



# Менеджмент в сфере услуг

## Услуги и сервис

## Услуги и сервис

# План лекции

- 1.Оператор цикла while
- 2.Оператор цикла do-while
- 3.Оператор цикла for

# 1.Оператор цикла while

Описание:

**while (выражение) оператор;**

Действие:

Выполняется оператор до тех пор, пока значение выражения в скобках истинно. Проверка значения выражения происходит перед каждым выполнением оператора. Когда значение **выражения** ложно, цикл while заканчивается. Если выражение ложно с самого начала, оператор не выполняется ни разу.

Комментарий:

Следует заметить, что после ключевого слова while и выражения, заключенного в круглые скобки, точка с запятой не ставится.

Оператор иногда называется телом цикла. В теле цикла должны выполняться действия, в результате которых меняется значение управляющего выражения. В противном случае можем получить бесконечный цикл.

## Пример:

```
/*Демонстрация цикла while*/  
#include <stdio.h>  
main( )  
{  
int i=1  
while (getchar()!='R') i++;  
/*оператор getchar() вводит любой символ  
с клавиатуры*/  
printf("Символ R %d-й",i);  
}
```

Приведенная выше программа позволяет определить порядковый номер первой введенной буквы **R** в последовательности символов. Она показывает использование цикла **while**, в теле которого всего одна инструкция (**i++ - увеличение значения целого числа i на единицу**). Если запустить эту программу на выполнение и ввести последовательность символов, например:

**abFk!Rgm** ,

то на экране появится строка:

**Символ R 6-й.**

## 2.Оператор цикла do-while

Описание:

**do оператор while (выражение);**

Действие:

В операторе **do-while** тело цикла выполняется по крайней мере один раз. Тело цикла будет выполняться до тех пор, пока выражение в скобках не примет ложное значение. Если оно ложно при входе в цикл, то его тело выполняется ровно один раз.

Комментарий: после слова **while** и выражения, заключенного в скобки, ставится точка с запятой. Если в теле цикла содержится более одной инструкции, то операторы цикла заключаются в фигурные скобки.

## Пример:

```
/*Демонстрация цикла do-while */  
#include <stdio.h>  
main()  
{  
int i=0; /*i=0, а не единице*/  
do i++; while (getchar()!='R');  
printf(“Символ R %d-й”,i);  
}
```

Программа, представленная выше,  
написана с циклом do-while.

## 3.Оператор цикла for

Описание:

**for (выражение 1; выражение 2; выражение 3) оператор;**

Действие:

В круглых скобках содержится три выражения. Первое из них служит для инициализации счетчика. Она осуществляется только один раз – когда цикл for начинает выполняться. Второе выражение необходимо для проверки условия, которая осуществляется перед каждым возможным выполнением тела цикла. Когда выражение становится ложным, цикл завершается. Третье выражение вычисляется в конце каждого выполнения тела цикла, происходит приращение числа на шаг.

Комментарий: в операторе цикла for точка с запятой после закрывающейся круглой скобки не ставится. Любое из трех или все три выражения в операторе могут отсутствовать, однако разделяющие их точки с запятыми опускать нельзя. Если отсутствует выражение 2, имеем бесконечный цикл.



**Пример:**

```
/*демонстрация цикла for*/
```

```
#include <stdio.h>
```

```
main()
```

```
{int i,j=1,k;
```

```
for (i=1;i<=3;i++)
```

```
printf(“Астана\n”);} /*В цикле for три раза выполняется  
функция вывода, i-управляющая переменная цикла*/
```

**Результат выполнения программы:**

**Астана**

**Астана**

**Астана**

## Оператор break

Описание:

Break используется для прекращения выполнения цикла из-за обнаружения ошибки, для организации дополнения к условию в заголовке цикла, для прекращения бесконечного цикла.

Пример:

```
while (st>0 && st<25)  
{  
if st==4||st==8||st==12)  
break;  
}
```

Работа цикла полностью прекращается, как только условие в операторе if становится истинным.

# Оператор continue

Описание: continue

Действие:

Этот оператор может использоваться во всех трех типах циклов. Как и в случае оператора break, он приводит к изменению характера выполнения программы. Однако вместо завершения работы цикла наличие оператора continue вызывает пропуск “оставшейся” части итерации и переход к началу следующей.

**Пример.** Заменяем в предыдущей программе оператор break на continue.

```
while (st>0 && st<25)  
{  
if (st==4||st==8||st==12)  
continue;  
}
```

При истинном условии в операторе if оператор continue вызывает пропуск идущих за ним операторов тела цикла и осуществляется переход к началу следующей итерации.

# Контрольные вопросы

1. Что такое циклический процесс?
2. Каким оператором реализуется цикл с предусловием?
3. Каким оператором реализуется цикл с постусловием?
4. Как работает оператор цикла for?
5. Как работают вложенные циклы?
6. Как применяются операторы break и continue?

# Литература

1. Подбельский В.В. Язык Си ++: Учебное пособие. - М.: Финансы и статистика, 1995, - 560 с.
2. Страуструп Б. Язык программирования Сг ++. - М.: Радио и связь, 1991. - 352 стр.
3. Собоцинский В.В. Практический курс Turbo Си ++. Основы объектно- ориентированного программирования. - М.: Свет, 1993. - 236 с.
4. Романов В.Ю. Программирование на языке Си ++. Практический подход. - М.: Компьютер, 1993. - 160 с.
5. Уинер Р. Язык турбо Си . - М.: Мир, 1991. - 384 с.
6. Юлин В.А., Булатова И.Р. Приглашение к Си. - Мн.: Высш. Шк., 1990,- 224 с.