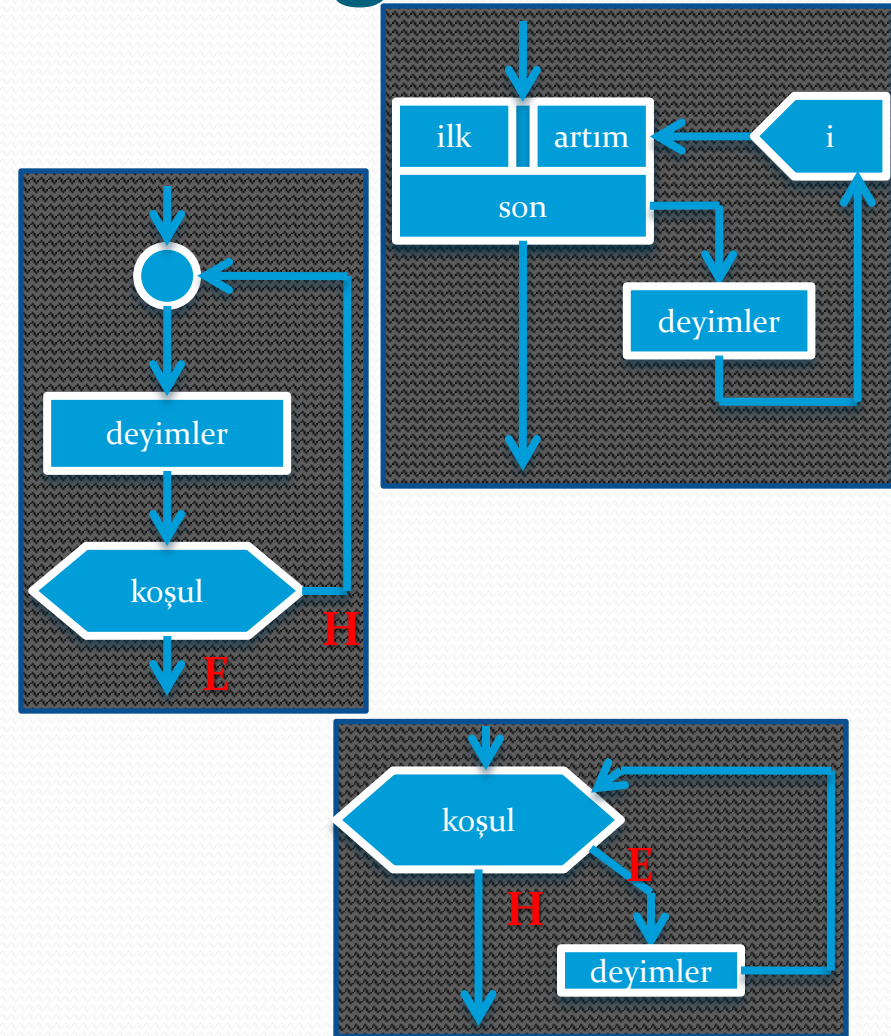


PROGRAMLAMA TEMELLERİ

Döngüler
for

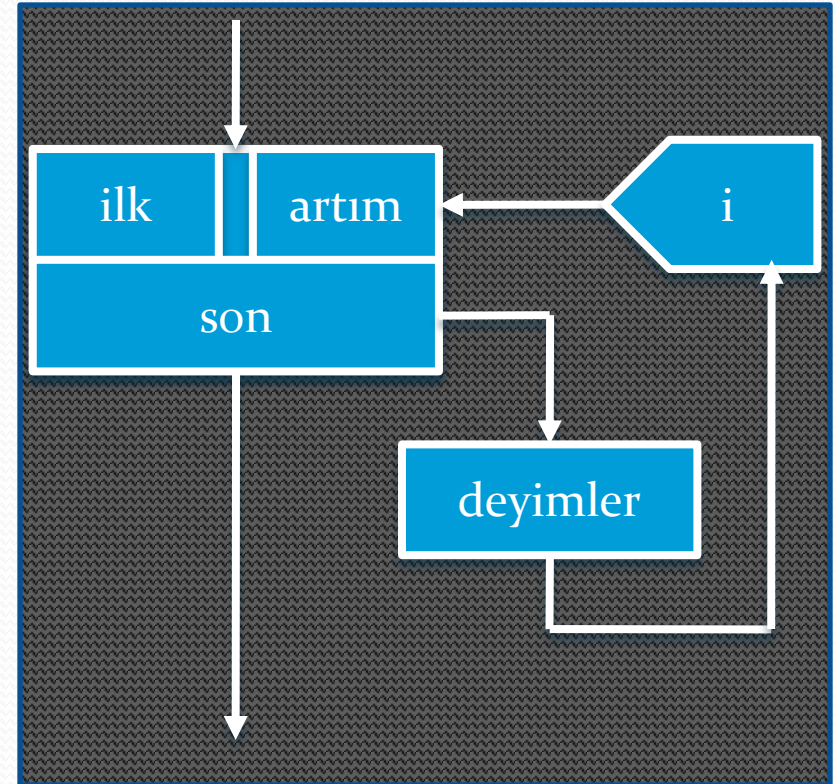
Kontrol Deyimleri: Döngüler

- Bazen komutların defalarca çalıştırılması gerekir.
- Yineleme bir koşula bağlı olabileceği gibi belli bir sayıda da olabilir.
- Döngüler:
 - Sabit Adımlı Döngü (for)
 - .. Olduğu Sürece (while)
