

# ЯЗЫК С

---

Лекция 4: Циклы for и while.

# Циклы

Цикл – одна из управляющих конструкций в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.

Циклы подразделяют на безусловные и условные, которые также делятся на типы.

# Безусловный цикл

- Иногда в программах используются циклы, выход из которых не предусмотрен логикой цикла. Такие циклы называются безусловными, или бесконечными. Специальных синтаксических средств для создания бесконечных циклов, ввиду их нетипичности, большинство языков программирования не предусматривают, поэтому такие циклы создаются с помощью конструкций, предназначенных для создания условных циклов.

# Условные циклы

- Условные циклы делят на циклы с предусловием и циклы с постусловием. Также отдельно выделяют цикл со счётчиком, которые обычно реализуются как цикл с предусловием.

# Цикл с предусловием

- Цикл с предусловием — цикл, который выполняется, пока истинно некоторое условие, указанное перед его началом. Это условие проверяется **до** выполнения тела цикла, поэтому тело может быть не выполнено ни разу (если условие с самого начала ложно).
- В большинстве языков программирования реализуется оператор **while**.

# Реализация

- `while (<условие>) { <тело цикла> }`

- `int i=0,a=10;`

- `while (i<10)`

- `{`

- `a+=i;`

- `}`

# Цикл с постусловием

- Цикл с постусловием — цикл, в котором условие проверяется **после** выполнения тела цикла. Отсюда следует, что тело **всегда выполняется** хотя бы один раз. В Си данный цикл реализуется с помощью оператора — `do...while`.

# Реализация

- `do { <тело цикла> } while (<условие>)`
  
- `int i=2,a=10;`
- `do`
- `{`
- `i+=a;`
- `}`
- `while (i<10)`



# Цикл со счётчиком

- Цикл со счётчиком — цикл, в котором некоторая переменная изменяет своё значение от заданного начального значения до конечного значения с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз. В большинстве процедурных языков программирования реализуется оператор `for`.

# Цикл for

```
for (<объявление счётчика>;<условие выхода из
цикла>;<тело for>) {
    <инструкция1>;
}
```

- Порядок работы цикла **for**:
- <объявление счётчика>
- <проверка условия выхода из цикла>
- <выполнение тела цикла>
- <выполнение тела **for**>

```
for (;i<10;i++) {  
    printf(“%d”,i);  
}
```

```
for (;i>10;) {  
    printf(“%d”,i);  
    i++;  
}
```

```
for (;true;) {  
    <инструкция1>;  
}
```

# Оператор break

- Команда досрочного выхода применяется, когда необходимо прервать выполнение цикла, в котором условие выхода ещё не достигнуто. Такое бывает, например, когда при выполнении тела цикла обнаруживается ошибка, после которой дальнейшая работа цикла не имеет смысла.

- `while(i<10)`
- `{`
- `i++;`
- `if (funct(i)==0) break;`
- `<операторы>`
- `}`

# Оператор continue

- Данный оператор применяется, когда в текущей итерации цикла необходимо пропустить все команды до конца тела цикла. При этом сам цикл прерываться не должен, а условия продолжения или выхода должны вычисляться обычным образом.

- `for (int i = 0 ; i < 10; ++i)`
- `{`
- `sum += i;`
- `if (i %2 == 1) continue;`
- `sum2 += i;`
- `}`

# Использование логических переменных в циклах

- `int i=1;`
- `while (i)`
- `{`
- `<операторы>`
- `}`

# Практическая работа

- Написать программу которая будет n раз считывать символ и в зависимости от него выполнять следующие действия:
- “a” – вывести “message 1”
- “n” – вывести номер обрабатываемого символа, введение данного символа не засчитывается в общий счётчик символов.
- “c” – считать символ и вывести его 3 раза через “,”
- Все остальные символы – вывести “unknown command”
- N считать при запуске, реализовать интерфейс.