

Dr. Holló Csaba, SZTE TTIK, Szoftverfejlesztés Tanszék

Programozási alapismeretek kurzus

Olvasási idő:
7 perc



Alapvető programozásnyelvi elemek C-ben és PHP-ban

VII. Struktúrák és objektumok

A lecke célja A struktúrák és objektumok hasonlóságának feltárása az objektumok technikai kezelésének jobb megértése, illetve a struktúrák fontosságának tudatosítása érdekében.

Tudás Ismeri a struktúrák és az objektumok közötti analógiákat.

Képesség Helyesen használja a struktúrákat és az objektumreferenciákat.

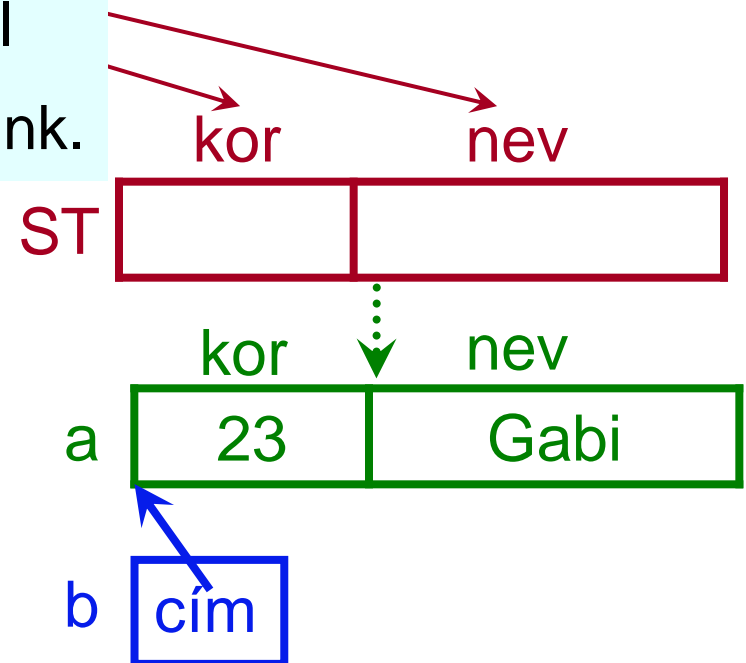
Struktúra típus C-ben

- Különböző típusú összetartozó adatok együttes tárolására, egyetlen egységként történő kezelésére szolgáló összetett típus.
- A struktúrában az értékeket tároló adattagokat egyedi azonosítókkal (nevekkel) látjuk el és hivatkozunk.
- Deklarálhatunk struktúra típusú változókat, illetve mutatókat is.

- További információkat a struktúrákról a [Sulinet Tudásbázisban](#) olvashatunk.

```
typedef struct ST {
    int kor;
    char nev[20]; } ST;

int main(){ ST a, *b; b = &a;... }
```

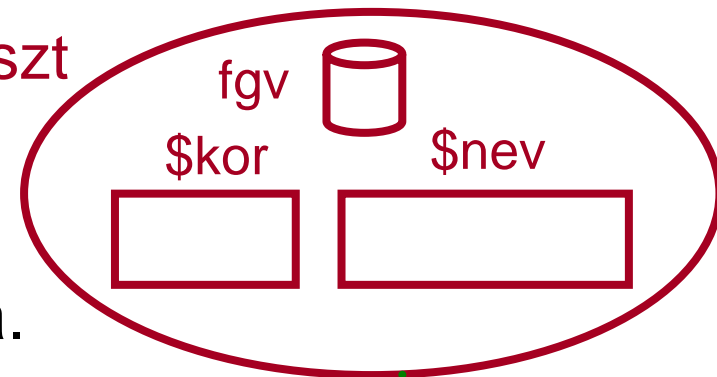


Osztály

- Metódusok (függvények) és
- adatok (mezők és jellemzők)
- egységbe zárt csoportjának típusa.

