

Для начала работы необходимо открыть среду разработки Microsoft Visual Studio – на ближайшее время она станет вашим родным домом, ближайшим другом и супругом одновременно.

Покликая по разным кнопочкам нужно найти в ней пункт “Создать проект”, и в появившемся окне выбрать шаблоны языка C++, а затем указать тип “Консольное приложение Win32”. Здесь же необходимо указать название создаваемого проекта и его месторасположение на диске (сохранять нужно в свою папку на сетевом диске).

Если вы всё сделали правильно, то ваш проект будет создан, а вместе с ним и 4 стандартных файла. Структуру проекта (директории и файлы) можно увидеть в окне Обозревателя Решений. Один из созданных файлов, под именем “Название_вашего_проекта.cpp”, будет содержать функцию `main` (или её разновидность с похожим названием). Этот файл нам и нужен.

Выглядеть он будет примерно как-то так:

```
1 // test_cpp_playground.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.
2 //
3
4 #include "pch.h"
5 #include <iostream>
6
7 int main()
8 {
9     std::cout << "Hello World!\n";
10    return 0;
11 }
12
13 // Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"
14 // Отладка программы: F5 или меню "Отладка" > "Запустить отладку"
15
```

Вот здесь, внутри функции `main`, мы и будем начинать писать любой код. Внутри – это значит в пределах фигурных скобок, от `{` и до `}`. Текст всех заданий можно писать последовательно друг за другом.

1. Для начала, просто создайте целочисленную переменную, и присвойте ей какое-нибудь значение. Допустим от -100 до 100.

Что-то наподобие этого:

```
7 int main()
8 {
9     int random_name = -39;
10
11
12
13     return 0;
14 }
```

Синтаксис объявления переменной такой: сначала пишется ключевое слово, которое определяет, что за тип данных будет у этой переменной (в данном случае, тип данных `int` – целое число).

Потом через пробел пишется имя переменной. Имена переменных придумывает сам программист и, как правило, они должны отражать её назначение в ходе программы.

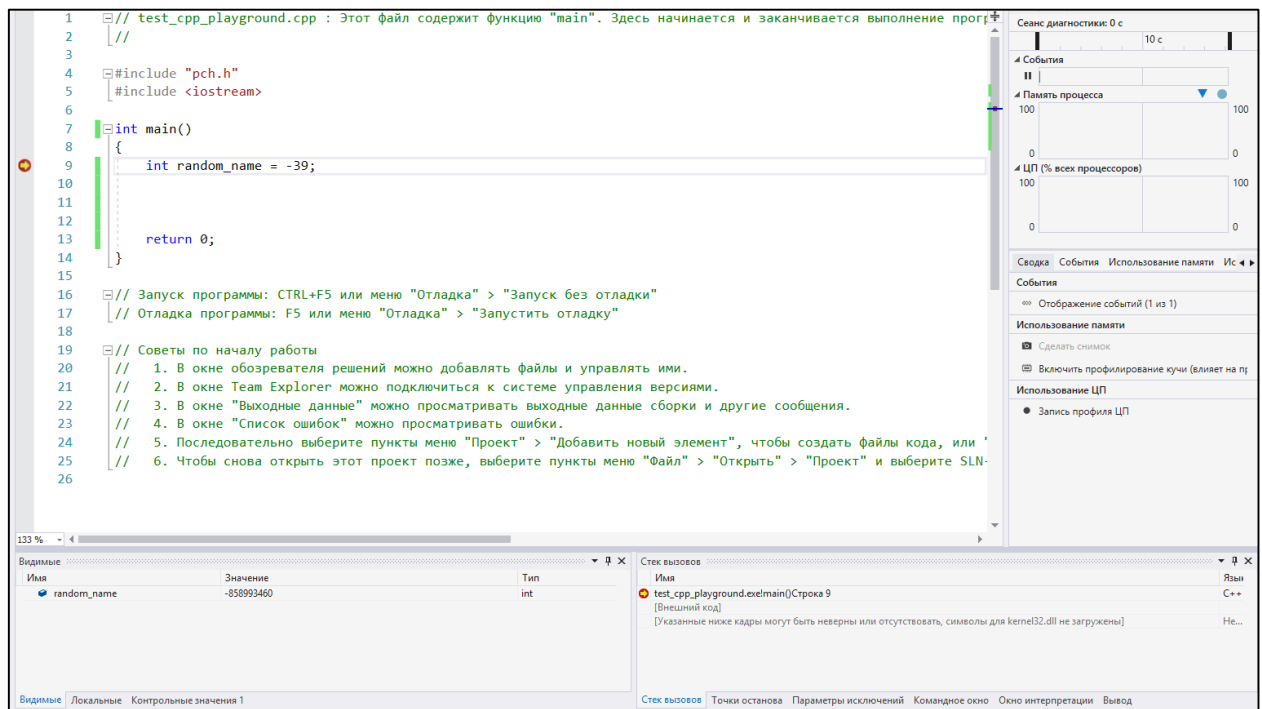
Далее идёт знак присваивания (равно) и начальное значение. Т.е. переменная `random_name` будет изначально иметь значение -39. Иначе

