

NUMARALANDIRMA ve TAKMA İSİMLER

Numaralandırma

C numaralandırması (*enumeration*), bir grup tamsayılardan oluşan (*integral*) değişmezden (*literal*) oluşan numaralandırılmış bir veri türüdür. Bir bütünün parçalarını tamsayı olarak ifade eden değişmezlere kullanıcı tanımlı adlar atamak istediğinizde kullanışlıdır. C dilinde numaralandırılmış sabitler “*integral type*” olarak adlandırılır ve **enum** anahtar kelimesiyle tanımlanırlar.

```
enum numaralandırmakimliği { DEĞER1, DEĞER2, ...};
```

Sözde kodda DEĞER1, DEĞER2 olarak belirtilenler aslında tamsayılara verilen isimlerdir. Yani tamsayı sabitlerdir. Sabit olduklarından büyük harfle yazılırlar ve belirtilmedikçe ilkinin değeri 0, ikincisinin değeri 1, ... şeklinde ilerler. Örnek;

```
#include <stdio.h>
enum cinsiyetKodlari {ERKEK = 1, KADIN = 2, BELIRSIZ = 0 };
/*
Burada tanımlanan tamsayı sabitler aynı konuya (burada cinsiyet) ilişkin bir bütünün parçaları
olan (integral) sabitlerdir.
*/
enum renkKodlari {
    SIYAH=0x000000, // Hiç belirtilmez ise 0
    KIRMIZI=0xFF0000, // Hiç belirtilmez ise 1
    YESIL=0x00FF00, // Hiç belirtilmez ise 2
    MAVI=0x0000FF, // Hiç belirtilmez ise 3
    BEYAZ=0xFFFFFF // Hiç belirtilmez ise 4
};
int main() {
    // Sabitleri konsola yazıyoruz
    printf("ERKEK = %d\n", ERKEK);
    printf("KADIN = %d\n", KADIN);
    printf("BELIRSIZ = %d\n", BELIRSIZ);
    int cinsiyet;
    puts("Cinsiyet Giriniz (0-1-2):");
    scanf("%d",&cinsiyet);
    switch (cinsiyet) {
        case ERKEK: puts("ERKEK Girdiniz"); break;
        case KADIN: puts("KADIN Girdiniz"); break;
        case BELIRSIZ: puts("Cinsiyet Bilinmiyor"); break;
    }
    return 0;
}
```

Takma İsimler

C Dilinde halihazırda mevcut veri tiplerinin adını yeniden tanımlamak ya da *takma isim* (*typedef*) verilebilir;

```
typedef mevcutisim takmaisim;
```

Aşağıda takma isim verilmiş yeni veri tipleriyle yazılmış bir kod örneği verilmiştir;

```
#include <stdio.h>

typedef char karakter;
typedef int tamsayi;
typedef long long buyukolceklitamsayi;
typedef unsigned int pozitifTamsayi;

int main() {
```

```
karakter k='A';
tamsayi t=12;
buyukolceklitamsayi bot=0xFFFFFFFF;
pozitifitamsayi pt=1;

printf("k:%d veya %c\n",k,k);
printf("t:%d\n", t);
printf("bot:%ld\n",bot);
printf("pt:%u\n",pt);

return 0;
}
```

Tip tanımlama, daha sonra göreceğimiz **yapi** (**struct**) ve **birlik** (**union**) veri tiplerinin çokça kullanıldığı kodlarda kodun boyunu oldukça kısaltır. Bunların dışında fonksiyonlara da takma isim verilebilir;

```
typedef int Fonk(int); /* Fonk bir fonksiyona verilen bir takma isimdir;
Fonk, int tipinde parametre alan ve int geri döndüren her fonksiyon olabilir */

int faktoriyel(int n) { //faktoriyel fonksiyonu da Fonk tipindedir.
    return (n==1)? 1: n*faktoriyel(n-1);
}
int onKat(int i) {
    return i*10;
}
int main() {
    Fonk *fn; // fn bir Fonk tipinde olan fonksiyonlara olan göstericidir.
    fn = &faktoriyel; // faktoriyel fonksiyonunu göster
    fn(3); // 3! Hesaplanır
    fn = &onKat; // onKat fonksiyonunu göster
    fn(4); // 4*10 hesaplanır
}
```