



Objektumorientált paradigma és programfejlesztés

Bevezető

Vámossy Zoltán

vamossy.zoltan@nik.uni-obuda.hu

Óbudai Egyetem

Neumann János Informatikai Kar





Objektumorientált paradigma

MODELLEZÉS ÉS PROGRAMKÉSZÍTÉS





Bevezető fogalmak

Program, programozás:

1. műveletek sorozata egy bizonyos cél elérése érdekében;
2. utasítássorozat, amely megmondja a számítógépnek, hogyan kezeljen egy problémát;
3. **programtervezés, írás és tesztelés** (belövés)

[Bakos Ferenc szerk.: Idegen szavak és kifejezések szótára, Akadémiai Kiadó, második, átdolgozott kiadás, 2002]





Bevezető fogalmak

- *Paradigma: gör-lat* **1.** bizonyításra v. összehasonlításra adott példa **2.** *nyelvt* ragozási minta **3.** *tud* Az adott korszakban elfogadott tudományos tételek együttese; a kor tudományos világképe.

[Bakos Ferenc szerk.: Idegen szavak és kifejezések szótára, Akadémiai Kiadó, második, átdolgozott kiadás, 2002]





Bevezető fogalmak

Programozási paradigma:

- Az a mód, ahogyan egy-egy programozási nyelv létrehozására felhasználják a programozási alapfogalmakat,
- valamint az a stílus, amelyet az így létrehozott nyelvek sugallnak.

[Bakos Ferenc szerk.: Idegen szavak és kifejezések szótára, Akadémiai Kiadó, második, átdolgozott kiadás, 2002]





Bevezető fogalmak

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

[Idegen szavak és kifejezések szótára, Akadémiai Kiadó]





Előadásrész tematika

Cél: az objektumorientált programozási paradigma segítségével történő modellezés és programkészítés megismertetése

Tartalom:

- Programozás mint modellkészítés
- Modellezés objektumokkal
- Objektumorientált program





Objektumorientált paradigma

PROGRAMOZÁS MINT MODELLKÉSZÍTÉS





Programkészítés

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





Programozási nyelvek

A programozási eszközök absztrakciós szintjei különbözőek:

- Assembly: adott számítógép architektúra utasításkészletének leképezése
- Imperatív programnyelvek: adatszerkezetek és algoritmusok absztrakciós szintje
 - a valós világ fogalmaihoz nem közeli
- Objektumorientált nyelvek: együttműködő objektumok absztrakciós szintje
 - *A valóságban is léteznek*





Programozás mint modellkészítés

A programkészítés egy másik megközelítése

- **Program:** a valós világ egy szeletének működő modellje
- Ebben a szemléletben a valós világ absztrakt modelljét (analízis modell) kell leképezni a programozási eszközre
- Ezt a folyamatot megkönnyíti az, ha az analízis modell elemei könnyen leképezhetők a nyelvi elemekre





Az OOP szemlélete

- Az analízis modell elkészítése során a valóságot objektumok halmazaként tekintjük. Ezen objektumok egymással kapcsolatban vannak és együttműködnek
- Természetes objektumok jellemzői:
 - Egyéniség (különállás)
 - Struktúra (adatok)
 - Állapot (adatainak aktuális értéke)
 - Viselkedés (más objektumokkal való kapcsolat, módosítják az állapotot, illetve az állapot befolyásolhatja a viselkedés működését)





Objektumorientált paradigma

MODELLEZÉS OBJEKTUMOKKAL

Vázlatos kép





Objektumok jellemzői

Az objektum: egyediséggel rendelkező diszkrét entitás

- Jellemzői: attribútumok, műveletek
 - Attribútum együttes: objektum állapot; az állapot időbeli változása az objektum viselkedése
 - Műveletek (metódusok): ezek modellezik az objektum viselkedését
- Adat és művelet egy helyen!





