

Оңтүстік Қазақстан мемлекеттік  
педагогикалық институты

# Презентация

Тақырыбы: Си тіліндегі операторлар

Орындаған: Сабырхан А.

Тобы: 111-15

Қабылдаған: Құлжатаева К.

ШЫМКЕНТ

- Си тілінің барлық элементтері келесі категорияларға шартты түрде бөліне алады:
- \* шартты операторлар, оған if шартты операторы және switch таңдау операторлары жатады;
- \* циклдік операторлар (for, while, do while);
- \* көшу операторлары (break, continue, return, goto);
- \* басқа операторлар (оператор “өрнек”, бос оператор).

- Програмадағы операторлар құрама операторларға фигуралық жақшалар көмегімен біріктіріле алады. Програмадағы кез келген оператор аттан және одан кейін келетін қос нүктеден тұратын таңбамен (метка) белгілене алады. Си тілінің барлық операторлары құрама операторлардан басқасы “;” үтір нүктемен аяқталады.

## Өрнек оператор (оператор выражение).

Үтір нүктемен аяқталатын кез келген өрнек оператор болып табылады. Өрнек оператордың орындалуында, өрнектің есептелуінде өрнектің алынған мәні ешбір пайдаланылмайды, сондықтан қағида бойынша мұндай өрнектер жанама эффектілерді тудырады. Мәндерді қайтармайтын функцияны тек өрнек оператордың көмегімен шақыруға болатынын байқайық. өрнектерді есептеу ережелері жоғарыда тұжырымдалған.

Мысалдар:

$++i;$

Бұл оператор  $i$  айнымалысының мәнін бірге өсіретін өрнекті көрсетеді.

$a=\cos(b*5);$

Бұл оператор өзіне меншіктеу және функцияны шақыру операцияларын қосатын өрнекті көрсетеді.

$a(x,y);$

Бұл оператор функцияны шақырудан тұратын өрнекті көрсетеді.

## Бос оператор

- Бос оператор тек үтір нүктеден тұрады. Бұл операторды орындау кезінде ештеңе болмайды. Ол әдетте келесі жағдайларда пайдаланылады:
- \* оператор орны керек етілмегенде жолдарда do, for, while, if операторларында, бірақ синтаксис бойынша ең болмағанда бір оператор талап етіледі;
- \* қажеттілік туғанда фигуралық жақша белгілеу.
- Си тілінің синтаксисі таңбадан кейін міндетті түрде оператор болғанын талап етеді. Фигуралық жақша оператор болмайды. Сондықтан да, егер басқаруды фигуралық жақшаға беру керек болса, бос операторды пайдалану қажетті.
- Мысал:

```
int main()
{
    :
    {if (...) goto a; /* жақшаға көшу */
    {...
a: ;}
    return 0;
}
```

## Құрама оператор

Құрама оператор фигуралық жақшаларға алынған бірнеше операторлар мен жариялаулар болып көрінеді:

```
{[жариялау]
:
оператор; [оператор];
:
}
```

Құрама оператордың соңына үтір нүкте қойылмайтынын ескерейік.

Құрама оператордың орындалуы оны құрайтын операторлардың жүйелі орындалуынан тұрады.

Мысал:

```
int main()
{
    int q,b;
    double t,d;
    :
    if (...)
    {
        int e,g;
        double f,q;
        :
    }
    :
    return(0);
}
```

## If операторы, switch таңдау операторы

- Оператордың форматтары:
- `if (өрнек) оператор-1;[else оператор-2;]`
- if операторының орындалуы өрнектің есептелуінен басталады. Әрі қарай орындалу келесі схема бойынша жүреді:
- \* егер өрнек ақиқат (яғни нөлден айрықша) болса, онда оператор-1 орындалады;
- \* егер өрнек жалған (яғни нөлге тең) болса, онда оператор-2 орындалады;
- \* егер өрнек жалған және оператор-2 жоқ болса (квадраттық жақшаға міндетті емес конструкция алынған), онда if-тен кейінгі орындалады.

# Мысалдар:

```
int main()
{
    int t=2, b=7, r=3;
    if (t>b)
    {
        if (b<r) r=b;
    }
    else r=t;
    return (o);
}
```





## switch таңдау операторы

- switch операторы әр түрлі варианттар жиынынан алынған таңдауды ұйымдастыру үшін арналған. Оператордың форматы келесі түрде:

```
switch (өрнек)
{[жариялау]
:
[case тұрақты өрнек 1]:[1 операторлар
тізімі]
[case тұрақты өрнек 2]:[2 операторлар
тізімі]
:
:
[default:[операторлар тізімі]]
}
```

- switch кілттік сөзінен кейінгі дөңгелек жақшаға алынған өрнек мәні бүтін Си тілінде мүмкін болатын кез келген өрнек бола алады.

## ● Мысал:

```
int i=2;
switch (i)
{
    case1: i+=2;
    case2: i*=3;
    case0: i/=2;
    case4: i-=5;
    default: ;
}
```

## **break операторы**

- break операторы оны біріктіретін switch, do, for, while операторларының ең ішкі орындалуының тоқтатылуын қамтамасыз етеді. break орындалғаннан кейін басқару келесі операторға беріледі.

## ● Мысал:

```
switch(a)
{
  case1:b=c; break;
  case2:
    switch(d)
      { case0:f=s; break;
        case1:f=9; break;
        case2:f-=9; break;
      }
  case3:b-=c; break;
  :
}
```