

2. OOP főbb jellemzői

OOP jelentése: Objektumorientált program

OOP nyelv használata ≠ OOP szemlélet

OOP: Egymással kommunikáló objektumok összessége, amelyben minden objektumnak megvan a jól meghatározott feladatköre.

Objektum: Logikailag összetartozó adatok és rajtuk dolgozó rutinok összessége.
Az objektum információt tárol, és kérésre feladatot hajt végre.

Üzenet (kérelem): Egy objektumba beprogramozott rutin (metódus) hívása.
Egy objektum csak akkor küldhet üzenetet egy másiknak, ha azzal kapcsolatban áll.

Felelősség: Minden objektumnak megvan a felelősségi köre. Az objektum felelős a feladatai elvégzéséért.

Bezárás / egységbe zárás: A feladatok elvégzésének „hogyanja” az objektum belügye. Az objektummal csak interfész-en keresztül lehet kommunikálni.

Láthatóság: A különböző objektumok különböző módon láthatják / nem láthatják egymás adatait, illetve metódusait.

- *Public*: mindenki láthatja (+)
- *Private*: csak osztályon belül látható, a leszármazott osztályok nem láthatják és nem is módosíthatják (-)
- *Protected*: annyiban különbözik a privát-tól, hogy a leszármazott osztályok módosíthatják is (#)

Poliformizmus (többalakúság): Ugyanarra a kérelemre a különböző objektumok különbözőképpen reagálhatnak.

Osztályozás: Az objektumpéldányokat viselkedésük szerint osztályokba soroljuk.

Öröklődés: Egy osztály örökölhet tulajdonságokat és viselkedési formákat egy másik osztálytól. Az utódosztályban csak az ősoosztálytól való eltérést kell megadni.

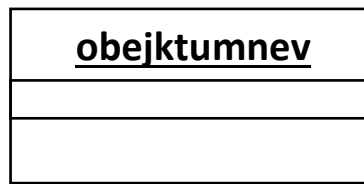
Futás alatti (késői) kötés: Hogy egy üzenet pontosan melyik metódus hívását jelenti, az sokszor csak a program futásakor derül ki. Ez a jelenség a futás alatti kötés.

UML alapelemei:

Osztály



Objektum példány



Függőségi kapcsolat



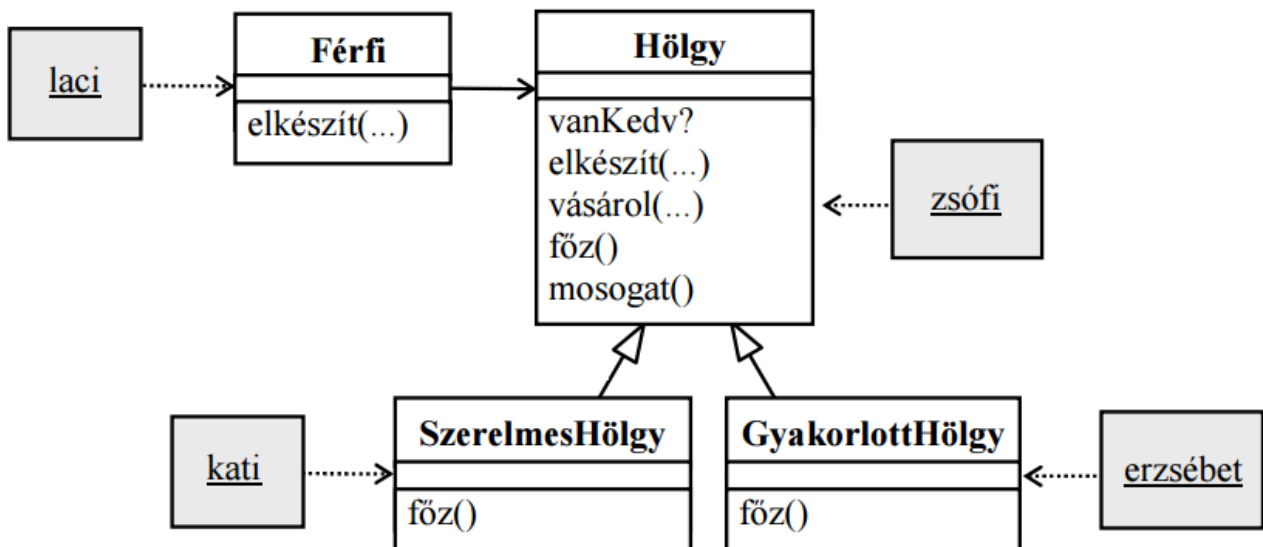
Társítási kapcsolat



Öröklési kapcsolat



Órai példa:



Otthoni feladat:

Bolt osztály, Húsbolt osztály, Zöldségbolt osztály metódusai, példányok, vásárlás menete