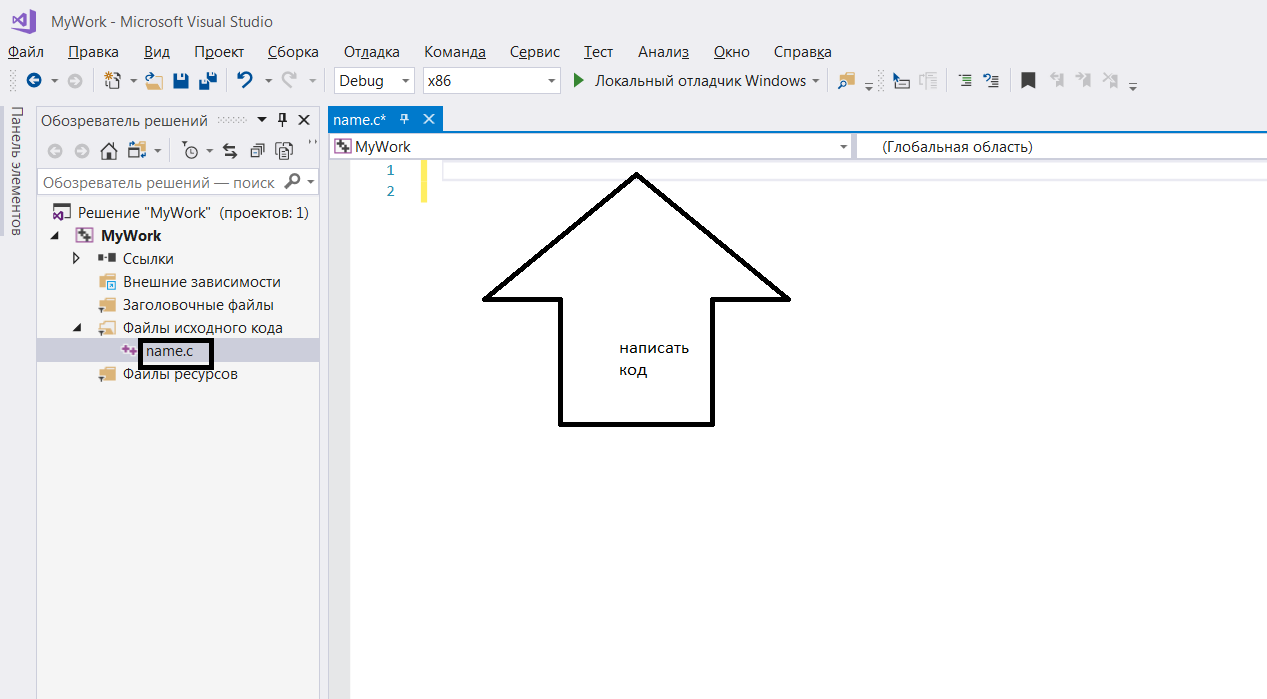
1. В окне **Обозреватель решений** щелкните правой кнопкой мыши папку **Исходные файлы** и последовательно выберите пункты **Добавить** и **Новый элемент**.



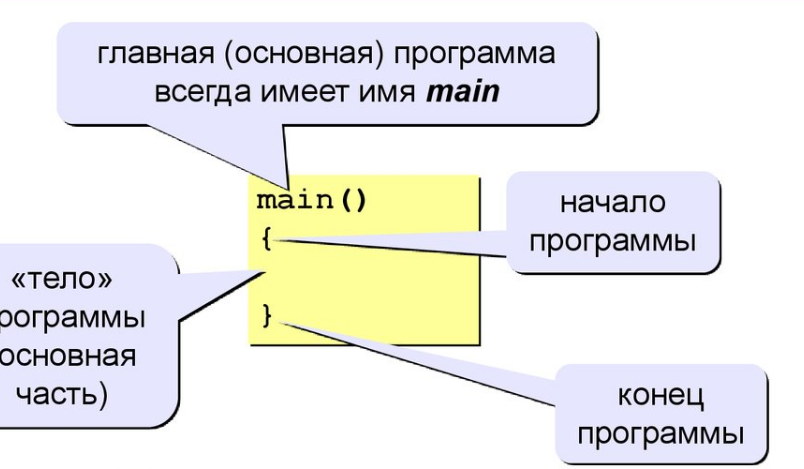
2. В узле **Код** выберите элемент **Файл C++ (.cpp)**, введите имя файла и замените расширение на .с и нажмите кнопку **Добавить**.