говоря, этой строчкой мы выделяем ячейку памяти компьютера под какое-то целое число, и записываем туда значение -39.

По аналогии, создайте еще 4 переменные с другими типами данными (какими, выберите сами) и любыми значениями.

2. Освойте искусство Отладки программы. Отладка — это процесс пошагового выполнения программы, с возможностью наблюдения за её состоянием на каждом шагу.

Вот примерно так должна выглядеть Visual Studio в режиме отладки:



В столбце слева от редактора кода по нажатию мыши напротив нужной строки можно расставлять точки останова (отображаются как красные кружки). По достижению строки с точкой останова программа в режиме отладки, как нетрудно догадаться, приостановится. Такая пауза позволит вам просмотреть значения всех видимых и локальных переменных, находящихся в текущей зоне видимости. Всю эту информацию можно почерпнуть из нижней части окна.

Нажатием клавиши F5 выполнение программы возобновляется до достижения следующей точки останова.

Но по началу гораздо удобнее будет выполнять всю программу целиком по шагам. Для того чтобы сделать шаг нужно нажать клавишу F10 (или F11, если вы хотите перейти внутрь вызываемой функции, но вы пока этого не хотите).

Попробуйте пройти всю вашу программу по шагам, нажимая клавишу F10, внимательно наблюдая за тем, как после выполнения каждого шага меняется описание видимых и локальных переменных в нижней панели окна.

В дальнейшем, для лучшего понимания программирования, рекомендуется каждую вашу программу выполнять и просматривать по шагам в режиме Отладки.

3. Создайте две целочисленные переменные A и B. Не инициализируйте их сразу, а сделайте это отдельным действием. После чего просуммируйте их и запишите результат в новую переменную. Вот так:

```
7 int main()
8 {
9     int a = -20;
10    int b = -56;
11
12    int c = a + b;
13
14    return 0;
15 }
```

По аналогии, сохраните в переменных результат разности, произведения и частного.

Далее, рассчитайте выражение $\frac{(a+b)*(a-b)+a/b}{(a*a*a-2*a*b+b*b)}$.

Пройдитесь по шагам в Отладке и посмотрите, чему равно значение этого выражения.

Теперь поменяйте типы данных всех переменных на *float*, вместо *int*. Снова пройдитесь по Отладке и посмотрите, как изменились значения в переменных.

Подсчитайте, сколько памяти компьютера используется в вашей программе (для переменных).

Модифицируйте код программы так, чтобы в нём использовалось лишь 3 разных переменных.