 - .. Olmayana Kadar (do .. while..)



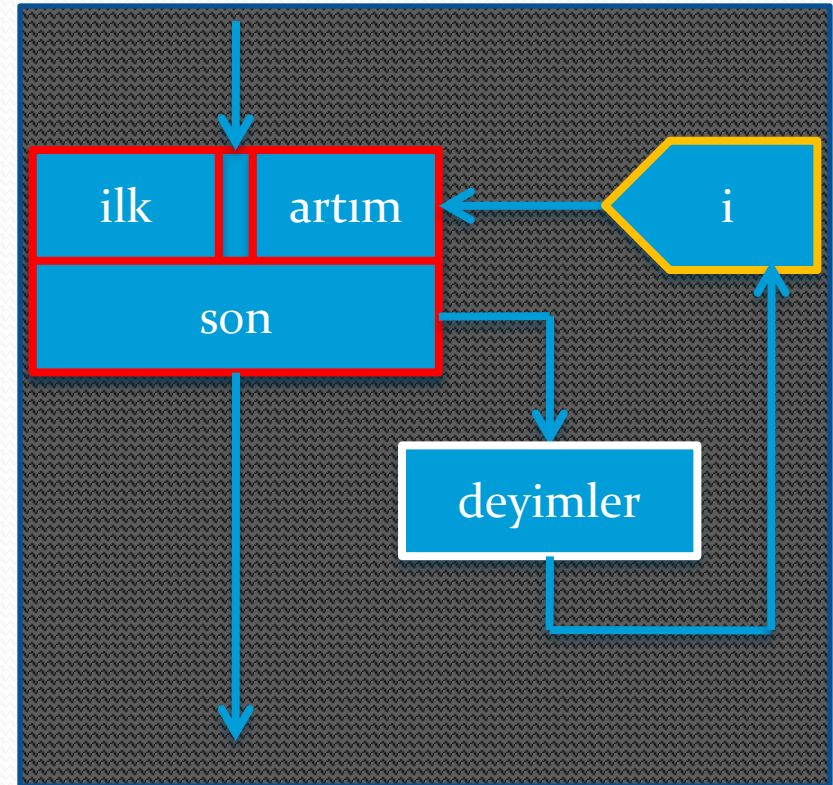
Döngüler: Sabit Adımlı Döngü

- Kaç kez tekrarlanacağı bellidir.
- Döngü değişkeni (Sayaç) başlangıç değerinden, bitiş değerine kadar verilen artım miktarı kadar adımla ilerletilir.
- Çoğunlukla dizilerle beraber kullanılır. Böyle durumda sayaç aynı zamanda indistir.