Файл **name.с** появится в папке "**Исходные файлы**" в окне **Обозреватель решений** и откроется в редакторе Visual Studio.



## 3. Написание простой программы



**Пример разработки консольного приложения – вывода текста**

Программа, написанная на языке С, состоит из одной или нескольких функций, одна из которых обязательно имеет идентификатор (имя) **main()** – основная, главная. Ее назначение – управление всей работой программы (проекта).

Консольное приложение отличается от всех других способом ввода-вывода данных.

Для вывода информации в консольном приложении чаще всего используются функции ***puts(…)*** и ***printf(…)***, которым в качестве аргумента передается строка текста, заключенного в двойные кавычки.

В файле, открытом в редакторе, введите код примера:

#include <locale.h>

#include <stdio.h>

void main()

{

setlocale(LC\_ALL, "RUS"); // для переключения русской кодировки

puts ("Моя программа"); // вывод строки

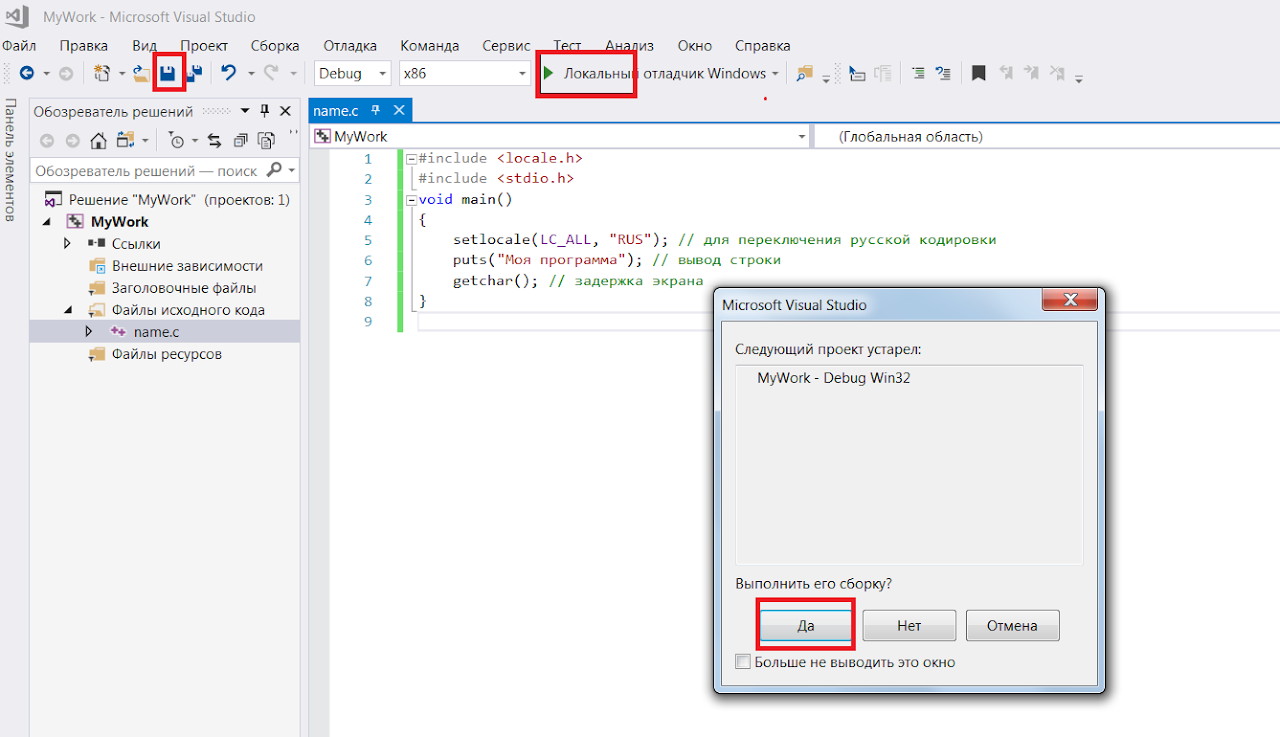
getchar(); // задержка экрана

}

Сохраните файл.