Objektumtól objektumig

A valós világ
eleme

Valós objektumok

1. szint

Absztrakció és/vagy formális leírás

Absztrakció

Absztrakt adattípus

2. szint

Megvalósítása programozási eszközzel

(implementáció)

Nyelvi elem

Osztály

3. szint

Példányosítás

Nyelvi elem

Objektum1

Objektum2

Objektum3





Objektumtól objektumig 1. szint

- Tanulmányozzuk a valós objektumokat, és megkeressük azoknak a probléma megoldása szempontjából fontos adatait és viselkedésmintáit. (Az absztrakció fogalmából: „A leglényegesebb tulajdonságok kiemelése”)
- Megkeressük azon objektumcsoportokat, amelyek azonos adatokkal és viselkedésmintákkal rendelkeznek, és elnevezzük ezeket a csoportokat. (Az absztrakció fogalmából: „általánosítás”)





Objektumtól objektumig

A valós világ
eleme

Valós objektumok

1. szint

Absztrakció és/vagy formális leírás

Absztrakció

Absztrakt adattípus

2. szint

Megvalósítása programozási eszközzel
(implementáció)

Nyelvi elem

Osztály

3. szint

Példányosítás

Nyelvi elem

Objektum1

Objektum2

Objektum3





Objektumtól objektumig 2. szint

- Az előzőleg definiált fogalom leképzése a programozási eszközre (osztálydefiníció). Ez egy minta a program működéséhez szükséges objektumok létrehozására





Objektumtól objektumig

A valós világ
eleme

Valós objektumok

1. szint

Absztrakció és/vagy formális leírás

Absztrakció

Absztrakt adattípus

2. szint

Megvalósítása programozási eszközzel
(implementáció)

Nyelvi elem

Osztály

3. szint

Példányosítás

Nyelvi elem

Objektum1

Objektum2

Objektum3





Objektumtól objektumig 3. szint

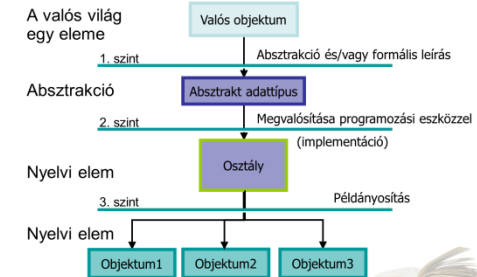
- Akkor és annyi objektumot kell létrehozni a program futása során, amelyek együttműködve képesek a valóságot modellezni.





Példa – 1. szint

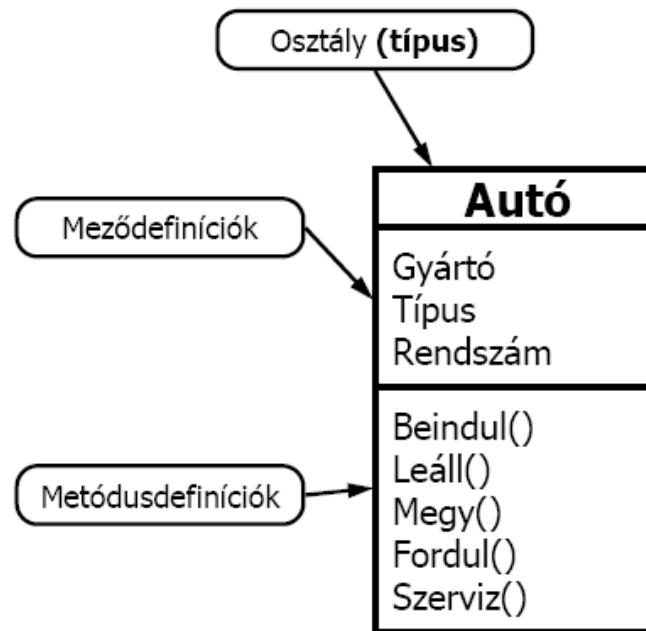
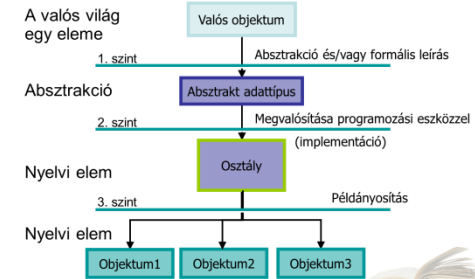
Objektumtól objektumig





