



# Programozás I.

## 5. Struktúrák

# Struktúrák

# Gyakorló feladatok

## ◉ 3.8 Feladat

- ◉ Készítsen programot, amely egy számítógép adatait képes tárolni (órajel, memória mérete, háttértár mérete)
- ◉ Billentyűzetről töltse fel az adatait
- ◉ Írassa ki a számítógép adatait

## ◉ 3.9 Feladat

- ◉ Módosítsa az előző feladatot úgy, hogy a program három számítógép adatait tárolja

# Probléma az adattárolással

- Mi van, ha a példákban nem 3, hanem 10, 20, 100 gép adatait kell tárolni?
- Egy megoldás: minden adathoz egy  $N$  elemű tömb.
  - Ez érezhetően nem szép.
- Ha több változó valami miatt összefügg, azokat jó lenne úgy is kezelni.

# Struktúra ötlete

- Az egymáshoz köthető változókat kezeljük egybe.
  - Pl.: A számítógéphez tárolunk órajelet, memóriát, háttértárat. (3 külön változó)
  - Mi lenne, ha lenne inkább egy számítógép változónk?

# Struktúra

- Struktúra létrehozása a **struct** kulcsszó segítségével történik, ami után egy nevet is meg kell adni.
- A struktúra megadása mindig az "int main()" sor előtt történjen!

```
struct computer  
{  
    float cpu;  
    int memory, hdd;  
};
```

 A ';' kell a végére, az zárja le a definíciót.

# Struktúra változó

- Ha létrehoztunk egy struktúrát, akkor egy új típust hoztunk létre. A típus megnevezéséhez a **struct** kulcsszó is hozzátartozik.
- Változó létrehozása:
  - `struct computer comp1, comp2;`
- Változó tömb létrehozása:
  - `struct computer array[100];`

# Struktúra elemeinek elérése

- A struktúra elemeit a '.' segítségével tudjuk elérni. Először meg kell adni a struktúra változó nevét, majd a '.' után a struktúrán belüli változó nevét.

```
struct computer c1;
```

```
c1.cpu=2.5;
```

```
c1.memory=8;
```

```
c1.hdd=750;
```



# Struktúra elemeinek elérése

- Az ily módon hivatkozott adattagokat változóként kell kezelni a megszokott módon:
  - `printf(„%d\\n”, c1.memory);`
  - `scanf(„%d”, &c1.memory);`

# Gyakorló feladatok

## ◉ 5.1 Feladat

- ◉ Készítsen programot, amely egy számítógép adatait képes tárolni (órajel, memória mérete, háttértár mérete)
- ◉ Billentyűzetről töltse fel a struktúra adatait
- ◉ Írassa ki a számítógép adatait

## ◉ 5.2 Feladat

- ◉ Módosítsa az előző feladatot úgy, hogy a program három számítógép adatait tárolja

# Típusdefiníció

- A **typedef** kulcsszó segítségével lehetőségünk van átnevezni létező típusokat.
  - `typedef int Integer;`
  - `typedef float Floating;`
- Innentől az átnevezett verziót is lehet használni
  - `Integer i, j;`
  - `Floating f;`

# Típusdefiníció

- A típus átnevezését struktúráknál is lehet használni a struktúra definíciójától függetlenül:
  - `typedef struct computer Computer;`
- Vagy egybekötve:

```
typedef struct computer
{
    float cpu;
    int memory, hdd;
} Computer;
```

# Tömb struktúrákból

- A struktúrával létrehozott típusból ugyanúgy lehet tömböt létrehozni, mint bármely más típusból:
  - `Computer comp_array[10];`
- Ekkor a tömb egy eleme egy struktúra:
  - `comp_array[3];`
- A struktúra adattagjait is el lehet érni:
  - `comp_array[3].hdd=1000;`

# Tömb a struktúrában

- Egy struktúrában bármilyen adattagok lehetnek, akár tömbök is:

```
struct temperature  
{  
    int x, y;  
    int measures[40];  
};
```

- Ilyenkor a struktúrában lévő tömb egyes elemeit is el lehet érni:
  - `struct temperature t;`
  - `t.measures[4]=42;`

# Struktúra a struktúrában

- Egy struktúra akár más struktúrákat is tartalmazhat:

```
struct workspace  
{  
    Computer pc;  
    int user_id;  
};
```

# Struktúra a struktúrában

- Ilyenkor az elemek eléréséhez láncolt hivatkozás kell:

```
struct workspace ws;
```

```
ws.user_id=16;
```

```
ws.pc.hdd=750;
```



# Gyakorló feladatok

## • 5.5 Feladat

- Készítsen programot, amely térbeli pontokat képes tárolni (x, y és z komponens)
- Olvassa be billentyűzetről tíz pont koordinátáját
- Írassa ki a beolvasott adatokat
- Írassa ki az origótól legtávolabb elhelyezkedő pontot

# Gyakorló feladatok

## • 5.7 Feladat

- Készítsen programot, amely háromszögek tárolására alkalmas. Egy háromszög három térbeli pontot tartalmaz ( $x$ ,  $y$  és  $z$  komponens)
- Olvassa be billentyűzetről három háromszöget
- Írassa ki a beolvasott adatokat
- Keresse meg és írassa ki a legkisebb kerületű háromszög csúcsait

# Gyakorló feladatok

## ◉ 5.9 Feladat

- ◉ Készítsen programot, amely háromszögeket tárol a következő formában:
  - ◉ Tárolunk 10 koordinátát (struktúrák)
  - ◉ Tárolunk 5 háromszöget (struktúrák)
  - ◉ A háromszöget három index segítségével határozzuk meg (a 10 elemű koordináta tömb indexei)
- ◉ Olvassa be billentyűzetről tíz koordinátát
- ◉ Olvassa be billentyűzetről a háromszögek indexeit. Minden háromszög beolvasása előtt jelenítse meg a tíz koordinátát azok indexeivel együtt
- ◉ Jelenítse meg a beolvasott adatokat

# Gondolkodtató feladatok

## ◉ 5.10 Feladat(!)

- ◉ Módosítsa az előző programot úgy, hogy a beolvasás után a program ellenőrizze, hogy vannak-e olyan koordináták a tömbben, amelyek egyetlen háromszögben sem szerepelnek. Ezeket törölje a tömbből, majd a használt értékeket rendezze a tömb elejére. Módosítsa a háromszögben tárolt indexeket is úgy, hogy ugyanazokra a pontokra hivatkozzanak.