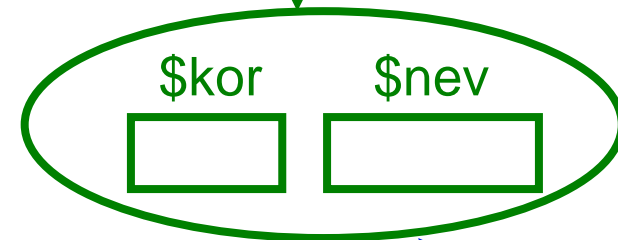
- Olyan, mint egy struktúra típus, csak metódusok (függvények) is tartoznak hozzá.
- Továbbá az adatoknak korlátozható az elérése (például az osztályhoz tartozó függvényekre).

Oszt



Objektum

- Technikailag egy osztály típusú változó, amit
- többnyire dinamikus változóként hozunk létre,
- tehát olyan, mint egy struktúra típusú dinamikus változó.



\$o1

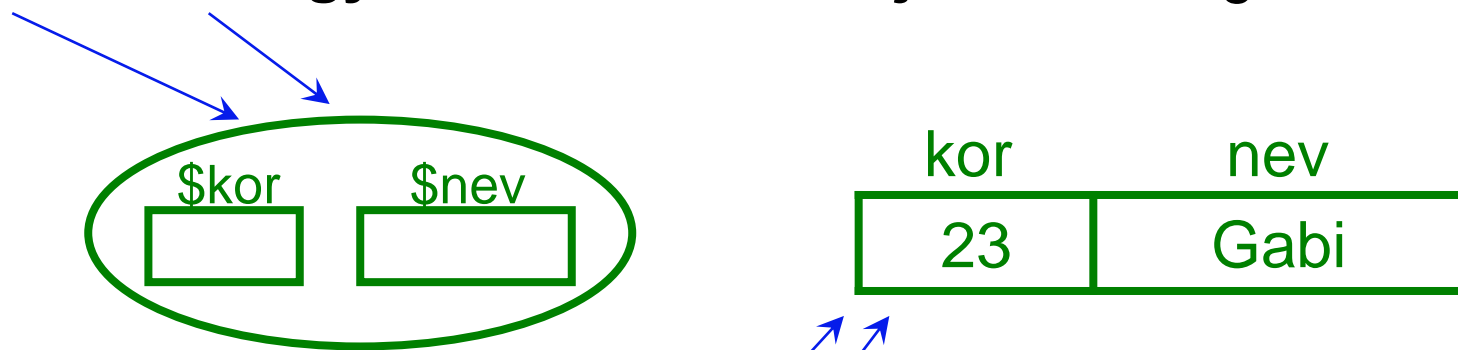
Objektumok hivatkozása PHP-ban

- Többnyire ún. **objektumkezelővel** (angolul: *handle*) hivatkozunk, ami C-ben struktúrára mutató típusatlan mutatónak felel meg, és természetesen illeszkedik a változó dinamikus jellegéhez is.
- Az, hogy az **objektumhivatkozás valójában mutató**, a változó objektumhoz való hozzárendelésekor szintaktikailag nem (feltétlen) látszik, ugyanakkor a helyes használat érdekében fontos ennek tudatában lenni.
- ❖ A PHP osztályokról, objektumokról, és objektumkezelőkről bővebben olvashatunk az ELTE [PHP programozás](#) oldalán.

Mutatók és objektumhivatkozások

Például, ha van két $\$o1$, $\$o2$ objektumhivatkozásunk, akkor:

- az $\$o1 = \$o2$ értékadás azt eredményezi, hogy
- $\$o1$ és $\$o2$ **ugyanarra a közös objektumra** fognak mutatni.



Ennek tehát C-ben két, $o1$ és $o2$ mutató felel meg, melyek ebben az esetben **ST** struktúrák címeit tárolnák. C-ben célszerűbb struktúra típusú mutatókat használni, de ez most a lényegen nem változtat. Az $o1 = o2$ értékadás után a két mutató ugyanarra a struktúrára fog mutatni, és ugyanazt módosítják.

