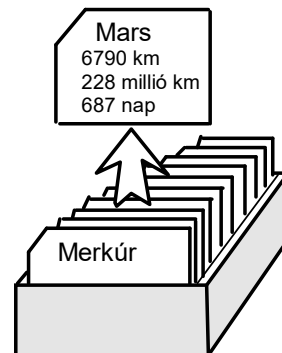
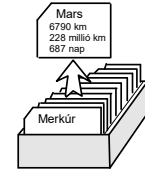


□ A struct, az union és az enum típus

- *A struktúra adattípus*
- *Struktúra típus definiálása*
- *Struktúra típusú változó definiálása*
- *Hivatkozás a struktúratagokra*
- *Mintapélda struktúrák alkalmazására*
- *Az enum típus*
- *Mintaprogram enum típus alkalmazására*



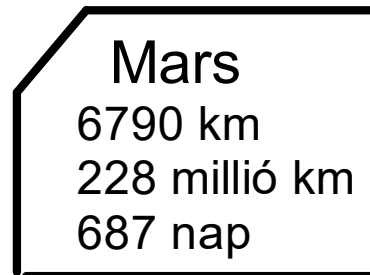
□ A struktúra adattípus



A struktúra adattípus egy **egyed** logikailag összetartozó, viszonylag kisszámú, különböző adattípusokkal ábrázolható jellemzőinek megadására szolgál.

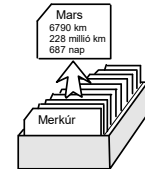
A jellemzők tárolására önállóan is használható **struktúratagok** szolgálnak.

A struktúra összes adattagjával egyetlen **egységként is kezelhető**, pl. értékátadásban, függvények paraméterátadásában vagy visszatérő értékeként.



Példaként a Nap
bolygói:

Data members of struct



Name of planet	Diameter [km]	Distance from the Sun[km]	Period	year/ day
Mercury	4 880	58 millió	88	day
Venus	12 100	108 millió	225	day
Earth	12 756	150 millió	365	day
Mars	6 790	228 millió	687	day
Jupiter	142 800	778 millió	12	year
Saturn	120 860	1427 millió	29,5	year
Uranus	52 000	2870 millió	84	year
Neptune	48 400	4497 millió	165	year
Pluto	3 000	5950 millió	248	year

One structure

□ Struktúra típus definiálása

Több lehetőség:

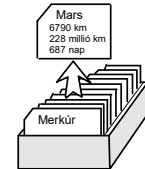
```
struct <struktúra_típusnév>
{ <típus1> struktúratag1 ;
  <típus2> struktúratag2 ;
  . . .
  <típusK> struktúratagK ;
};
```

vagy **typedef** utasítással kétféle alakban:

```
typedef struct <struktúra_típusnév>
{ <típus1> struktúratag1 ;
  <típus2> struktúratag2 ;
  . . .
  <típusK> struktúratagK ;
};
```

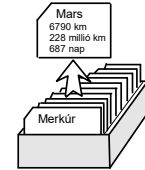
vagy

```
typedef struct{ <típus1> struktúratag1 ;
                <típus2> struktúratag2 ;
                . . .
                <típusK> struktúratagK ;
} <STRUKTÚRA_TÍPUSNÉV>;
```



Példák struct típus definiálására:

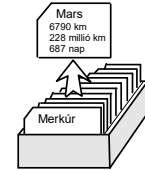
```
struct ponttípus { float x ; float y ; } ;  
typedef struct komplextípus { double Re ; double Im ; } ;  
typedef struct { char nev[ 40 ] ;  
                char tankor [ 6 ] ;  
                } HALLGATOREKTÍP ;
```



□ **Struktúra változó definiálása**

a típus-definiálási formától függően eltérő, a struct szó után megadott típusnév csak a struct szóval együtt jelenti az adott struktúratípust, míg a harmadik alak típusneve önállóan szerepelhet struktúraváltozók definiálásában:

```
struct <struktúra_típusnév> <változónév> ;  
  
<STRUKTÚRA_TÍPUSNÉV> <változónév> ;
```



Példák struktúraváltozók definiálására:

```
struct ponttipus P1, P2 ;  
struct komplextip cmp , * cmpmut ;  
HALLGATOREKTIP hallgatovekt [120] ;
```

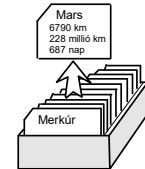
Az utolsó példa egy struktúrából álló vektort definiál.

