

# Программирование на C++



Минцифры  
России

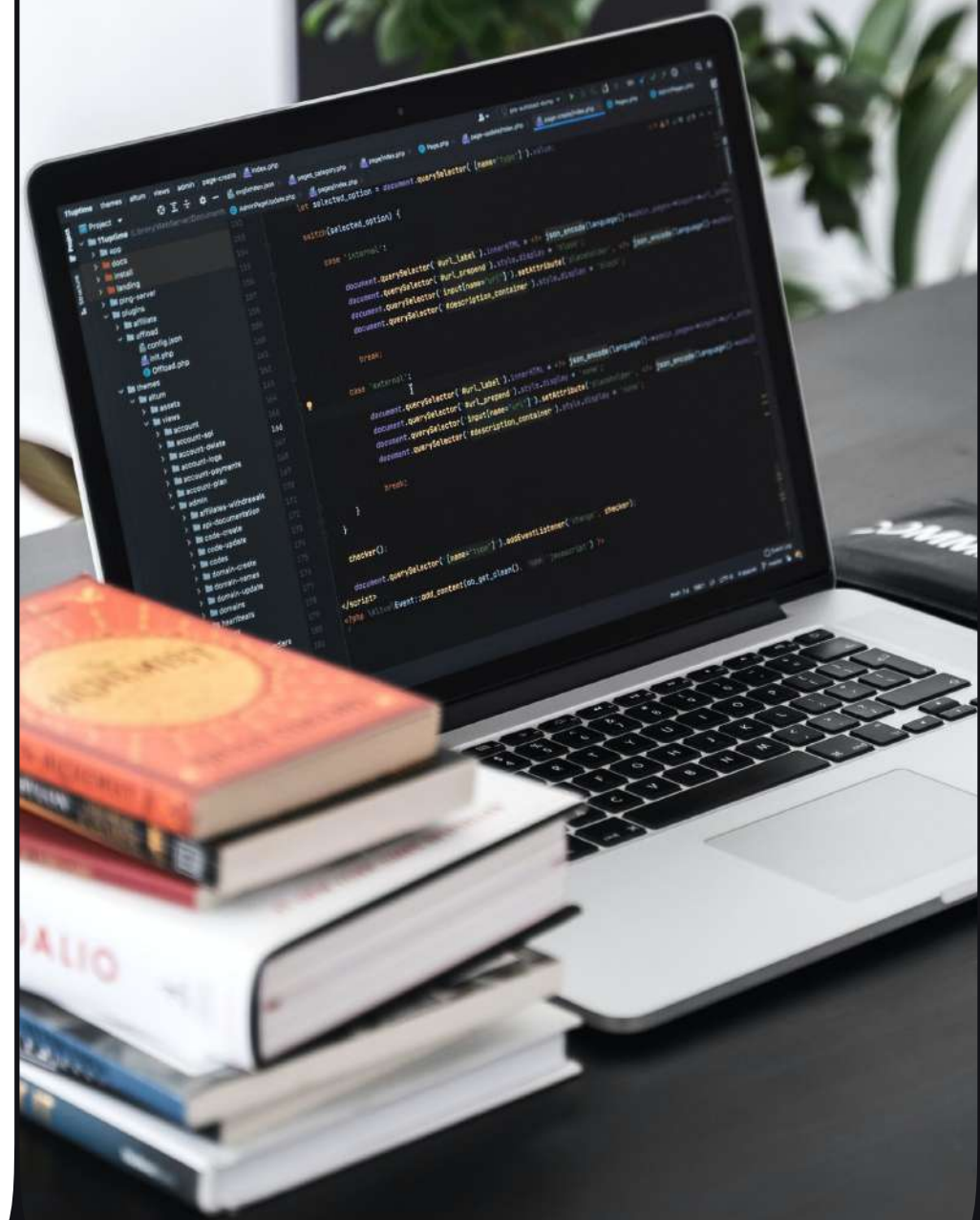
UCHi **DOMA**

**20.35**  
УНИВЕРСИТЕТ

Урок 9 Модуль 1

# Параметрический цикл

Полезные материалы



# Задача



Вывести 5 раз, слово «Привет»

