# Objektum Orientált programozás – alapismeretek

## A program készítés

Absztrakciós folyamat, amelyben a valós világban létező jelenséget (megoldandó problémát) valamilyen programozási eszköz absztrakciós szintjén képezünk le.

• A munka nehézsége függ

– a leképezendő jelenség/folyamat bonyolultságától (komplexitásától)

– a valóság és a programozási eszköz absztrakciós szintjének különbözőségétől ("távolságától")

## Absztrakció

a leglényegesebb tulajdonságok kiemelése és általánosítása

## Programozási nyelvek

A programozási eszközök absztrakciós szintjei különbözőek:

• assembly: az adott számítógép architektúra instrukciókészletének leképezése

• funkcionális (imperatív) nyelvek: adatszerkezetek és algoritmusok absztrakciós szintje.

– Közel állnak ezek a valóság fogalmaihoz? (nem nagyon... )

• objektum orientált nyelvek: együttműködő objektumok absztrakciós szintje .

– Léteznek a valóságban ezek? (IGEN!!!)

## A programozás, mint modellkészítés

A programkészítés egy másik megközelítése

**Program**:

• a valós világ egy szeletének működő modellje  
Ebben a szemléletben a valós világ egy absztrakt modelljét (analízis modell) kell leképezni egy programozási eszközre.

Ezt a folyamatot megkönnyíti az, ha az analízis modell elemei könnyen leképezhetők nyelvi elemekre.

## Az objektum orientált programozás (OOP) szemlélete

Az analízis modell elkészítése során a valóságot objektumok halmazaként tekintjük. Ezen objektumok egymással kapcsolatban vannak és együttműködnek.

## A természetes objektumok jellemzői

• egyéniség (különállás)  
• struktúra (adatok)   
• viselkedés (más objektumokkal való kapcsolata)  
• állapot (adatainak aktuális értéke)

## Természetes objektumok: példa

Szalacsi Sándor

• Elég bonyolult struktúrájú, számos viselkedés formára képes objektum.  
• Véralkohol szintje, mint adat a struktúrájához tartozik  
• Az alkohol pillanatnyi értéke az állapothoz tartozik  
• Egyes viselkedésformák (pl. borozik) módosíthatják az állapotot  
• Más viselkedésformák működését (pl. járás) az aktuális állapot befolyásolhatja

## A természetes objektumok jellemzői

Az objektum: egyediséggel rendelkező diszkrét entitás

jellemzői: attribútumok, műveletek

attribútum együttes: objektum állapot, ennek időbeli változása az objektum viselkedése

műveletek(operations): ezek modellezik az objektum viselkedését.



## Objektumtól objektumig: 1. szint

• Tanulmányozzuk a valós objektumokat, és megkeressük azoknak a probléma megoldása szempontjából fontos adatai és viselkedésmintáit.

(Az absztrakció definíciójából a "leglényegesebb tulajdonságok kiemelése".)

– Például a Szalacsi Sándor objektumnak a Posta szempontjából lényegtelen az alkoholszintje, de lényeges, hogy melyik szám alatt lakik.

• Ezután megkeressük azon objektumcsoportokat, amelyek azonos adatokkal és viselkedésmintákkal rendelkeznek, és elnevezzük ezt a csoportot. (Az absztrakció definíciójából az "általánosítás".)  
– Így jön létre a Posta szempontjából a "címzett" fogalom.

## 2. szint

• Az előzőleg definiált fogalom leképezése egy programozási eszközre (osztálydefiníció elkészítése). Ez valójában egy minta a program működéséhez szükséges objektumok létrehozására.

– Például a Posta szempontjából egy címzettnek vannak lényeges adatai (neve, irányítószáma, utca neve stb.) és vannak lehetséges viselkedésformái (levelet adathat fel, csomagot fogathat, otthon tartózkodhat vagy ideiglenes lakcímen, de például nem nézheti meg a szomszédja nyugdíjának összegét.)

