

**Лекція 5.
Цикли.
Цикл while**

1. Циклічні алгоритми

Задача.

Скласти алгоритм фарбування паркану



Питання:

- скільки дошок має паркан?

- 10 дошок

- тоді алгоритм такий:

1. Підійти до лівої дошки паркану
2. Пофарбувати одну дошку
3. Зробити один крок вправо на ширину дошки
4. Пофарбувати одну дошку
5. Зробити один крок вправо на ширину дошки
6. Пофарбувати одну дошку
7. ...

Питання:

- а якщо 100 дошок?
- а якщо 200? 500? 1000?
- тоді в алгоритмі буде 200, 500 або 1000 однакових дій

В програмуванні:

якщо треба виконати якусь дію **N** разів, можна розмістити **N** однакових рядків програмного коду:

```
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");
```

*5 раз
виконається
однаковий
оператор*

Проблеми такого підходу:

1) щоб повторити операцію багато разів, наприклад, 1000, доведеться писати 1000 однакових рядків

2) якщо потрібно буде внести правку у повторюваний оператор, то доведеться правити 1000 однакових рядків

Важливий принцип програмування:

У програмі не має бути однакових фрагментів програмного коду, або їх має бути максимально мало

```
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");  
printf("Привіт! ");
```

*так
робити
не треба*

Для повтору дії (оператора) використовують **цикли**.

Цикл – багаторазове повторення виконання операторів.

Цикл повинен мати **умову продовження (завершення) циклу**


Інакше відбудеться **зациклювання** (зависання програми)

В мові Сі є три оператори циклів:

1) оператор **while**

2) оператор **for**

3) оператор **do ... while**



*цикли з
передумовою*



*цикл з
постумовою*

2. Оператор циклу while

while (умова)

оператор;



тіло циклу

*буде повторюватися, доти, доки
умова буде істинною*

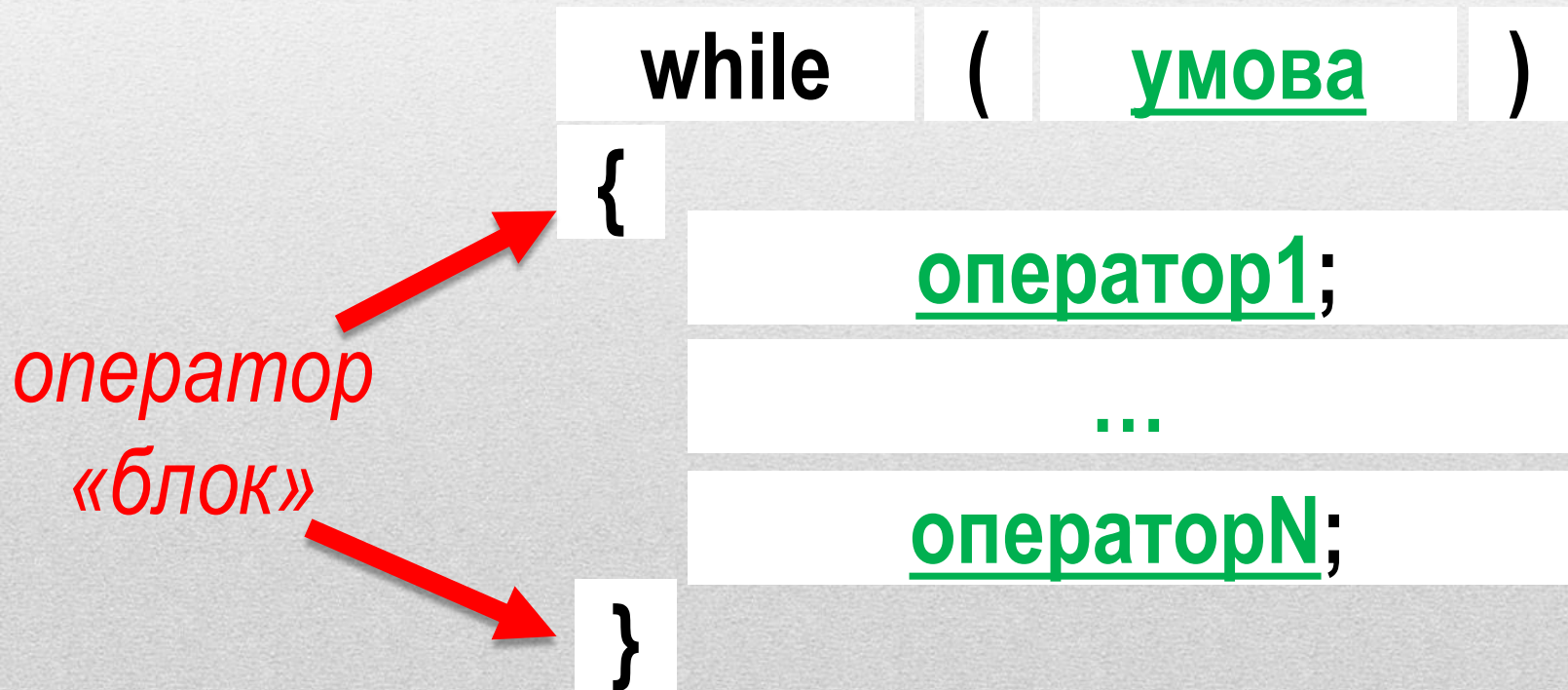
```
while ( умова ) оператор;
```

Цикл працює так:

1. Обчислюється вираз умови.
2. Якщо умова **істинна** (вираз умови не дорівнює нулю), то виконується **оператор** і перехід до пункту 1.
3. Якщо умова **хибна**, то здійснюється вихід з циклу.

