**Лабораторная работа №15**

**Тема: Организация ввода/вывода в С++**

**Цель работы:** научиться создавать эффективный интерфейс с пользователем, использовать в программах данные разных типов.

**Коды формируемых компетенций:** ПК 2.3, ПК .2.4, ОК 1, ОК 2, ОК3, ЛР13, ЛР14.

**Теоретическое введение.**

1. Имена переменных. В языке программирования С++ приветствуется присваивание переменным выразительных имен. Если переменная представляет стоимость поездки , то ей следует присвоить такое имя, например, как cost\_of\_trip или costOfTrip , но не х или cot .

В языке С++ необходимо придерживаться следующих правил присваивания имен:

* Допускается использование только тех символов, которые представляют алфавит и цифры, а также символ подчеркивания (\_).
* Первым символом не должна быть цифра.
* Символы в верхнем регистре отличаются от символов в нижнем регистре.
* В качестве имени нельзя использовать ключевое словоязыка С++.
* Имена, начинающиеся с двойного подчеркивания или с одного подчеркивания и следующим за ним символом в верхнем регистре, зарезервированы для реализаций С++, то есть их используют компиляторы и ресурсы. Имена, начинающиеся с одного символа подчеркивания, зарезервированы для использования в качестве глобальных идентификаторов в реализациях С++.
* Имя может иметь произвольную длину, и все символы в имени являются значимыми.

Обычно, если переменной присваивается имя , состоящее и з двух или более слов, то для разделения слов используется символ подчеркивания , как в имени my\_data , или же первую букву каждого слова, кроме первого, записывают в верхнем регистре, как в имени myData.

1. **Описание и инициализация переменных**

Прежде чем использовать в программе какую-то переменную, надо дать ей описание, то есть сказать, какое имя имеет переменная и каков её тип. Вначале указывается тип переменной, а затем её имя. Например:

int k; // это переменная целого типа int

double x; // это переменная вещественного типа удвоенной точности: х=2.77;

char sym; //это переменная символьного типа принимает значения одного символа: sym=’Д’;

string imy; //это переменная строкового типа может принимать данные, состоящие из букв, цифр, знаков: imy=”Вася”;

Если имеется несколько переменных одного типа, то допускается их описание через запятую в одном операторе, например:

double a, b, c;

После описания переменной её можно использовать, но возникает вопрос: а какое значение имеет переменная сразу же после её описания? Ответ таков: в программе на языке C или C++ переменная после описания имеет произвольное значение, то есть ей просто выделяется свободный участок памяти и на этом всё заканчивается. В переменной хранится та последовательность двоичных цифр, которая была там до выделения памяти под эту переменную. Такой подход таит определённую опасность: можно по ошибке использовать в выражении переменную, которой не было присвоено ни какое значение:

double x, y;

y = 2 \* x;

Так как переменная x ни как не была определена, т. е. имела произвольное значение (чаще всего — это очень большое число или наоборот — очень маленькое), то и переменная y получит явно не то значение, на которое рассчитывал пользователь.

Чтобы избежать такой ошибки, Бьерн Страуструп рекомендует инициализировать переменные, то есть не просто выделять память под переменные, но и задавать им при этом необходимые значения. Например:

double a=3, b=4, c=5;

Инициализация переменных выполняется один раз на этапе компиляции, она не снижает скорость работы программы, но при этом уменьшает риск использования переменной, которая не получила ни какого значения.

1. **Константы**

Константы как и переменные предназначены для хранения данных, но в отличие от переменных должны инициализироваться при объявлении и им нельзя присвоить новое значение. Константы можно определить двумя способами:

#define PI 3.14159265 //определяем число π=3.14

const double pi=3.14159262;

предпочтительнее использовать второй способ так как при определении константы с помощью #define она не имеет типа,а тип константы вам может понадобиться.

**Виды констант:**

1. Целые десятичные числа, начинающиеся не с 0.

2. Целые восьмеричные числа, начинаются с 0.

3. Целые шестнадцатеричные числа. Запись начинается с 0х

4. Если в записи числовой константы присутствует десятичная точка (2.5) или экспоненциальное расширение (1Е-8),то компилятор рассматривает её как вещественное число и ставит ей в соответствие тип double

5. Символьные константы заключаются в апострофы. Например, **‘a’** . Строковые константы, представляющие собой символьные последовательности, заключается в двойные кавычки **“abs”**

Особую разновидность символьных констант представляют управляющие символы. Их назначение – управление выводом на экран.

