Практическая работа №3
Составление циклических алгоритмов

1 Цель работы

1.1 Научиться составлять циклические алгоритмы, используя основные алгоритмические конструкции;

1.2 Научиться составлять полные и неизбыточные наборы тестов.

2 Литература

2.1 Павловская, Т. А. C/C++. Программирование на языке высокого уровня. - Санкт-Петербург: Питер, 2017 - 461 с.

2.2 Семакин, И.Г. Основы программирования: учебник/ И.Г. Семакин, А.П. Шестаков. – Москва: Мастерство; НМЦ СПО; Высшая школа, 2016 – 432 с.

3 Основное оборудование

3.1 Персональный компьютер.

4 Подготовка к работе

4.1 Повторить лекционный материал.

4.2 Подготовить бланк отчета.

5 Задание

5.1 Разработать программу для предложенной задачи с использованием цикла со счетчиком: напечатать таблицу умножения на число n (значение n вводится с клавиатуры; $1 \leq n \leq 9$).

5.2 Пользователь вводит делимое и делитель, посчитать частное двух чисел. При вводе некорректного делителя (на ноль делить нельзя), запрашивать его снова. Использовать цикл с предусловием while.

5.3 Написать программу запрашивающую у пользователя рост учеников и выводящую средний рост после того, как пользователь ввел число 0. Цикл — для ввода роста каждого из учеников.

6 Порядок выполнения работы

6.1 Изучить пиложение п. 9

6.2 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист;

7.2 Цель работы;

7.3 Условия задач;
7.4 Графическое описание алгоритмов из п. 6.1−6.3;
7.5 Вывод по проделанной работе.

8 Контрольные вопросы

8.1 Что такое алгоритм?
8.2 Какой алгоритм называется циклическим?
8.3 В чем разница между циклом while и do...while?

9 Приложение

Циклы позволяют выполняет некоторый набор инструкций множество раз, пока соблюдается определенное условие.

Цикл while

Цикл while выполняет некоторый код, пока его условие истинно, то есть возвращает true. Он имеет следующее формальное определение:

```
while(условие)
{
    // выполняемые действия
}
```

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
6 * 6 = 36
7 * 7 = 49
8 * 8 = 64
9 * 9 = 81
```

После ключевого слова while в скобках идет условное выражение, которое возвращает true или false. Затем в фигурных скобках идет набор инструкций, которые составляют тело цикла. И пока условие возвращает true, будут выполняться инструкции в теле цикла.

Например, выведем квадраты чисел от 1 до 9:

```
#include <iostream>

int main()
```

```
{
	int i {1};
	while(i < 10)
	{
		std::cout << i << " * "  << i << " = "  << i * i << std::endl;
		i++;
	}
}
```

Здесь пока условие i < 10 истинно, будет выполняться цикл while, в котором выводится на консоль квадрат числа и инкрементируется переменная i. В какой-то момент переменная i увеличится до 10, условие i < 10 возвратит false, и цикл завершится.

Консольный вывод программы:

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
6 * 6 = 36
7 * 7 = 49
8 * 8 = 64
9 * 9 = 81
```

Каждый отдельный проход цикла называется итерацией. То есть в примере выше было 9 итераций.

Цикл for

Цикл for имеет следующее формальное определение:

```
for (инициализатор; условие; итерация)
{
	// тело цикла
}
```

Инициализатор выполняется один раз при начале выполнения цикла и представляет установку начальных условий, как правило, это инициализация

счетчиков - специальных переменных, которые используются для контроля за циклом.

Условие представляет условие, при соблюдении которого выполняется цикл. Как правило, в качестве условия используется операция сравнения, и если она возвращает ненулевое значение (то есть условие истинно), то выполняется тело цикла, а затем выполняется итерация.

Итерация выполняется после каждого завершения блока цикла и задает изменение параметров цикла. Обычно здесь происходит увеличение счетчиков цикла.

Например, перепишем программу по выводу квадратов чисел с помощью цикла for:

```
#include <iostream>

int main()
{
	for(int i {1}; i < 10; i++)
	{
		std::cout << i << " * " << i << " = " << i * i << std::endl;
	}
}
```

Первая часть объявления цикла - int i {1} - создает и инициализирует счетчик i. Фактически это то же самое, что и объявление и инициализация переменной. Счетчик необязательно должен представлять тип int. Это может быть и другой числовой тип, например, float. И перед выполнением цикла его значение будет равно 1.

Вторая часть - условие, при котором будет выполняться цикл. В данном случае цикл будет выполняться, пока переменная i не станет равна 10.

И третья часть - приращение счетчика на единицу. Опять же нам необязательно увеличивать на единицу. Можно уменьшать: i--. Можно изменять на другое значение: i+=2.

В итоге блок цикла сработает 9 раз, пока переменная i не станет равна 10. И каждый раз это значение будет увеличиваться на 1. И по сути мы получим тот же самый результат, что и в случае с циклом while:

```
1 * 1 = 1
2 * 2 = 4
3 * 3 = 9
4 * 4 = 16
5 * 5 = 25
6 * 6 = 36
7 * 7 = 49
8 * 8 = 64
9 * 9 = 81
```