## 4. Проверка работы программы

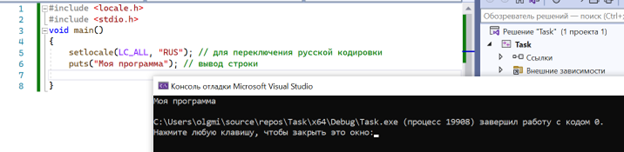
В меню **Отладка** выберите команду **Запуск без отладки** (комбинация клавиш <Ctrl-F5>) или нажмите на зеленый треугольник **Локальный отладчик Windows**



// Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"

// Отладка программы: F5 или меню "Отладка" > "Запустить отладку"

Процесс сборки займет некоторое время, в результате появится консоль



// В окне обозревателя решений можно добавлять файлы и управлять ими.

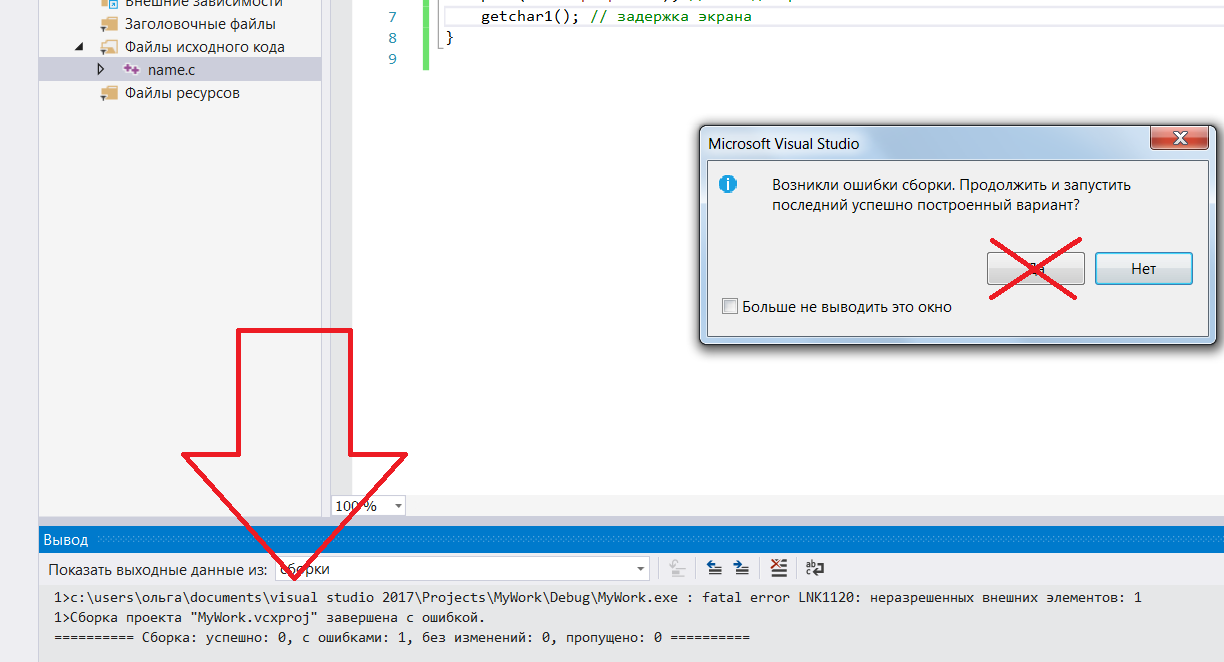
// В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения.

// В окне "Список ошибок" можно просматривать ошибки.

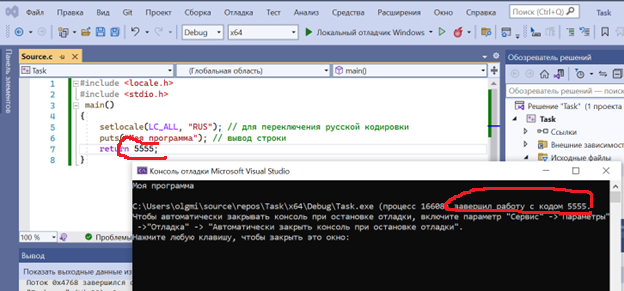
// Последовательно выберите пункты меню "Проект" > "Добавить новый элемент", чтобы создать файлы кода, или "Проект" > "Добавить существующий элемент", чтобы добавить в проект существующие файлы кода.

// Чтобы снова открыть этот проект позже, выберите пункты меню "Файл" > "Открыть" > "Проект" и выберите SLN-файл.