Mutatók és objektumhivatkozások

```
ST *o1, *o2, *o3; o1 = (ST*) malloc(sizeof(ST));
```

```
o1->kor = 20; strcpy(o1->nev, "Csilla");
```

```
o2 = o1; o2->kor = 25;
```

```
strcpy(o2->nev, "Gabi");
```

o1 **is**
módosult

```
o1->kor=25, o1->nev=Gabi  
o2->kor=25, o2->nev=Gabi
```

Ahogy C-ben az **o1** és **o2** használatával történő értékadások ugyanazt a struktúrát módosítják,

PHP-ban az **\$o1=\$o2** értékadás után az **\$o1**, illetve **\$o2** hivatkozásokkal történő értékadások ugyanazt a közös objektumot módosítják.

❖ További információk mutatókról, objektumkezelőkről és referenciákról.

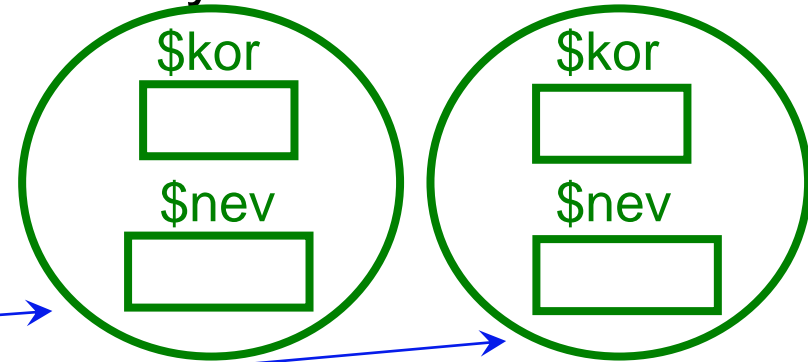
Klónozás PHP-ban

PHP-ban az objektumok között értelmezett egy másfajta értékadás is, amit **klónozásnak** neveznek és amelynek a szintaxisa:

```
$o3 = clone $o1;
```

Ennek során létrejön egy új objektum

- melyre `$o3` fog mutatni és
- az `$o1` által mutatott objektum tartalma átmásolódik az új, `$o3` által mutatott objektumba



Továbbá meg lehet adni egy függvényt, mely minden klónozáskor lefut (például módosítja az új objektum jellemzőit).

- A PHP objektumok klónozásáról további tudnivalókat olvashatunk a [Sapientia](#) egyetem, illetve a [php.net](#) oldalán.

Klónozás

C-ben a klónozásnak az felel meg, mintha

- létrehoznánk egy új struktúra típusú változót,
- abba átmásolnánk az eddigi tartalmát,
- de ez továbbra is két különböző változó lenne.

o1 **nem**
módosult

```
o3 = (ST*) malloc(sizeof(ST));
```

```
*o3 = *o1; o3->kor = 40;
```

```
strcpy(o2->nev, "Laci");
```

```
o1->kor=25, o1->nev=Gabi  
o3->kor=40, o3->nev=Laci
```

Osztályok és objektumok

Megemlítendő, hogy az objektumorientáltságnak sok olyan további eleme van, amelyekről ebben a fejezetben nem esett szó, mivel itt kizárólag a struktúrák és objektumok összehasonlítására koncentráltunk.

Kérdések, feladatok

1. $\$o3 = \text{clone } \$o1$; esetén az $\$o3$ által mutatott objektum módosítása következtében módosulni fog-e az $\$o1$ által mutatott objektum?
2. $\$o3 = \text{clone } \$o1$; esetén az $\$o1$ által mutatott objektum módosítása következtében módosulni fog-e az $\$o3$ által mutatott objektum?

Megoldások. Mindkét esetben: NEM, mivel a `clone` következtében $\$o1$ és $\$o3$ két különböző objektum lesz.

3. Az $\$o1 = \$o2$; $\$o3 = \$o2$; utasítások után, az $\$o1$ által mutatott objektum módosítása következtében módosulni fog-e az $\$o3$ által mutatott objektum?

Megoldás. Igen, mindkettő az $\$o2$; által mutatott objektumra mutat.

EFOP-3.4.3-16-2016-00014

AP1 HALLGATÓI DIPLOMA-SZERZÉST SEGÍTŐ SZOLGÁLTATÁSOK

Jelen tananyag a
Szegedi Tudományegyetemen
készült az
Európai Unió támogatásával.
Projekt azonosító:
EFOP-3.4.3-16-2016-00014



SZÉCHENYI 2020

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE