Lesson\_4

|  |  |
| --- | --- |
| Коллеги, здравствуйте, рад всех приветствовать на четвёртом уроке по основам языка С.  На этом уроке мы познакомимся с таким базовым понятием программирования, как циклы.  Цикл - это одно или несколько действий повторяющихся до тех пор пока не наступит условие, прекращающее эти действия. |  |
| Базовый цикл в языке С записывается при помощи ключевого слова while после которого в круглых скобках пишется условие. При истинности данного условия будет выполняться тело цикла, которое в свою очередь пишется в фигурных скобках. Общий внешний вид очень похож на условный оператор из предыдущего видео, с той лишь разницей, что по окончании выполнения операторов внутри фигурных скобок мы переходим не на следующую строку, а обратно в проверку условия  Выведем на экран все числа в заданном промежутке, границы которого мы обозначим как а и b. Для этого нам необходимо их инициализировать, то есть объявить и задать начальные значения, и пока а меньше б заходить в тело цикла где мы будем выводить на экран и инкрементировать меньшее число, пока оно не станет равным второму. Как только числа сравняются условие входа в тело цикла перестанет быть истинным, и мы не зайдём в него, оказавшись на следующей после тела цикла строке. Один проход тела цикла и возврат в управляющую конструкцию называется итерацией, Этот термин можно очень часто услышать в беседах программистов. *Запустим нашу программу и убедимся что все работает.*  В нашей программе мы использовали оператор инкремента в постфиксной записи, которая означает, что значение переменной а сначала будет передано функции вывода на экран, а уже потом увеличено на единицу. Также вы можете использовать префиксную запись оператора инкремента, при которой, как не сложно догадаться, значение сначала будет увеличено, а уже потом передано функции вывода на экран. Запустим снова, чтобы увидеть разницу, были значения 10-19, стали 11-20. | СЛАЙД?  int a = 10;  int b = 20;  while (a < b){  printf(“%d”, a++);  }  ++а |
| Давайте напишем еще один пример, в котором подсчитаем сколько чётных чисел на промежутке от a до b. для этого нам понадобится циклически пройтись по всем числам от а до b и в каждой итерации цикла (то есть для каждого числа) сделать проверку, является-ли число чётным. Если является -увеличить счётчик положительных чисел для заданного промежутка.  Цикл while используется когда мы не можем достоверно сказать сколько итераций нам понадобится выполнить. *На этом примере мы видим что тело цикла может быть любой сложности*. *Оно может содержать как условные операторы так и другие циклы*. После того как выполнится условие в скобках и наш цикл будет завершён выведем в консоль количество чётных чисел.  Запустим нашу программу и убедимся в правильности ее работы. Программа выдала результат – 5 четных чисел, давайте пересчитаем вручную:  10 12 14 16 18, значение 20 не войдет в результат работы программы, поскольку на последней итерации, когда а = 20 условие в круглых скобках окажется ложным – 20 не меньше 20. А раз условие не выполнено мы не попадаем в тело цикла | int a = 10;  int b = 20;  int even = 0;  while (a < b){  if(a%2 == 0)  even++;  a++;  }  printf(“There are %d even numbers in sequence”, even) |
| Простой цикл while, по очевидным причинам, попадает в категорию циклов с предусловием. Сначала программа проверяет условие, затем по результатам этой проверки либо выполняет либо не выполняет тело цикла.  Ну а раз есть циклы с предусловием, то логично предположить наличие циклов с постусловием. Как ни удивительно, в языке С есть и такой. Это разновидность цикла while, который записывается как ключевое слово do {тело цикла} while (условие для очередной итерации цикла);  В этой конструкции сначала гарантированно один раз выполнится тело цикла, и только потом будет принято решение нужно ли выполнять его снова. | СЛАЙД? |
| Данные циклы широко применяются для проверки пользовательского ввода до получения приемлемого результата и для ожидания ответов на запросы, что логично, поскольку глупо было бы ожидать ответ, не послав запрос.  Например, мы пишем калькулятор, и знаем, что числа нельзя делить на ноль. Предположим, что наш пользователь этого не знает. Что ж, будем предлагать пользователю ввести делитель, пока он не введёт что-то отличное от нуля. Если пользовательский ввод равен нулю, значит нам нужно спросить его снова. Это и будет условием для очередной итерации цикла. Когда пользователь введёт удовлетворяющее нашему условию число произведём необходимые подсчёты и выведем их результаты в консоль.  *Запустим нашу программу. Каждый раз когда мы вводим 0 программа будет повторно задавать нам вопрос. Когда введем любое другое число получим результат нашего деления.* | do{  printf(“Enter a divider for 100 (remember. that you can’t divide by zero)\n”);  scanf(“%d”, &input);  }  while (input == 0);  printf(“100 / %d = %d”, input, 100/input); |
| С помощью циклов в программировании выполняются все рутинные задачи, такие как поиск значений в больших наборах данных, создание разного рода прогрессий, построение графиков, сортировки, ожидание ответов на запросы, чтение потоков данных и многие другие. |  |
| До встречи на следующем уроке. |  |