И если что-то пошло не так ищите ошибку в коде



Успешное выполнение программы отслеживается внешней системой по коду возврата



1.2 Осуществите вывод сообщения "Hello Word!".

Найдите в [справочнике](http://paulfertser.info/polnyy_spravochnik_po_c_gerbert_shildt/main.htm) описание функции puts() (*глава 13*).

Программа - это последовательность инструкций, в который указано, как выполнять вычисления. Каждая программа на С состоит из функций:

**имя\_функции(список параметров)**

Библиотечные или стандартные функции вызываются по имени:

 puts("Что-то сообщить");

Собственные - это код, который пишется в фигурных скобках:

myfunction()

{

//тело функции от скобки до скобки

}

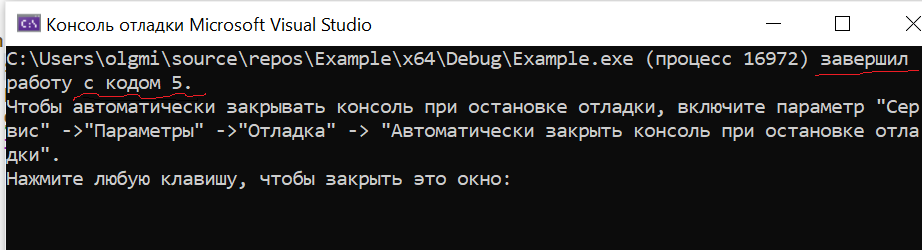
1.3 Удалите первую строку  #include <stdio.h>  и прочитайте сообщения об ошибке.

1.4 Добавьте [локализацию](https://sites.google.com/view/course-of-study1-c/главная/инструменты-разработки/языковые-настройки-и-локализация) для печати сообщений на русском языке.

1.5 Добавьте вызов функции getchar();

1.6 Добавьте оператор return 0;

Обратите внимание на сообщения ОС при завершении программы.



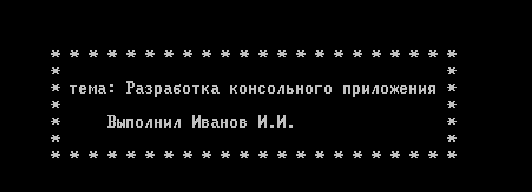
*Рис. Выполнение программы в консоле MVS*

**Задание 2**

1. Напишите программу, которая выводит тему лабораторной работы, и информацию об ее исполнителе (группа, студент).

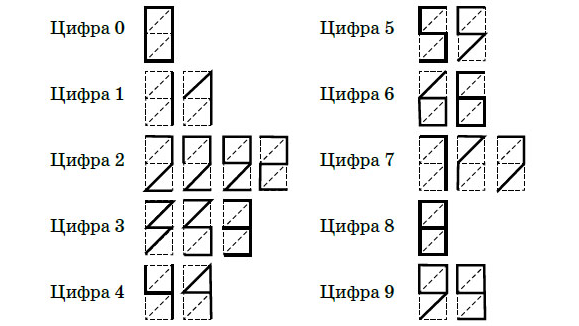
2. Реализуйте форматирование темы работы по центру консоли, а информации об исполнителе – слева к краю.

3. Окружите текст рамкой из символов -\* или – или >.



**Задание 3**

Составьте программу, которая выводит число, месяц и последние две цифры года рождения по принципу написания индекса на конверте с помощью символов \, |, \_,/.

*Сочетания символов, состоящих из косой черты (****\****), за которой следует буква или набор цифр, называются escape-последовательностями. Для управления выводом, например, для перехода на  новую строку, табуляции необходимо использовать escape-последовательности, к-е следует рассматривается как один символ и, следовательно, является допустимой символьной константой.  В связи с этим косая черта не является одиночным символом и для ее вывода необходимо записать два таких символа '\\'*

**Задание 4**

1. Выделите код для рисования рамки  (зад. 1)  и перенесите его в отдельную функцию **name()**:

*void name ()*

*{*

*puts();// здесь код создания рамки и названием работы и ФИО исполнителя*

*}*

2.  Расположите функцию **name()** в файле программеперед **main()**

3. Добавьте вызов  функции **name()**  в  тело **main()** вместо вынесенного в функцию name() кода.

4. Убедитесь, что программа осуществляет вывод.

5. Аналогичным образом постройте функцию **date()** для вывода цифр даты рождения.