□ **Hivatkozás a struktúratagokra**

A hivatkozás operátora a . (pont). Példák:

```
P1.x = 6.234; P1.y = 12*34.5 ;  
cmp.Re = cmp.Re + 4.67 ; cmp.Im = 12.45 ;  
strcpy( hallgatovekt [12] . nev , "Nagy Tamás" );  
strcpy( hallgatovekt [12] . tankor , "G1BV3" );
```

Megj.: Az strcpy() függvénnyel másolhatjuk át a szövegeket a struktúratagokba, mivel karaktervektorok számára nincs értelmezve az egyben történő értékadás.



Mutatóval elérhető struktúrák tagjainak hivatkozása
hagyományos módon, vagy a \rightarrow nyíl operátorral
történhet:

```
(*cmpmut) . Re = 34.7;  
cmpmut ->Re = 34.7;
```

Azonos típusú struktúrák között értelmezve van az értékadás,
pl.:

```
P1 = P2 ;
```

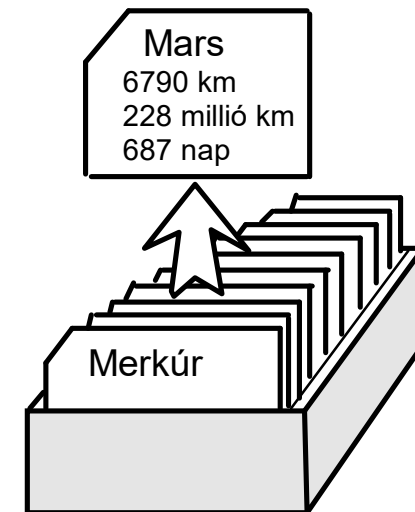
A struktúrák is elláthatók definiáláskor kezdőértékkel, pl.:

```
HALLGATOREKTIP diak = {"Gál Éva", "G1BG1"};  
struct komplextip cmp = { -123.44 , 254.3 } ;
```

□ Mintapélda struktúrák alkalmazására:

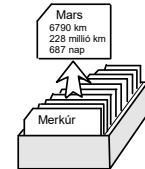
Hozzuk létre a Nap bolygóinak adatait tároló struktúra-vektort,
majd irassuk ki a Mars adatait!

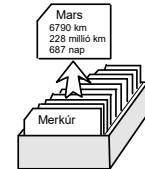
```
#include <stdio.h>
#include <conio.h>
#define bsz 9           // bolygók száma
typedef struct {
    char nev [15];
    unsigned int atmero;
    long int tavolsag;
    float kering_ido;
    char evvagynap[ 4 ];
} BOLYGOTIP;
BOLYGOTIP bolygok[ bsz ];
```





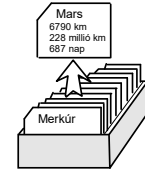
```
main()
{
    int i;
    char nev [15];
    printf("Bolygók adatbázisa\n\n");
    for (i = 0; i < bsz ; i++)
    {
        printf( " \nA bolygó neve: " );
        scanf( "%s", bolygok[ i ].nev );
        printf( "Átmérője=" );
        scanf( "%u", &bolygok[ i ].atmero );
        printf( "Naptól való távolsága=" );
        scanf( "%ld", &bolygok[ i ].tavolsag );
        printf("Keringési idő=");
        scanf("%f",&bolygok[ i ].kering_ido);
        printf("Években, vagy napokban? (év/nap):");
        scanf("%s",bolygok[ i ].evvagynap);
    }
}
```





```
printf( "\n\nA keresett bolygó neve:" );
scanf( "%s", nev );
for ( i = 0; i < bsz ; i++)
{ int j = 0;
  while (nev[ j ] == bolygok[ i ].nev[ j ] && nev[ j ] && bolygok[ i ].nev[ j ] )
    j++; /* a while ciklus magja */
  if (nev[ j ] == bolygok[ i ].nev[ j ]) /* a két név azonos */
  {
    printf( "\nA bolygó átmérője= %u", bolygok[ i ].atmero );
    printf( "\nNaptól való távolsága= %ld millió km", bolygok[ i ].tavolsag );
    printf( "\nKeringési ideje= %f %s",
            bolygok[ i ].kering_ido, bolygok[ i ].evvagynap);
    break; /* kiugrik a for ciklusból */
  }
}
getch();
}
```

□ Az enum típus

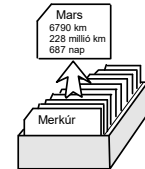


Az enum típus felsorolt azonosítók automatikus sorszámozására, konstans sorszámérték hozzárendelésére alkalmas.