4. Попробуйте без ошибок переписать и запустить следующий код:

```
1 #include "pch.h" // or maybe you'll need "stdafx.h"
2 #include <iostream>
3
4 int main()
5 {
6     using std::cout;
7     using std::endl;
8
9     cout << "Тип объекта Его размер в байтах" << endl;
10    cout << " Размерность int: " << sizeof(int) << "\n";
11    cout << " Размерность char: " << sizeof(char) << "\n";
12    cout << " Размерность unsigned: " << sizeof(unsigned) << "\n";
13    cout << " Размерность long: " << sizeof(long) << "\n";
14    cout << " Размерность float: " << sizeof(float) << "\n";
15    cout << " Размерность double: " << sizeof(double) << "\n";
16    cout << "\n";
17
18    return 0;
19 }
```

Запустите. Что-то не так, заметили?

Это всё потому, что C++ адекватно воспринимает только английские символы.

5. Но этого можно избежать, добавив в программу немного кода специальных инструкций для смены кодировки символов на русскоязычную.

Для того, чтобы позволить программе выводить на консоль русскоязычные символы, необходимо вызвать в начале функции `main` функцию `setlocale` в следующем виде:

```
setlocale(LC_CTYPE, "Russian");
```

Может быть, данная запись вызовет ошибку – не найдет идентификатор `setlocale`. Это означает, что для использования этой функции потребуется подключить библиотеку, в которой она находится. Это делается посредством добавления в файл директивы `include`:

```
#include <locale>
```

Предыдущие действия касались только вывода на консоль. Если же вам нужно вводить данные через консоль, то потребуются иные действия. В принципе они довольно схожи: в начале `main`-а нужно прописать вызов специальных функций:

```
SetConsoleCP(1251);  
SetConsoleOutputCP(1251);
```

А также директивы `include`:

```
#include <Windows.h>
```

Однако для ввода этого может быть недостаточно, и если русскоязычный ввод всё равно некорректно воспринимается программой, нужно поменять шрифт в консоли (если что, это то чёрное окошко, в котором запускается программа). Когда окно консоли появится на экране, нужно правой кнопкой мыши нажать на его заголовок, выбрать из контекстного меню пункт «Значения по умолчанию», далее перейти на вкладку «Шрифт», и в списке доступных шрифтов выбрать `Lucida Console`. И не спрашивайте почему.

6. Разработайте консольное приложение, выполняющее вопросно-ответный алгоритм взаимодействия с пользователем в соответствии с примером.

Тут используется много материала, до которого мы еще не дошли. Но некоторые пояснения можно дать и сейчас:

На 12 строке создается последовательность из 80 символов (читай, строка). Далее в эту память будет помещаться то, что введет с клавиатуры пользователь в консоли.

Функция `printf` выводит строку на консоль.

Функция `gets_s` считывает строку с консоли в переменную `str` (чтобы ввести строку её нужно напечатать в консоли и нажать Enter).

На 16 строке описано условие: если значение в переменной `str` (строка) равно «Да» или значение в переменной `str` равно «да», то выполняется строка `puts("Верно")`, которая выводит это сообщение на экран. Если условие не

верно – выполняется блок, идущий после ключевого слова *else* (иначе), т.е. вывод на консоль строки “Неверно”.

```
6  int main()
7  {
8      setlocale(LC_CTYPE, "Russian");
9      SetConsoleCP(1251);
10     SetConsoleOutputCP(1251);
11
12     char str[80];
13
14     printf("Столица Шотландии - Эдинбург?\n");
15     gets_s(str, sizeof(str));
16     if (strcmp(str, "Да") == 0 || strcmp(str, "да") == 0)
17     {
18         puts("Верно");
19     }
20     else
21     {
22         puts("Неверно");
23     }
24
25     return 0;
26 }
```

В этом примере используется только один вопрос. Сделайте так, чтобы задавалась последовательность из 5-10 вопросов.

В конце программы выведите процентное соотношение правильных ответов к общему числу вопросов.

Не забудьте пройти по шагам программы в режиме Отладки.