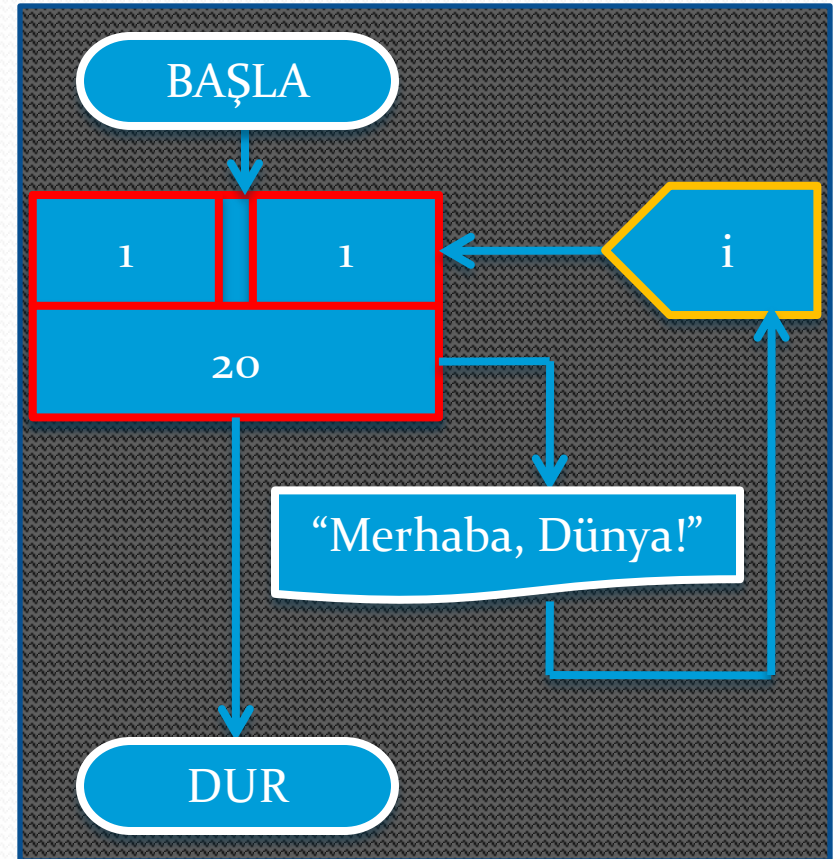
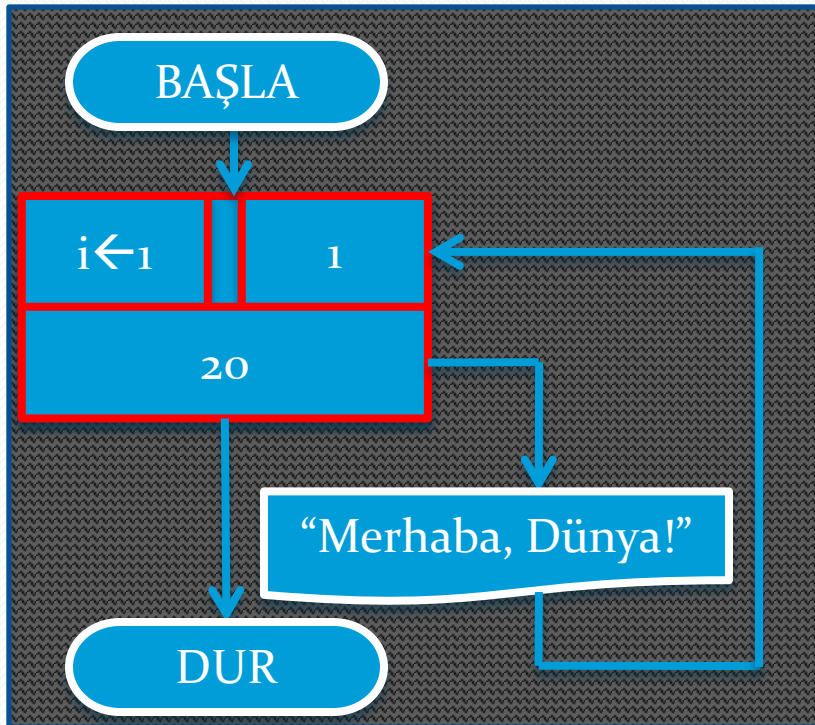
Döngüler: Sabit Adımlı Döngü

- Sabit adımlı döngü sembolü klakete benzer (kırmızı kenarlık). Üç bölmesi vardır:
 - İlk değer
 - Artım
 - Son değer
- Döngü sonunda sayaç sembolü (turuncu kenarlık) kullanılmayacaksa ilk değer yerine ilk değer ataması yapılır.