Példa – 1. szint

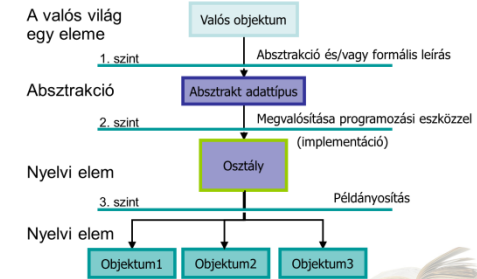
Objektumtól objektumig





Példa – 2. szint

Objektumtól objektumig



```
class Autó
```

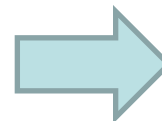
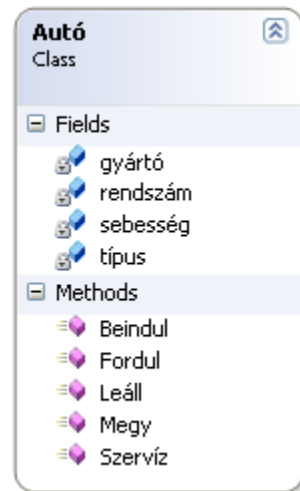
```
{
```

```
private string gyártó;  
private string típus;  
private string rendszám;  
private float sebesség;
```

```
public Autó(string gy, string t, string r)  
{ gyártó = gy; típus = t; rendszám = r; }
```

```
public void Beindul()  
{ sebesség = 0.0F; }  
public void Leáll()  
{ }  
public void Megy()  
{ }  
public void Fordul()  
{ }  
public void Szervíz()  
{ }
```

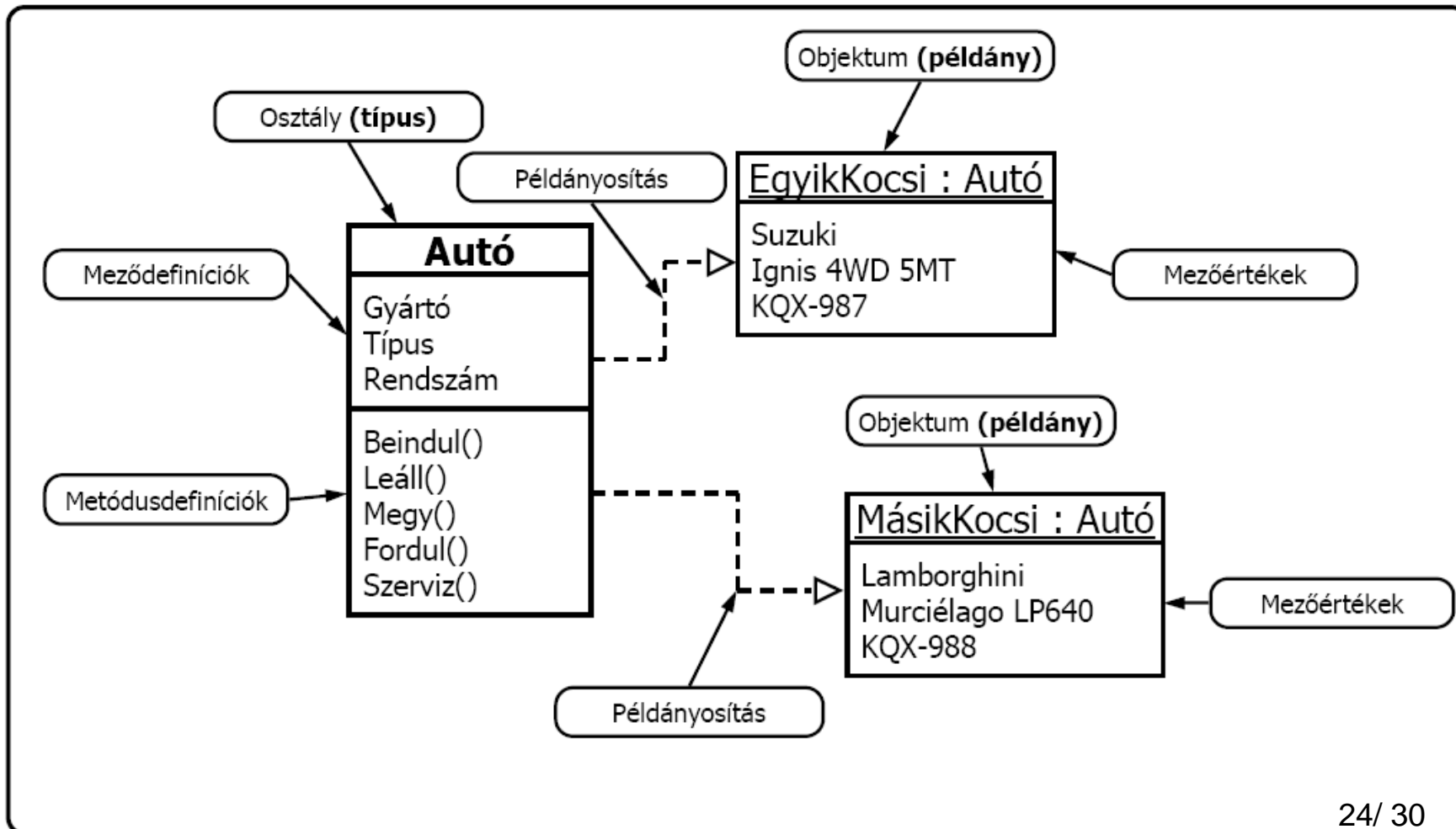
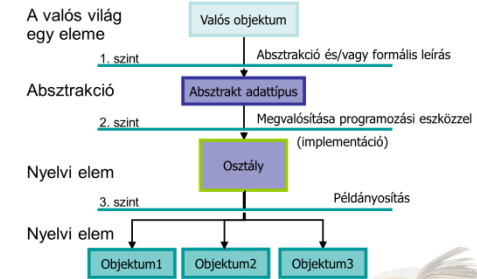
```
}
```