*A 0-val kezdődő automatikus értékhozzárendelést megváltoztathatjuk, ha valamelyik azonosító számára előírjuk az értéket, mely **int** típusú lehet. A nem megadott értékű azonosítók az előttük levőtől eggyel nagyobbak.*

*Az **enum** típus konstansai egész értékeként viselkednek. Amennyiben önálló típusnévvel látjuk el, a felsorolt értékek egy ilyen típusú adat értékészletként foghatók fel.*

hétfő kedd szerda csütörtök péntek szombat vasárnap



Példák enum típusú definíciókra:

```
enum { false, true } ; /* false = 0, true = 1 értékű */  
enum fibonacci { a, b, c = 1, d = 2, e = 3, f = 5 };
```

Definiálás önálló típusnévvel:

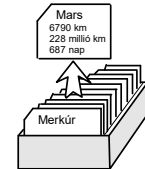
```
typedef enum evszaktip { tavasz, nyar, osz, tel };  
enum evszaktip evszak; /* evszak változó definiálása */
```

□ *Mintaprogram enum típus alkalmazására*

Egy két TV-szaküzlettel rendelkező vállalatnál ismertek boltonként egy-egy mátrix alakjában a naponta forgalmazott készülék-darabszámok típusonként, valamint az egyes típusok egységárai. Határozzuk meg az egyes munkanapok két boltra és típusokra nézve összesített forgalmát forintban!

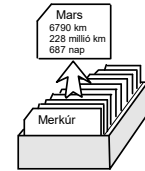
```
/* Forgalomszámítás */
#include <stdio.h>
#include <conio.h>
void main()
{
  const char Nap[5][10]=
    {"Hétfo", "Kedd", "Szerda", "Csütörtök", "Péntek"},
    Gyarto[4][10]={"Sony", "Panasonic", "Grundig", "Philips"};
  typedef enum naptip {Hetfo, Kedd, Szerda, Csutortok, Pentek};
  typedef enum gyartotip {Sony, Panasonic, Grundig, Philips};
  unsigned int darab1[Pentek+1][Philips+1],
    darab2[Pentek+1][Philips+1],
    osszes[Pentek+1][Philips+1];

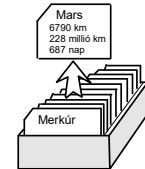
  enum naptip nap;
  enum gyartotip gyarto;
  float egysegarak[Philips+1];
  float bevetelek[Pentek+1];
```





```
printf("Forgalomszámítás\n");  
/* Feltöltés billentyűzetről: */  
for (nap = Hetfo; nap <= Pentek ; nap++)  
    for (gyarto = Sony; gyarto <= Philips; gyarto++)  
    {  
        printf("\nAdja meg a %si napon eladott\n", Nap[nap]);  
        printf("%s típusú készülékek darabszámát:\n", Gyarto[gyarto]);  
        printf(" Az első üzletnél = "); scanf("%u",&darab1[nap][gyarto]);  
        printf(" A második üzletnél= "); scanf("%u",&darab2[nap][gyarto]);  
    }  
/* Az egységárak beolvasása: */  
printf("\n");  
for (gyarto = Sony; gyarto <= Philips; gyarto++)  
    {  
        printf("Adja meg a %s típusú TV árát (Ft/db)= ",Gyarto[gyarto]);  
        scanf("%f",&egysegarak[gyarto]);  
    }  
}
```





```
/* A két bolt forgalmának (mátrixoknak) összegzése: */  
for (nap = Hetfo; nap <= Pentek; nap++)  
{  
    for (gyarto = Sony; gyarto <= Philips; gyarto++)  
    {  
        osszes[nap][gyarto] = darab1[nap][gyarto] + darab2[nap][gyarto];  
    }  
}  
  
/* Napi bevétel számítása (mátrix szorzása vektorral): */  
for (nap = Hetfo; nap <= Pentek; nap++)  
    for (bevetelek[nap] = 0, gyarto = Sony; gyarto <= Philips; gyarto++)  
        bevetelek[nap] += osszes[nap][gyarto] * egysegarak[gyarto];  
  
/* Napi összesített bevételek kiíratása: */  
for (nap = Hetfo; nap <= Pentek; nap++)  
    printf("\nA %si összesített bevétel:    \t %13.2f Ft",  
        Nap[nap],bevetelek[nap]);  
  
getch();  
}
```