Döngüler: Sabit Adımlı Döngü

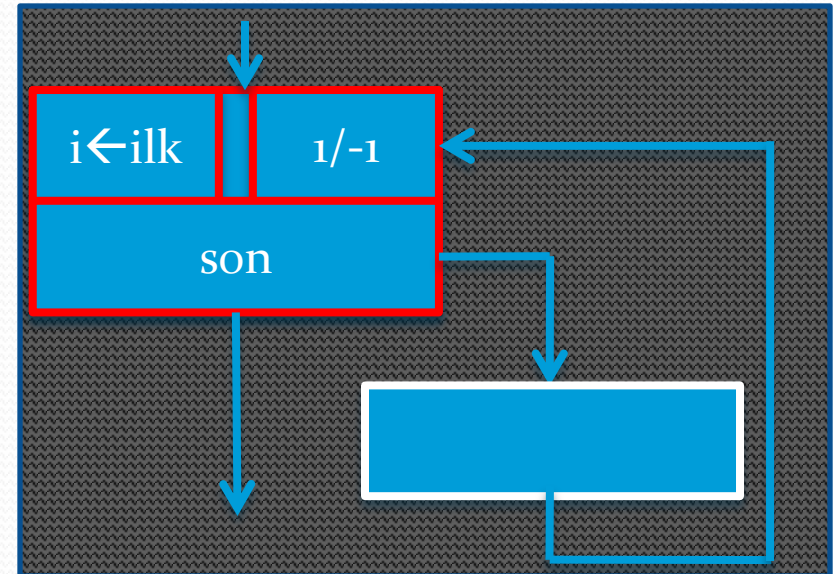
- Diyelim ki ekrana 20 defa “Merhaba, Dünya!” yazmak istiyoruz.



C++'ta Sabit Adımlı Döngü

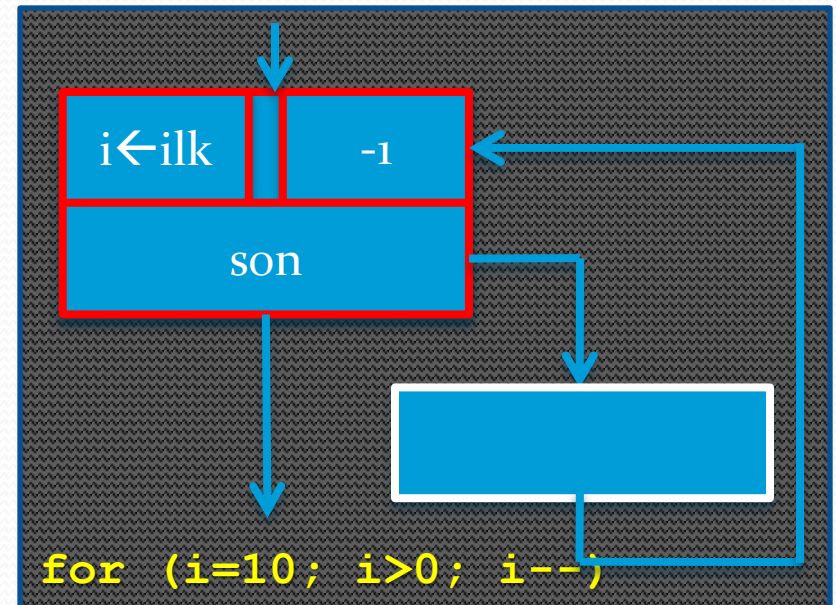
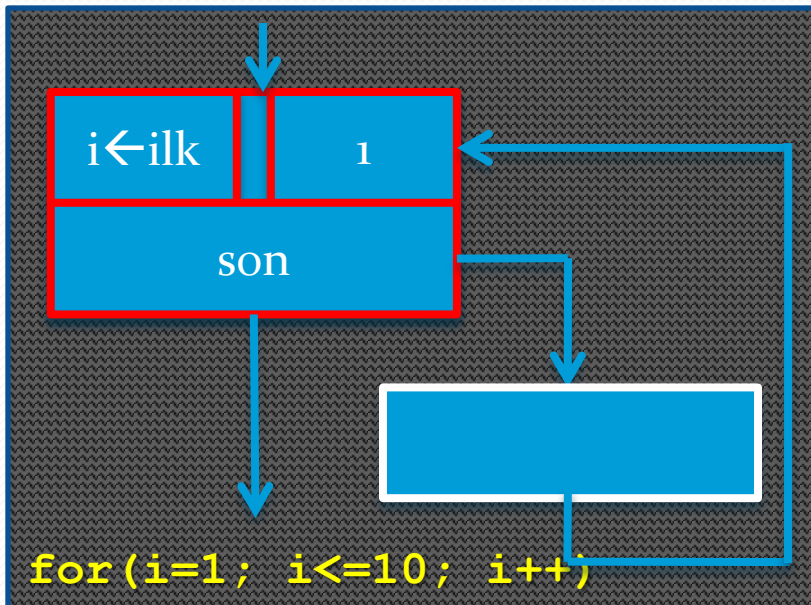
- Sabit adımlı döngüleri C++'ta «for» ile kodluyoruz.
- Yazım:

```
for (<ilk değer ataması> ; <koşul> ; <artım ifadesi>)  
{  
  <deyim/ler>;  
}
```