В більшості випадків, потрібно повторити виконання не одного оператора, а кількох.

Тоді використовують **оператор «блок»**:



Приклад 1. Знайти суму парних чисел від 2 до 10 (включно).

```
int summa = 0;  
int a = 2;  
while (a <= 10)
```

*умова
виконання
тіла циклу*

```
{  
    summa = summa + a;  
    a = a + 2;  
}
```

*тіло
циклу*

```
printf("Summa = %d\n", summa);
```

Приклад 2. Користувач вводить N чисел з клавіатури. Порахувати суму та добуток цих чисел.

Спершу почнемо з роздумів над завданням

Змоделюємо процес роз'язання даної
задачі людиною:

Числа:	3	7	0.5	-2
Добуток:	3	21	10.5	-21
Сума:	3	10	10.5	8.5

і так далі ...

1) оскільки не сказано, які саме числа, то передбачаємо найбільш загальний випадок – коли числа **дробові**

2) сума та добуток дробових чисел будуть **дробовими**

3) на початку роботи програми попросимо користувача ввести кількість чисел

Кількість – це ціле число, оскільки користувач не може ввести 1.5 (півтора числа) чи 7.25 (сім цілих двадцять п'ять сотих).

Записуємо програмний код:

```
#include <stdio.h>
int main()
{
    int n;
    double summa = 0, dob = 1, x;
    printf("n = "); scanf("%d", &n);
    while (n > 0)
    {
        printf("x = "); scanf("%lf", &x);
        dob = dob * x;
        summa = summa + x;
        n--;
    }
    printf("Summa = %f\nDob = %f\n", summa, dob);
}
```

сума повинна бути ініціалізована нулем

добуток ініціалізується одиницею

поки n більше нуля, повторюємо тіло циклу; якщо n стало рівне 0, тоді завершуємо цикл

змінна n – це кількість чисел, які вводить користувач; ми будемо позначати під n скільки ще чисел треба ввести; ввели число – зменшили n

Приклад 3. Користувач вводить N чисел з клавіатури. Знайти мінімальне та максимальне з них.

Почнемо з роздумів над завданням

1) введемо позначення для змінних і виберемо для них типи даних:

n – кількість чисел, які планує ввести користувач (ціле число);

i – міститиме кількість чисел, які вже ввів користувач (ціле число);

x – число, яке вводить користувач (дробове число);

min – мінімальне число (дробове число);

max – максимальне число (дробове число);

Записуємо програмний код:

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

```
int main(){
```

```
int n, i = 0;
```

змінна *i* виступатиме **лічильником** (буде використовуватися для підрахунку кількості введених чисел); поки числа не почали вводитися, *i* дорівнює **нулю**

```
double min = 1e7, max = -1e7, x;
```

записуємо у змінну *min* нескінченність, а в *max* – мінус нескінченність

```
printf("n = "); scanf("%d", &n);
```

```
while (i < n){
```

повторюємо процес, поки *i* менше за *n*; коли *i* стане $\geq n$, вийдемо з циклу

```
printf("x = ");
```

```
scanf("%lf", &x);
```

якщо введене число *x* більше попереднього значення *min*, то число *x* є мінімумом

```
if (x < min) min = x;
```

```
if (x > max) max = x;
```

аналогічно з максимумом

```
i++;
```

після того, як було введено число, збільшуємо лічильник *i* на одиницю

```
printf("Min = %f\nMax = %f\n", min, max);
```

```
}
```

Вивести на екран числа від 0 до 9

```
#include<stdio.h>

int main() {
    int i = 0;

    while (i < 10) {
        printf("%d\n", i);
        i++;
    }
    return 0;
}
```

?

```
int i = 0;
while (i < 10) {
    printf("%d\n", i);
}
```

```
int i = 0;
while (i <= 9)
    printf("%d\n", i++);
```

```
int i = 0;
while (i > 0) {
    printf("%d\n", i);
    i++;
}
```

```
int i = 0;
while (i < 10)
    printf("%d\n", i++);
```

```
int i = 0;
while (i++ < 10) {
    printf("%d ", i);
}
```

```
int i = 0;
while (++i < 10) {
    printf("%d ", i);
}
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
Press any key to continue . . .
```

```
#include<stdio.h>

int main() {
    int i = 1, j = 1;
    while (i < 10) {
        while (j < i) {
            printf("%c", '*');
            j++;
        }
        printf("\n");
        j = 1; i++;
    }
    return 0;
}
```

Знайти максимальне
число у введеній
послідовності, при
умові виходу із циклу
- число 0.

```
#include<stdio.h>

int main() {
    int num = 0;
    int max = num;
    while (1) {
        printf("Please, enter number: ");
        scanf("%d", &num);
        /*умова виходу із циклу*/
        if (num == 0) break;
        if (num > max) max = num;
    }

    printf("max number was %d\n", max);
    return 0;
}
```