\n – переход на новую строку

\t – табуляция

Именованную константу можно определить с помощью предпроцессорной директивы #define

**Задание 1. Нижеприведенная программа демонстрирует как с помощью средств самого языка С++ узнать всю информацию о типах переменных. В программе использованы функции:**

INT32\_MAX – выдает максимально возможное число, которое можно записать в переменную типа int для 32-разрядной архитектуры;

**sizeof(int)** – выдает размер памяти, занимаемой переменной типа int

// Программа примера 1

#include "stdafx.h"

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int n\_max =INT32\_MAX;

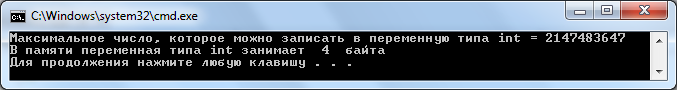
cout << "Максимальное число, которое можно записать в переменную типа int = "<< n\_max<<endl;

int n\_byte = sizeof(int);

cout << "В памяти переменная типа int занимает " << n\_byte << " байта"<<endl;

return 0;

}

Результат работы этой программы:

**Задание 2.** Известно, что кроме целочисленного типа int существуют еще типы short и long. Дополните программу так, чтобы выводилось не только максимальное число типа int, но и максимальные числа типов short и long. Также с помощью функции sizeof(), узнайте, сколько байт отводится под эти типы.

**Задание 3**. Пусть пользователь вводит десятеричное число. Необходимо вывести его в шестнадцатеричной и восьмеричной системах счисления .

Воспользуемся возможностями операторы вывода **cout**, для которого существуют манипуляторы **dec** , **hex** и **oct** , которые сигнализируют объекту **cout** о форматах отображения целых чисел: десятичном, шестнадцатеричном и восьмеричном. Введите и протестируйте работу нижеприведенной программы.

// Программа примера 2

#include "stdafx.h"

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int x;

cout << "Введите любое десятичное число: ";

cin >> x;

cout << "В шестнадцатеричном формате = ";

cout<< hex; // переключаем вывод в шестнадцатеричное представление

cout << x<<endl;

cout << oct; // переключаем вывод в восьмеричное представление

cout << "В восьмеричном формате= ";

cout << x << endl;

return 0;

}

**Задание 4.** Изучаем тип переменных – **char**. Этот тип предназначен для хранения символов, таких как буквы и цифры. Самым распространенным набором символов в США является ASCII, где каждый символ представлен числовым кодом (кодом ASCII) . Например, символу А соответствует код 65, символу М соответствует код 77 и так далее. Для того, чтобы получить код введенного символа нужно присвоить этот код переменной целого типа. Рассмотрите как это сделано в ниже приведенной программе. Введите эту программу, протестируйте.

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

char letter;

cin >> letter;

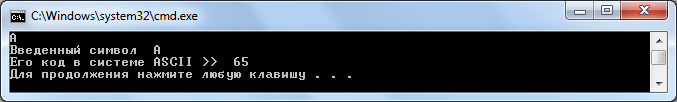
int kod = letter;

cout << "Введенный символ "<<letter<<endl<<"Его код в системе ASCII >> "<< kod<<endl;

return 0;

}

Результаты работы программы:



Дополнительно: добавьте в программу нижеприведенный фрагмент:

kod = kod + 1;

letter = kod;

cout << letter;

Что будет выведено на экран?

**Задание 5. Создайте переменные с разными типами данных и предложите пользователю ввести в них значения. После ввода, отобразите их на экране.**

**Задание 6.** Пользователь должен ввести 2 числа. Вам надо показать на экран произведение этих чисел, сумму и разницу. Покажите так же среднее арифметическое этих введенных чисел.

**Контрольные вопросы**

1. Перечислите возможные типы переменных и какие значения они могут принимать?
2. Какое значение имеет переменная x после описания: int x;
3. Что такое «инициализация переменной»? Когда она выполняется?
4. Что такое константа? Как задать константу?
5. Что выполняет директива #define e=2.71
6. Виды констант?
7. Как задать восьмеричную, шестнадцатеричную константы?
8. Для чего нужна предпроцессорная директива #define?
9. Что будет записано в переменную n\_max после выполнения команды?

int n\_max =INT32\_MAX;

1. Необходимо задать строковую константу, содержащую все гласные строчные буквы русского алфавита. Как это сделать?