C++'ta for

- C++'ta for döngüleri diğer döngülerden bile daha esnek bir yapıya sahiptir.
- Eğer ileri doğru ise ($ilk < son$) artım pozitifdir, $i++$ gibi
- geriye doğruysa ($ilk > son$) artım negatifdir, $i--$ gibi



C++'ta for

- C++'ta «for» döngüsü şöyle çalışır:
 1. Önce ilk değer ataması bölümünde yer alan ifade/ler çalıştırılır
 2. Döngüye başında koşul kontrol edilir; koşul sağlanırsa döngü içerisindeki komutlar çalıştırılır. Sağlanmazsa çalıştırılmaz.
 3. Döngü komutları çalıştırılır.
 4. Artım bölümündeki ifade/ler çalıştırılır.
 5. Koşul bölümündeki koşul sağlanırsa 3. adıma gidilir, sağlanmazsa döngü sonlandırılır, takip eden komut çalıştırılır.

C++'ta for

- C++'ta «for» döngüleri diğer döngülerden bile daha esnek bir yapıya sahiptir.
- İstenirse «ilk değer ataması», «koşul» veya «artım» yazılmayabilir;

```
for(;;) { .. }
```

```
int i=0;
```

```
for( ; i<10; i++) { .. }
```

- İstenirse «ilk değer ataması», «koşul» veya «artım» bir veya daha fazla ifade aralarına «,» konarak yazılabilir;

```
for(i=0, j=0;;) { .. }
```

C++'ta for

- Artım veya azalım ifadesi illâ birer birer gelişmez

```
for( ; ; i+=2) { .. }
```

```
for( i=100; i>1; i-=j) { .. }
```

- Koşul illâ sayacı kontrol etmek zorunda değildir.
- «for» döngüsü istenirse «while» döngüsü olarak kullanılabilir.

```
for( ;<koşul>;) { .. }
```

```
while (<koşul>) { .. }
```

C++'ta for

- Döngü değişkeni (sayaç) döngü içerisinde değiştirilebilir.
- Ama sabit adımlı döngü olarak kullandığınızda sayacın değerinin döngü içerisinde değiştirilmesi adım sayısını bozabilir.

