

Objektumorientált tervezés

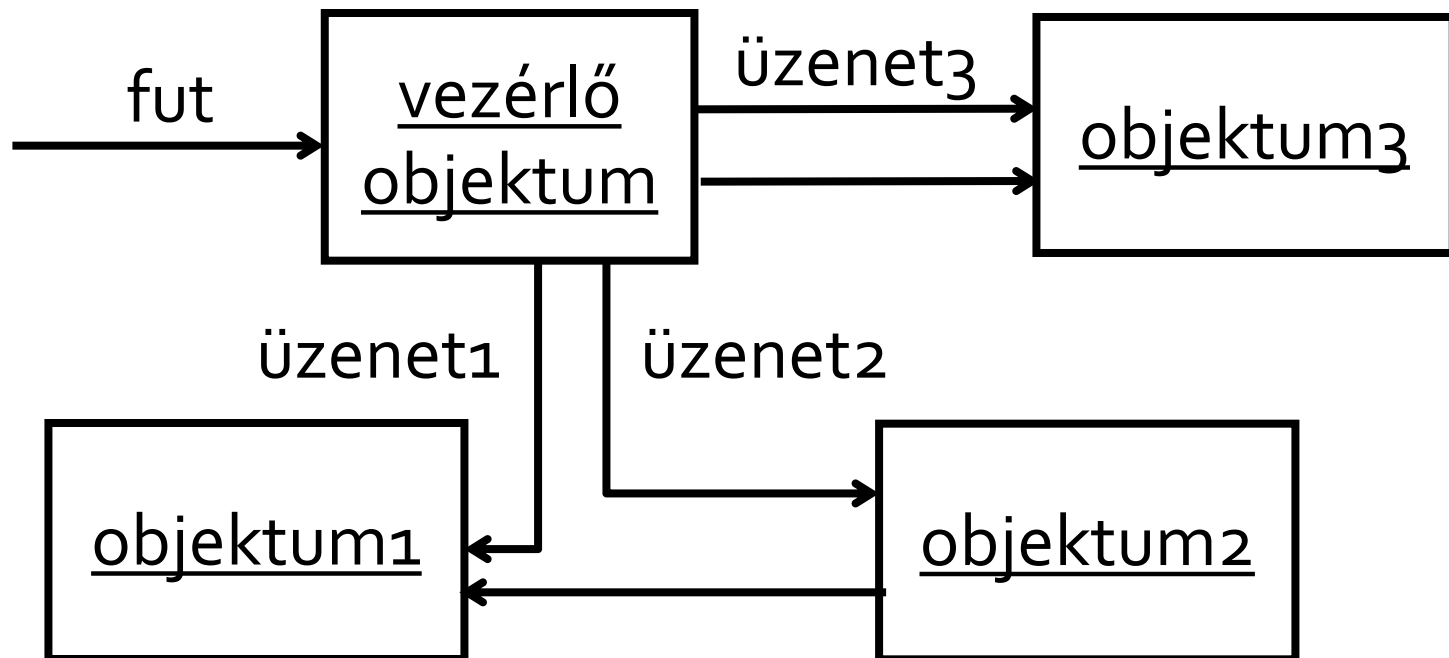
- Kialakulása a 60-as évek végén tehető
- Válasz a szoftverkrízisre
- A szoftverfejlesztés során az együtt változó részek elkülöníthetők
- Projektek közötti újrahasznosíthatóság növelése

Objektumorientált tervezés

- Bezárás - az információ elrejtése: az adatokat csak interfészeken (metódusok) keresztül lehet elérni.
- Öröklődés - az utód osztály örökli az ős adatait, metódusait, valamint tartalmazhat újakat, és felülírhat régi metódusokat.
- Polimorfizmus – többalakúság: ugyanarra a metódus hívásra, különböző objektumok különbözőképpen reagálhatnak.
- Utánzás, modellezés elvének alkalmazása a programozásban

OOP program

- Egy objektumorientált program egymással kommunikáló objektumok összessége, melyben minden objektumnak megvan a feladatköre



Objektumközpontú tervezés lépései

- *A szükséges osztályok azonosítása:* az adott alkalmazási terület milyen osztályok, objektumok bevezetését teszi szükségessé. A jó tervezés kulcsa a valóság valamely részének közvetlen modellezése, vagyis a program fogalmainak osztályokra való leképezése.
- *Az osztályok feladatainak meghatározása:* meg kell határozni a rendszerben előforduló valamennyi osztály szerepét és feladatát.
- *Az osztályokkal együttműködő osztályok meghatározása*

Objektumközpontú tervezés lépései

- *Felhasználási esettanulmányok alapján finomítani kell az osztályok feladatköreit (use case):* kiválasztunk jellemző alkalmazási példákat, és a terv alapján nyomon követjük az üzenetek objektumok közötti áramlását. A használati esetek a rendszert (dinamikusan) működő egységként tekintik.
- *Át kell tekintenünk az osztályok egymás közötti viszonyát*
- *Hierarchiába szervezés:* ez az öröklődés révén lehetővé teszi a kód újrahasznosíthatóságát; elvont osztályok
- *Újrahasznosítható tervezési keretrendszerek létrehozása:* egymással együttműködő osztályok rendszere valamilyen feladatkör megoldására

Objektumorientált tervezés

- Objektumközpontú tervezés szintjei:
 1. **Architektúrális tervezés** - a létrehozandó programot mint egészet érintő kérdésekben döntünk.
 2. **Külső interfész tervezése** - a külvilággal történő kapcsolattartás módját és részleteit írja le.
 3. **Az objektumtervezés** (más néven **részletes tervezés**) - a programot felépítő elemek – objektum orientált megközelítésben az osztályok és objektumok – egyenkénti specifikálása oly módon, hogy azok akkor is együtt tudjanak működni, ha a különböző elemeket, különböző programozók implementálják.