Példa – 3. szint

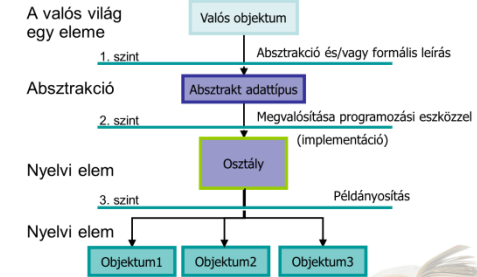
Objektumtól objektumig





Példa – 3. szint

Objektumtól objektumig



```
class Vezérő
```

```
{  
    static void Main()  
    {  
        Autó EgyikKocsi = new Autó("Suzuki", "Ignis 4WD 5MT", "KQX-987");  
        Autó MásikKocsi = new Autó("Lamborghini", "Murcuélago LP640", "KQX-988");  
        EgyikKocsi.Beindul();  
        EgyikKocsi.Megy();  
        MásikKocsi.Beindul();  
        MásikKocsi.Megy();  
        EgyikKocsi.Fordul();  
        MásikKocsi.Leáll();  
        EgyikKocsi.Szervíz();  
    }  
}
```



OO programkészítés folyamata

- A probléma leírásában keressük az objektumokat
- Objektumcsoportok közös tulajdonságait osztályokkal írjuk le
- Megkeressük az osztályok kapcsolatait
- Az osztályokat és kapcsolataikat implementáljuk
- Létrehozunk 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





Objektumorientált program

- Egymással kommunikáló objektumok halmaza
 - Struktúra: osztályok és objektumok közötti kapcsolatok
 - Objektumok: az osztályok példányai
 - Dinamikusan létrehozott objektumok kommunikálnak
 - Működés: üzenetváltások (előre definiált üzenetkészletből)
 - Üzenetáramlás
 - Üzenetek hatására objektum állapotváltozás
 - Megoldás algoritmusát a rendelkezésre álló objektumok közötti üzenetsorozat szolgáltatja
 - (Nem feltételez szekvenciális végrehajtást)





Objektumorientált paradigma

OBJEKTUMORIENTÁLT SZOFTVERCIKLUS

Részletek majd a Szoftvertechnológia című tárgyban





Elemzés, tervezés, megvalósítás

- Objektumorientált elemzés (OOA)
 - A megoldandó feladat leírása osztályok és objektumok segítségével
- Objektumorientált tervezés (OOD)
 - A feladateleírás objektumközpontú részekre bontása (dekompozíciója)
 - Logikai modell – az osztályok és objektumok
 - Fizikai modell – a logikai modellt megvalósító modulok és folyamatok
 - Statikus és dinamikus jellemzők
- Objektumorientált programozás (OOP)
 - A modell megvalósítása egymással kommunikáló, dinamikusan létrehozott objektumok segítségével





Köszönöm a figyelmet!