Örnek: 1

- Ekrana 20 defa “Merhaba, Dünya!” yazan C++ programı...

```
#include <cstdlib>
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
```

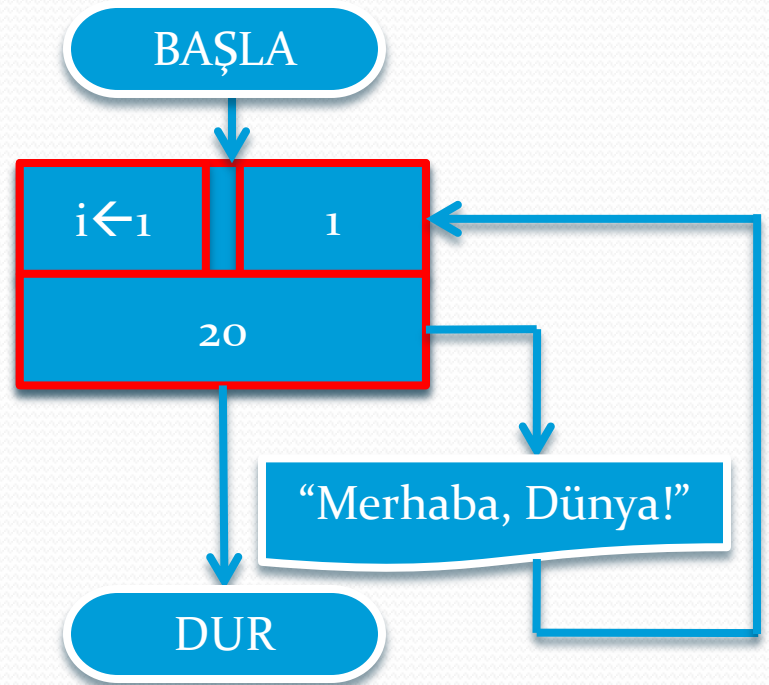
```
    for(int i=0; i<20; i++)
```

```
        cout<<"Merhaba, Dünya!"<<endl;
```

```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

```
}
```



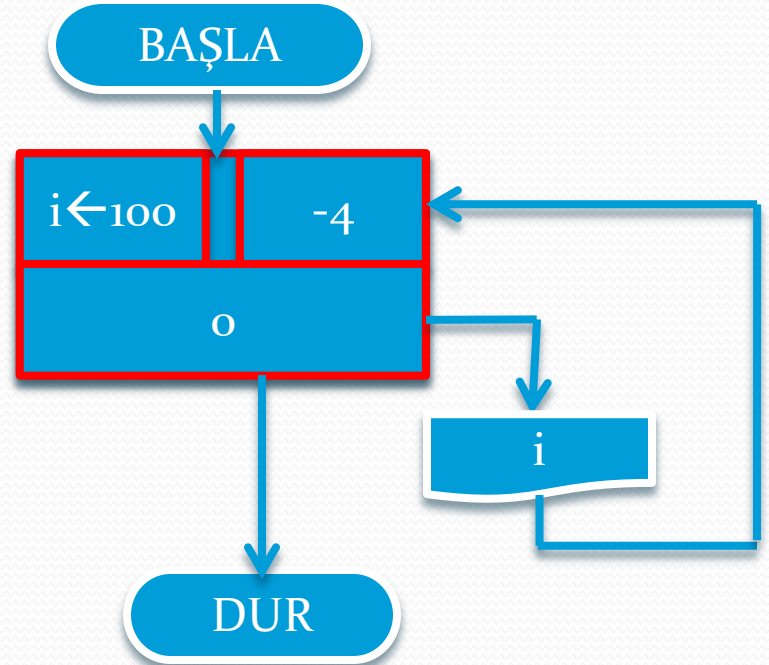
Örnek: 2

- 100'den geriye 4'er ritmik sayılar...

```
#include <cstdlib>
#include <iostream>
#include <iomanip>

using namespace std;

int main(int argc, char *argv[])
{
    for(int i=100; i>0; i-=4)
        cout<<setw(8)<<i;
    cout<<endl<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



Örnek: 3

- “n” adet sayının ortalaması...

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    double sayi, t=0.0, ort;
    int i, n;
    cout << "Kac adet sayi? ";
    cin>>n;
    for(i=1; i<=n; i++)
    {
        cout << i << ". sayi: ";
        cin >> sayi;
        t += sayi;
    }
    ort = t / n;
    cout << "Ortalama: " << ort << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

