

## Кайталануучу алгоритмдерди программалоо

Кайталануучу алгоритмдерди C++ тилинде программалоо үчүн үч түрдөгү цикл оператору каралган. Аларга кайталануунун саны берилген цикл, шарттан кийинки цикл жана шартка чейинки цикл операторлору кирет. Кайталануунун саны берилген цикл операторунда кайталанууну уюштурган өзгөрмөнү параметр деп аташат. Ошондуктан бул цикл операторун параметрлүү цикл деп да коюшат.

1) Параметрлүү цикл операторунун жалпы берилиши. for (n1;n2;n3)

<оператор>;

n1-параметрдин баштапкы мааниси.

n2-кайталанууну уюштуруучу шарт б. а. циклден чыгууну же улантууну текшерүүчү шарт.

n3-параметрдин өзгөрүү кадамы.

<оператор> - циклдин тулкусу деп аталат б. а. циклдин кайталануучу бөлүгү.

Циклдин тулкусу жөнөкөй же курама болушу мүмкүн. Циклдин тулкусу курама болсо б. а. бирден көп операторлорду же функцияларды кармаса анда циклдин тулкусу фигуралык кашаага алынып жазылат. Көпчүлүк учурда циклдин параметрин i -тамгасы менен белгилешет.

1) Параметирдин мааниси өсүү тартибинде, же кемүү тартибинде болушу мүмкүн.

*Мисалы.* 10дон 1ге чейин бүтүн сандардын көбөйтүндүсүн тап?

```
#include <iostream>
```

```
#include <conio.h>
```

```
main ( )
```

```
{int i,p;
```

```
clrscr();
```

```
p=1;
```

```
for (i=10; i>=1; i--) p*=i;
```

```
cout<<"p=",p;
```

```
}
```

Жыйынтык: p=24320

2) Циклдин параметиринин өзгөрүш кадамы ар кандай болушу мүмкүн.

```
for (i=2; i<90; i+=7)<оператор>;
```

*Мисалы.* 5ке так бөлүнгөн эки орундуу сандардын суммасын тапкыла.

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

```
#include <conio.h>
```

```
main ( )
```

```
{int i,s ;
```

```
clrscr();
```

```
s=0;
```

```
for (i=10 ; i<=99; i+=5)
```

```
s+=i;
```

```
cout<<"s=",s;
```

```
}
```

Жыйынтык: s=945

3) Циклдин параметри катары символдорду колдонууга болот. for (char ch='a';

```
ch<='z'; ch++)<оператор>;
```

4) n3-өзгөрүү катары параметирди кармаган туюнтманы колдонсо болот.

```
for(x=1; y<=50; y=3*x++ +5) cout<<"x="<<x<<"y="<<y);
```

Бул мисалда параметирдин өзгөрүү кадамы катары  $y=3*x++ +5$  туюнтмасындагы  $x++$  чоңойтуучу амал колдонулду.

5) Цикл операторунун жалпы жазылышындагы n3- параметирдин өзгөрүү кадамы жазбай койсо болот.

*Мисалы;* int  
i,n; n=3;  
for(i=5; n<=100;)n\*=i;

Бул *мисалда* параметирдин өзгөрүү кадамы циклдин тулкусуна т.а n\*=i: операторунда өзгөрөт. Жогорудагы цикл операторун башкача түрүн жазса да болот.

int i,n; n=3;  
for (i=5; n<=100; n\*=i);

Бул жазууда циклдин тулкусу куру оператор менен жазылып калды.

б) Цикл операторунда n1 параметири бир гана жолу аткарылгандыктан, дайыма эле өзгөрмөнүн ысымын кармай бербейт.

*Мисалы:*

for (printf (“ сан киргиз\n”); n<=5;scanf (“%d”,&n);

бул мисалда n дин маанисин киргизүүдө 5тен чоң сан болмоюнча цикл аткарыла берет.

7) үтүр (,) - белгиси параметирлүү цикл операторунда, параметирлердин ордуна, бир канча туюнтмаларды жазууда ажыратуу белгиси катары колдонулат.

*Мисалы:*

for (i=0,j=50; i<j ; i++,j--);

бул мисалда циклдин өзгөрүү параметири катары кош i, j параметирлери колдонулган.

### **Шарттан кийинки кайталануучу оператор**

Шарттан кийинки кайталануучу оператордун жалпы түрүнүн берилиши *while(<туюнтма>)<циклдин тулкусу>;* Бул оператордун аткарылышы адегенде туюнтманын мааниси эсептелинет, эгерде ал чын болсо <циклдин тулкусу> аткарылат ал эми жалган болсо иш аракет цикл операторунан кийинки операторго берилет. <циклдин тулкусу> жөнөкөй, курама же куру операторлор болушу мүмкүн. Чексиз кайталануу болбошу үчүн <туюнтманын> маанисин өзгөртүп туруучу иш аракеттер <циклдин тулкусунда> болушу зарыл.

*Мисалы:* эки сандын эң чоң жалпы бөлүүчүсүн эсептөөчү программаны карайлы.

```
#include <iostream>
```

```
#include <conio.h>
```

```
main ()
```

```
{ int x,y;
```

```
clrscr();
```

```
cout<<“Эки бүтүн санды киргиз\n”; cin>>x>>y;
```

```
while (x!=y)
```

```
{ if (x>y)x-=y; else y-
```

```
=x;
```

```
}
```

```
printf (“Эки сандын эң чоң жалпы бөлүүчүсү=%d”,x);
```

```
}
```

**Жыйынтык:** Эки бүтүн санды киргиз

9,12

Эки сандын эң чоң жалпы бөлүүчүсү=3

### Шартка чейинки кайталануучу оператор

Жалпы жазылышты: *do*<циклдин тулкусу> *while* (<туюнтма>); Бул оператордун аткарылышы шарттан кийинки кайталануучу операторго эле окшош, бир айырмачылыгы

<циклдин тулкусу> сөзсүз бир жолу аткарылат.

1-Мисал. Берилген аралыкта, x санынын жатаарын аныктоочу праграмма.

```
#include <iostream>
#include <conio.h>
main()
{ int a,b,x;
clrscr();
cout<<"a жана b аралыгын бергиле\n";
cin>>a>>b;
do
{
cout<<"x санын киргиз\n";
cin>>x;
if(x<a|| x>b) cout<<"x берилген аралыкта жатпайт\n";
}
while(x<a||x>b);
cout<<"x саны берилген аралыкта жатат\n";
cout<<a<<x<<b;
}
```

Жыйынтык: a жана b аралыгын бергиле

5,9  
x санын киргиз  
7  
x саны берилген аралыкта жатат  
5<7<9

2-Мисал. Сандарды баскычтар аркылуу киргизип жана алардын суммасын эсептөөчү программа.

```
#include <iostream>
#include <conio.h>
main()
{
int sum,n; clrscr();
cout<<"бүтүн сан киргизүүнү башта\n";
cout<<"бүтүн сандарды киргизүүнү аяктоо үчүн 0дү басыңыз\n";
sum=0;
do
{
cout<<"бүтүн сан киргиз\n";
cin>>n;
sum+=n;
}while (n!=0);
cout<<"sum=",sum;
}
```

Жыйынтык: бүтүн сан киргизүүнү башта

бүтүн сандарды киргизүүнү аяктоо үчүн 0дү басыңыз бүтүн сан киргиз  
1  
бүтүн сан киргиз  
3

бүтүн сан киргиз

5

бүтүн сан киргиз

7

бүтүн сан киргиз

9

бүтүн сан киргиз

0

sum=25