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int k=0;
5      while (k<5)
6      {
7          ... k++;
8          printf("Привет\n");
9      }
10     return 0;
11 }
```

создание переменной-счётчика

условие цикла по счётчику

увеличение значения счётчика на 1

?

Приходится создавать специальную переменную и вручную менять и отслеживать её значение.

# Цели урока



изучить оператор  
параметрического цикла  
(повторения)



отработать на практике  
составление алгоритмов  
с параметрическим  
циклом на Си

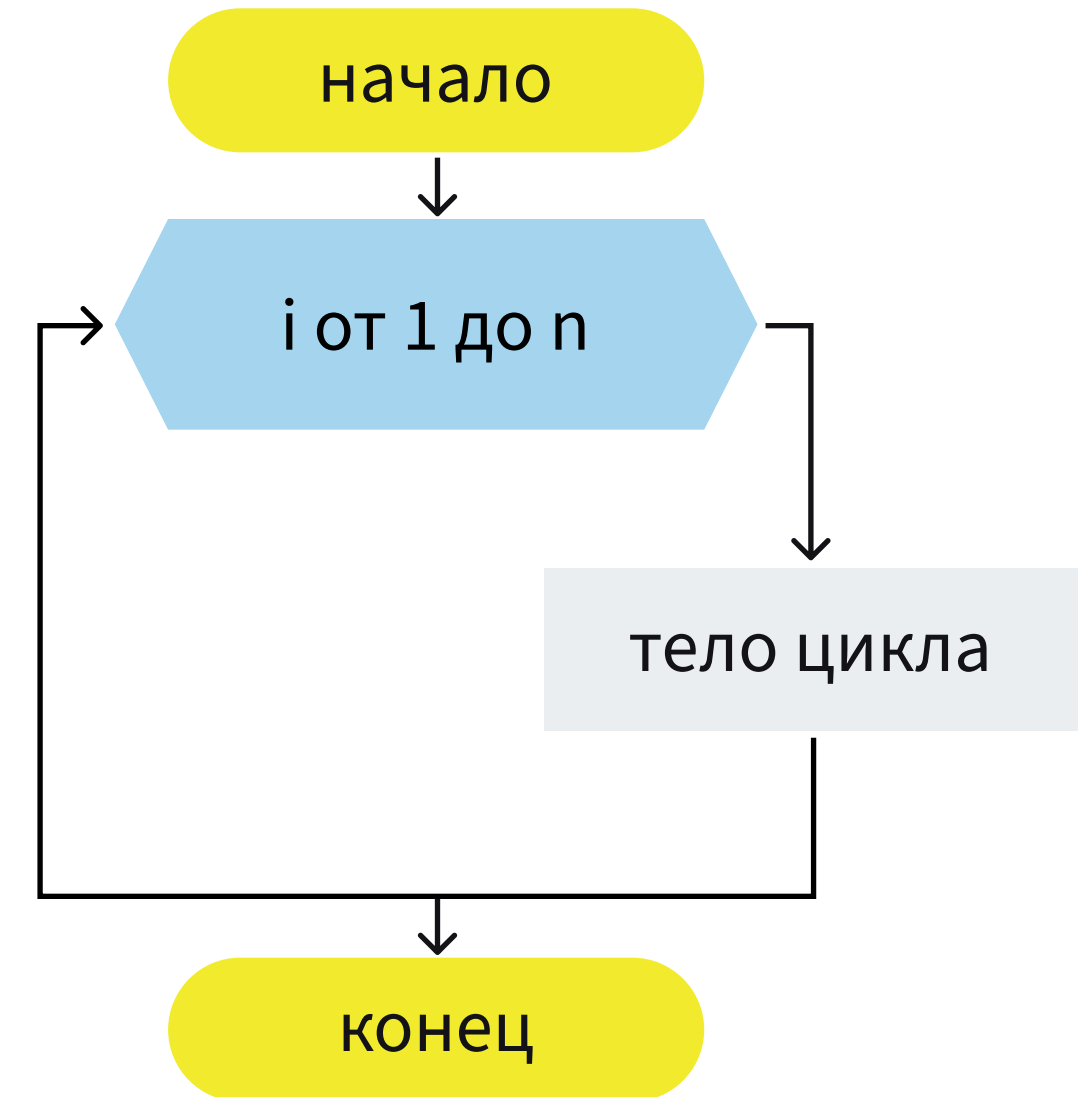


# Параметрический цикл

```
1  for (Инициализация; Условие; Модификация)
2  {
3      тело цикла
4  }
```

**for** — параметрический цикл (цикл с фиксированным числом повторений). Для организации такого цикла необходимо осуществить три операции:

- ✦ **Инициализация** — присваивание параметру цикла начального значения
- ✦ **Условие** — проверка условия повторения цикла
- ✦ **Модификация** — изменение значения параметра для следующего прохождения тела цикла



# Параметрический цикл

```
1  for (Инициализация; Условие; Модификация)
2  {
3      тело цикла
4  }
```



**Инициализация** параметра осуществляется только один раз — когда цикл **for** начинает выполняться.



Проверка **Условия** повторения цикла осуществляется перед каждым возможным выполнением тела цикла.



Когда выражение, проверяющее **Условие** становится ложным (равным нулю), цикл завершается.



**Модификация** параметра осуществляется в конце каждого выполнения тела цикла. Параметр может как увеличиваться, так и уменьшаться.

# Пример

```
1  #include <stdio.h>
2  int main ( )
3  {
4      for (i=0; i<5; i++)
5      {
6          printf("Привет\n");
7      }
8      return 0;
9  }
```

Пример работы программы:

Привет  
Привет  
Привет  
Привет  
Привет

# Пример

Программа запрашивает число, пока сумма введенных чисел меньше 100

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int a=4, sum=0;
5      while (sum<100)
6      {
7          ... scanf("%d",&a);
8          ... sum+=a
9      }
10     return 0;
11 }
```

