

Практическое занятие №1.

Тема: Управление программой при работе с простыми типами данных

Цель: получение практических навыков использования операторов цикла и условных операторов при разработке программ.

Общие сведения о переменных

В языке C++ для обозначения основных типов данных применяются следующие обозначения (ключевые слова):

- **char** – символ;
- **int** – целое число;
- **double** – дробное (вещественное) число.

Переменная – это именованная область памяти, в которой хранятся данные определенного типа. У переменной есть имя и значение. Имя служит для обращения к области памяти, в которой хранится значение. Во время выполнения программы значение переменной можно изменять. Перед использованием любая переменная должна быть описана (объявлена). Для этого, вначале указывается тип данных переменной, а затем через пробел – идентификатор, обозначающий её имя. Объявление заканчивается точкой с запятой.

Примеры:

```
double a1;    // объявлена переменная a1 для дробных чисел  
int two;     // объявлена целочисленная переменная two  
char s;      // объявлена символьная переменная s.
```

Для объявления нескольких однотипных переменных их разделяют запятыми после указания типа данных.

Примеры:

```
int a,b,c;      //объявлены три целочисленные переменные  
double a1,b1;  // объявлены две вещественные переменные.
```

Ниже представлена программа, в которой вычисляется сумма двух целых чисел.

```
#include<iostream>  
using namespace std;  
int main( )  
{
```

```

int a;           // объявление целочисленной переменной a
int b;           // объявление целочисленной переменной b
int c;           // объявление целочисленной переменной c
    a=24;        // занесение числа 24 в переменную a
    b=46;        // занесение числа 46 в переменную b
    c=a+b;       // вычисление суммы и занесение её в c
    cout << c;   // вывод на экран значения переменной c
return 0;
}

```

Условные операторы

Конструкция if – else. Составной оператор **if – else** используется для разветвления хода вычислений в зависимости от условия; поэтому он называется условным.

```

#include<iostream>
using namespace std;
int main( )
{
    int a,b;
    int x,y;
    cout << "Insert a, b" << "\n";
    cin >> a >> b;
        if (a>b)
            x=a + b;
        else
            x= a – b;
        y = 25 + x;
        cout << y;
return 0;
}

```

В этом примере вводятся с клавиатуры два целых числа. Затем выполняется операция сравнения **a>b**. Если условие истинно, то выполняется сложение **a + b** и результат присваивается переменной **x**. После чего ветка **else** пропускается и выполняется оператор **y = 25 + x**. Если же условие не выполняется, то переменной **x** присваивается разность **a – b** и, далее, выполняется оператор **y = 25 + x**. В заключении значение **y** выводится на экран. Если введенные значения: **a=10, b=5**, то в результате выполнения программы на экран будет выведено число 40.

Переключатель switch. Условный оператор **if – else** разветвляет выполнение программы на два направления. Чтобы увеличить их количество используется переключатель **switch**.

```

#include<iostream>
using namespace std;

```

```

int main( )
{
    int a;
    cout <<"Insert numeral(0 - 7 )";
    cin >> a;
    switch(a)
    {
        case 1: cout << " One ";    break;
        case 2: cout << " Two " ;    break;
        case 3: cout << "Three ";    break;
        case 4: cout << " Four " ;    break;
        case 5: cout << " Five " ;    break;
        case 6: cout << " Six " ;    break;
        case 7: cout << "Seven " ;    break;
        default: cout << " Error of enter ";
    }
    return 0;
}

```

Пример демонстрирует перевод цифр в их словесный эквивалент. Переменной **a** с клавиатуры присваивается целочисленное значение. В операторе **switch** анализируется содержание переменной **a**. Допустим её значение равно 4. Тогда будет выполнена строка:

```

    case 4: cout << " Four " ;    break;

```

на экран будет выведено слово **Four**. Оператор **break** в конце строки означает немедленный выход из оператора **switch**.