## 3. szint

• Akkor és annyi objektumot kell létrehozni a program futása során, amelyek együttműködve képesek a valóságot modellezni.  
– Például amikor Szalacsi Sándor Jókai utcai lakos csomagot akar küldeni Gipsz Jakab budapesti lakosnak, akkor

• Létrejön egy feladó típusú objektum Szalacsi Sándor adataival, egy küldemény típusú objektum a megfelelő adatokkal, és egy címzett típusú objektum Gipsz Jakab adataival.

• A Szalacsi objektum üzen a csomagkuldes objektumnak, megadva a lakos nevét és címét, aki (ellenőrzések után) egy csomagot küld a lakos objektumnak, megváltoztatva ezzel az állapotát(címzett).

## Az objektum orientált programozás alapelvei - fogalmai

1. Osztály (class)
2. Objektum (object)
3. Egységbezárás (encapsulation)
4. Információ rejtés (information hiding)
5. Üzenet (message)
6. Öröklődés (inheritance)
7. Polimorfizmus (polymorphism)

**Osztály**, amelynek részei:

• adatok (az attribútumok modellezésére)  
• módszerek (a műveletek modellezésére).  
Objektumok közös tulajdonságait definiálja. Programozás technikai szempontból egy **típus**.

**Objektum**

• Egy osztály egy működőképes példánya.  
• Egy adott osztályban definiált tulajdonságokkal tetszőleges számú objektum példányosítható.  
• Minden objektum természeténél fogva különbözik az összes többitől.  
• Egy adott osztályból példányosított valamennyi objektumnak ugyanolyan lehetséges viselkedés módjai (műveletei) vannak, de saját állapotuk van.

**Egységbezárás**:

Az osztály az adatait és a módszereket egy egységgé teszi.

• az adatok és a módszerek lokálisak  
• a módszerek hozzáférhetnek az adatokhoz

**Információ rejtés**:

• Egy objektum adatai a külvilág (más objektumok) számára hozzáférhetetlenek.  
• Egy objektum a külvilággal csak az interface-én keresztül tarthatja a kapcsolatot. Interface: a külvilág számára elérhető módszerek együttese.  
• A módszerek implementációja rejtett.

**Üzenet**

• Az objektummal való kommunikáció módja.  
• A módszerek aktivizálását (invocation) jelenti

**Öröklődés**

• Hierachikus kapcsolat(rendszer).  
• A leszármazott osztály örökli az ős osztály adatait és módszereit.  
• Az örökölt módszereket felül definiálhatja a maga számára.  
 • Új (saját, csak rá jellemző) adatokat és módszereket definiálhat.  
• Egy leszármazott osztály csak bővítheti, pontosíthatja az őst, de nem utasíthat el örökölt   
adatokat vagy módszereket.

**Polimorfizmus (többalakúság)**

• Bizonyos elemek viselkedése attól a környezettől függ, amelyben alkalmazzuk.  
• A gyakorlatban ez azt jelenti, hogy egy nyelvi elem (például egy kódrészlet) attól függően, hogy hol alkalmazzuk, más-más működést eredményezhet.

## Az objektum orientált program

**Egymással kommunikáló objektumok halmaza**

• működés: üzenetváltások  
• struktúra: osztályok és objektumok közötti kapcsolatok

## Az objektum orientált programkészítés folyamata

• A probléma leírásában keressük az objektumokat  
• Objektum csoportok közös tulajdonságait osztályokkal írjuk le  
• Megkeressük az osztályok kapcsolatait  
• Az osztályokat és kapcsolataikat implementáljuk egy programozási eszközzel

• Létrehozzuk a megfelelő példányokat (objektumokat) a megfelelő időben, és megoldjuk a kommunikációjukat.  
• Az implementációhoz általában szükségünk van az eredeti probléma objektumain túl technikai szerepű osztályokra / objektumokra is, például

– adatszerkezetek kezelésére (tömb, stb)  
– vezérlési feladatok megoldására  
– megjelenítési feladatok megoldására  
– stb.

## Objektum orientált programozási nyelvek

• Nagyon sok van, számos integrált fejlesztő eszközzel.  
• Az első sikeres: Smalltalk  
• Legelterjedtebbek:

– C++, Java, C#, ObjectPascal (főleg a Delphi fejlesztőeszköz miatt), Python, PHP, stb...