Результат работы программы:

10

20

50

40



# Пример

Вывести на экран цифры от 1 до 5

```
1  #include <stdio.h>
2  int main ( )
3  {
4      for (i=1; i<6; i++)
5      {
6          printf("%d\n",i);
7      }
8      return 0;
9  }
```

Результат работы программы:

```
1
2
3
4
5
```

# Пример

Вывести на экран сумму цифр от 1 до 10

```
1  #include <stdio.h>
2  int main ( )
3  {
4      int sum=0;
5      for (int i=1; i<11; i++)
6      {
7          ...    sum=sum+1
8      }
9      printf("%d\n",sum);
10     return 0;
11 }
```

Результат работы программы:

55

# Рекомендации по выбору цикла



**while** — используется, когда заранее не известно количество повторений, выполнение тела цикла зависит от верности условия. Переменные, участвующие в выражении, проверяющем условие, проинициализированы заранее вне тела цикла.



**do...while** — используется, когда заранее не известно количество повторений, выполнение тела цикла зависит от верности условия. Для проверки условия требуется вычислить значение выражения, которое затем будет размещено в теле цикла.



**for** — используется, когда заранее известно количество повторений.

# Вложенный цикл

Вложенные циклы — когда один цикл находится внутри другого

```
1  for (i=0; i<n; i++) // внешний цикл - Цикл1
2  {
3      for (j=0; j<n; j++) // вложенный цикл - Цикл2
4      {
5          ;           // тело цикла Цикла2
6      }
7      // тело цикла Цикла1;
8  }
```

# Пример

Вывести числа от 0 до 99, по 10 в каждой строке

```
1  #include <stdio.h>
2  int main ( ) {
3      for(int i=0; i<10; i++) // цикл для десятков
4      {
5          for (int j=0; j<10; j++) // цикл для единиц
6          {
7              printf("%2d", i*10+j); // выводим вычисленное число (2 знакоместа) и пробел
8          }
9          printf("\n"); // во внешнем цикле переводим строку
10     }
11     return 0;
12 }
```

# Пример

Результат работы программы:

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |

# Операторы прерывания и продолжения цикла

## break

оператор прерывания цикла



Оператор break позволяет выйти из цикла, не завершая его.

## continue

оператор продолжения цикла



Оператор continue позволяет пропустить часть операторов тела цикла и начать новую итерацию.

# Пример break

Вывести числа от 0 до 99 ниже главной диагонали

```
1  #include <stdio.h>
2  int main ( ) {
3      for(int i=0; i<10; i++) // цикл для десятков
4      {
5          for (int j=0; j<10; j++) // цикл для единиц
6          {
7              if (j>i) // если число единиц больше числа десятков в числе
8                  break; // выходим из вложенного цикла и переходим к новой строке
9              printf("%2d", i*10+j); // выводим вычисленное число (2 знакоместа) и пробел
10         }
11         printf("\n"); // во внешнем цикле переводим строку
12     }
13     return 0;
14 }
```



# Пример

Вывести числа от 0 до 99 ниже главной диагонали

Результат работы программы:

```
0
10 11
20 21 22
30 31 32 33
40 41 42 43 44
50 51 52 53 54 55
60 61 62 63 64 65 66
70 71 72 73 74 75 76 77
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98 99
```

# Пример continue

Вывести числа от 0 до 99 исключая числа, оканчивающиеся на 3 или 7

```
1  #include <stdio.h>
2  int main ( ) {
3      for(int i=0; i<10; i++) // цикл для десятков
4      {
5          for(int j=0; j<10; j++) // цикл для единиц
6          {
7              if ((j==3) || (j==7)) // если число единиц в числе равно 3 или 7,
8                  continue;        // переходим к следующей итерации цикла
9              printf("%2d", i*10+j); // выводим вычисленное число (2 знакоместа) и пробел
10         }
11         printf("\n"); // во внешнем цикле переводим строку
12     }
13     return 0;
14 }
```

# Пример continue

Вывести числа от 0 до 99 исключая числа, оканчивающиеся на 3 или 7

Результат работы программы:

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 4  | 5  | 6  | 8  | 9  |
| 10 | 11 | 12 | 14 | 15 | 16 | 18 | 19 |
| 20 | 21 | 22 | 24 | 25 | 26 | 28 | 29 |
| 30 | 31 | 32 | 34 | 35 | 36 | 38 | 39 |
| 40 | 41 | 42 | 44 | 45 | 46 | 48 | 49 |
| 50 | 51 | 52 | 54 | 55 | 56 | 58 | 59 |
| 60 | 61 | 62 | 64 | 65 | 66 | 68 | 69 |
| 70 | 71 | 72 | 74 | 75 | 76 | 78 | 79 |
| 80 | 81 | 82 | 84 | 85 | 86 | 88 | 89 |
| 90 | 91 | 92 | 94 | 95 | 96 | 98 | 99 |