Циклы

Цикл for. Он применяется, когда все условия цикла известны и являются постоянными. Ниже приводится пример вычисления суммы чисел от 20 до 50.

```

#include<iostream >
using namespace std;
int main( )
{
    int i, s=0;
    for(i=20; i<=50; i++)
        s = s + i;
    cout <<"Sum = " << s;
    return 0;
}

```

В данном примере после выполнения первого шага сумма будет равна 20. После этого **i** увеличится на 1 и станет равным 21. На втором шаге сумма

примет значение 41(20+21). Вновь **i** увеличится на 1, и т. д. до тех пор, пока **i** не примет значение 51. В этом случае условие продолжения цикла перестанет выполняться (**i** <= 50) и произойдет выход из цикла.

Цикл while. Также как и цикл **for**, цикл **while** относится к циклам с предусловием, которое указывается в скобках после слова **while**. Реализуем предыдущий пример, используя вместо цикла **for** цикл **while**.

```
#include<iostream>
using namespace std;
int main( )
{
    int i=20;
    int s=0;
    while (i <= 50)
    {
        s = s + i;
        i++;
    }

    cout <<"Sum = " << s;
    return 0;
}
```

При использовании цикла **while** внимания требует параметр цикла, который необходимо проинициализировать перед использованием и изменять в теле цикла.

Конструкция do-while. Описывает циклы с постусловием: сначала выполняется тело цикла (хотя бы один раз), а затем проверяется условие. Рассмотрим пример вычисления произведения чисел от 5 до 15.

```
#include<iostream>
using namespace std;
int main( )
{
    int m=1; // начальное значение произведения = 1
    int k=5; // начальное значение параметра цикла
    do
    {
        m = m * k;
        k++; // изменение параметра цикла
    }
    while (k<=15); // условие выполнения цикла
    cout << m;
}
```

```
return 0;  
}
```

Практические задания

Выберите один из вариантов задания.

Вариант А.

Задание. Разработать программу «Калькулятор», которая имитирует вычислительные действия простейшего арифметического калькулятора.

Вначале с клавиатуры вводится первое число, затем – знак арифметической операции (+, – , * , /) и второе число. В зависимости от знака операции над числами выполняется соответствующее действие.

Результат операции выводится на экран и используется в следующей операции до тех пор, пока вместо знака операции не будет введен символ «С», или символ «Q». При этом, если вводится символ «С», то результат обнуляется, но калькулятор «продолжает работать», т.е. – «запрашивает» первое число и т.д.

Если вводится символ, отличный от символа знака операции, символа «С» или символа «Q», то программа «Калькулятор» продолжает работать, но «просит» повторить ввод. Если вводится символ «Q», то работа программы заканчивается с выводом последнего результата.

Вариант В.

Задание. Разработать программу «Калькулятор», которая выполняет простые вычислительные действия: вначале вводится первое число, затем – знак арифметической операции (+, – , * , /) и второе число. В зависимости от знака операции над числами выполняется соответствующее действие. Результат операции выводится на экран и сохраняется от действия к действию до тех пор, пока вместо знака операции не будет введен символ «Z» - завершение программы с выводом последнего полученного результата.

Вариант С.

Задание. Разработать программу, которая демонстрирует выполнение арифметических операций с двумя числами. Вначале вводятся два числа, затем – знак арифметической операции (+, – , * , /). В зависимости от знака операции над числами выполняется соответствующее действие. Результат операции выводится на экран.

Отчет оформляется по общеустановленным правилам в *электронном виде* со следующим содержанием:

- 1) титульный лист (см. ПРИЛОЖЕНИЕ),
- 2) тема и цель практического занятия,
- 3) задание на практическое занятие,
- 4) текст программы с комментариями,

- 5) результаты работы программы (вид экрана) и
- 6) выводы по созданному проекту и использованию средств языка программирования.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Факультет Информационных систем и технологий
Кафедра Безопасности информационных систем

Дисциплина «Технологии программирования»

ОТЧЕТ

за практическое занятие №1

**Тема: Управление программой при работе с простыми типами
данных**

Выполнил
Студент 2 курса, гр. ИБ-01вп
_____ Иванов М.А.

Принял
Доцент кафедры БИС
_____ Медведев В.А.