

Objektum orientált programozás

Bevezetés

Ficsor Lajos

Miskolci Egyetem

Általános Informatikai Tanszék

Utolsó módosítás: 2008. 03. 04.

A program készítés

- Absztrakciós folyamat, amelyben a valós világban létező jelenséget (megoldandó problémát) valamilyen programozási eszköz absztrakciós szintjén képezzük le.
- A munka nehézsége függ
 - a leképezendő jelenség/folyamat bonyolultságától (komplexitásától)
 - a valóság és a programozási eszköz absztrakciós szintjének különbözőségétől ("távolságától")

Absztrakció

A idegen szavak szótára (Akadémiai Kiadó) szerint:

absztrakció lat 1. elvonatkoztatás, elvonás; **a leglényegesebb tulajdonságok kiemelése és általánosítása** 2. elvont fogalom

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? (Sajna nem...)
- 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

Ficsor Lajos

- Elég bonyolult struktúrájú, számos viselkedés formára képes objektum.
- A vér alkoholszintje, mint adat a struktúrájához tartozik
- Az alkoholszint pillanatnyi értéke az állapothoz tartozik
- Egyes viselkedésformák (pl. sört iszik) módosíthatják az állapotot
- Más viselkedésformák működését (pl. járás) az aktuális állapot befolyásolhatja

A természetes objektumok jellemzői

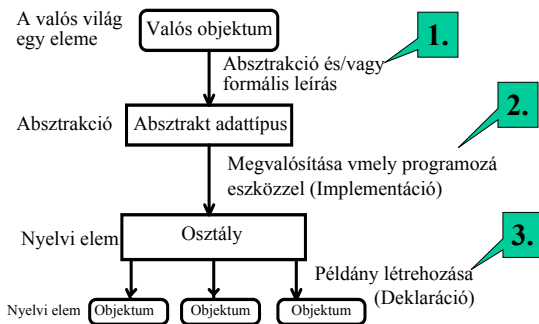
Az objektum: egyediséggel rendelkező diszkrét entitás

jellemzői: attribútumok, műveletek

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

műveletek(operations): ezek modellezzik az objektum viselkedését.

Objektumtól objektumig



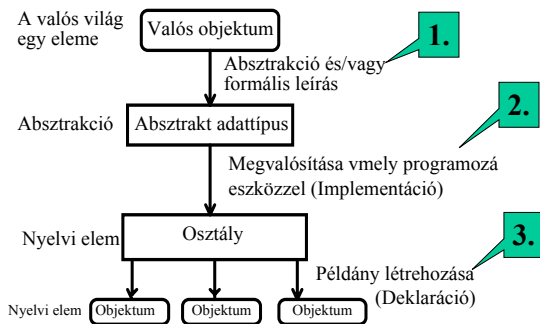
Objektumtól objektumig: 1. szint

- Tanulmányozzuk a valós objektumokat, és megkeressük azoknak a probléma megoldása szempontjából fontos adatai és viselkedésmintáit. (Az absztrakció definíciójából a "leglényegesebb tulajdonságok kiemelése".)
 - Például a Ficsor Lajos objektumnak a Neptun szempontjából lényegtelen az alkoholszintje, de lényeges, hogy melyik tanszéken dolgozik.

Objektumtól objektumig: 1. szint (folyt.)

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

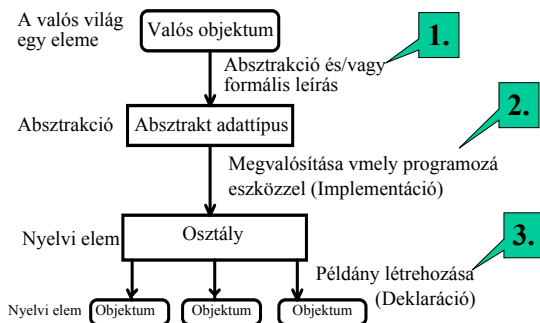
Objektumtól objektumig



Objektumtól objektumig: 2. szint

- Az előzőleg definiált fogalom leképezése egy programozási eszközre (osztálydefiníció elkészítése). Ez valójában egy minta a program működéséhez szükséges objektumok létrehozására.
 - Például a Neptun szempontjából egy oktátónak vannak lényeges adatai (neve, tanszéke, minősítése stb.) és vannak lehetséges viselkedésformái (kurzust hirdethet meg, vizsgaidőpontot írhat ki, jegyet adhat stb., de például nem nézheti meg egy hallgató adószámát.)

Objektumtól objektumig



Objektumtól objektumig: 3. szint

- Akkor és annyi objektumot kell létrehozni a program futása során, amelyek együttműködve képesek a valóságot modellezni.
 - Például amikor Ficsor Lajos oktató be akarja írni Kisházi Brünhilda hallgatónak a GEIAL214N kurzus vizsgájához tartozó jeles vizsgajegyet, akkor
 - Létrejön egy oktató típusú objektum Ficsor Lajos adataival, egy vizsgaalkalom típusú objektum a megfelelő adatokkal, és egy hallgató típusú objektum Kisházi Brünhilda adataival.
 - A Ficsor objektum üzen a vizsgaalkalom objektumnak, megadva a hallgató nevét és jegyét, aki (ellenőrzések után) egy üzenetet küld a hallgató objektumnak, megváltoztatva ezzel az állapotát.

Az objektum orientált programozás alapelvei

- Osztály (class)
- Objektum (object)
- Egységbezárás (encapsulation)
- Információ rejtés (information hiding)
- Üzenet (message)
- Öröklődés (inheritance)
- Polimorfizmus (polymorphism)

Az objektum orientált programozás alapelvei

Osztály, amelynek részei:

- adatok (az attribútumok modellezésére)
- módszerek (a műveletek modellezésére).

Objektumok közös tulajdonságait definiálja.

Programozás technikai szempontból egy **típus**.

alapelvek (folyt.)

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.

alapelvek (folyt.)

Egységbezárás:

Az osztály az adatait és a módszereket egy egységgé teszi.

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

alapelvek (folyt.)

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.

alapelvek (folyt.)

Üzenet

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

alapelvek (folyt.)

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.

alapelvek (folyt.)

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

Az objektum orientált programkészítés folyamata

- 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...
