

3. Az objektumorientált paradigma alkalmazása.

Osztály, objektum, példányosítás. Ötölődés, osztályhierarchia
Polimorfizmus, metódustúlterhelés. A binárisi eseményre
Abstrakt osztályok és interfészek. Típusutazás

„Nagyrészt szülő-örök” kapcsolat \Rightarrow *Ötölési reláció (specializáció)*
Ötölési reláció

Programozási paradigma

Programozási mód, program felépítésére használt eszközrendszer, a programok alkotóelemeit leíró és meghatározó módszer

Objektumorientált programozás

Egy programozási paradigma, mely főként az objektumok közötti, a program működése során objektumok kommunikációja.

Osztály

Egy csoport, felhasználási típus, mely adattagokból és metódusokból áll. Egy törzsről, mely alapján példányok, azaz objektumok. Valós dolog absztrakciója.

Lehetnek *privát*, *publikus*, *protected* tagjai.

privát \Rightarrow implementáció *interfész* \Rightarrow publikus *privát* \Rightarrow publikus dolog, privát + leszármazottak

Objektum / Példány

Az osztály „törzsről” alapján elkészített adat- és metódusokkal, azaz információval, és ezen információkhoz hozzáféréssel végző műveletek tartalmán. Van állapota, viselkedés, valós időben aszinkronizálható

Példányosítás

Az a folyamat mely során egy osztályból egy konkrét objektumot hozunk létre

Általában konstruktorral történik \Rightarrow lehet adataival, vagy üresen

Öröklődés, osztályhierarchia

Öröklődés

Olyan implementációs és modellezési eszköz, amely lehetővé teszi, hogy egy osztályból olyan újabb osztályokat származtassunk, melyek rendelkeznek az eredeti osztályon már definiált tulajdonságokkal, ismeretekkel és viselkedéssel

Osztályhierarchia

Egymestől leszármazó osztályok hierarchikus rendszere

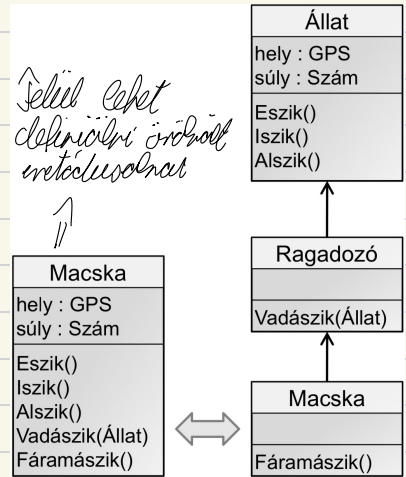
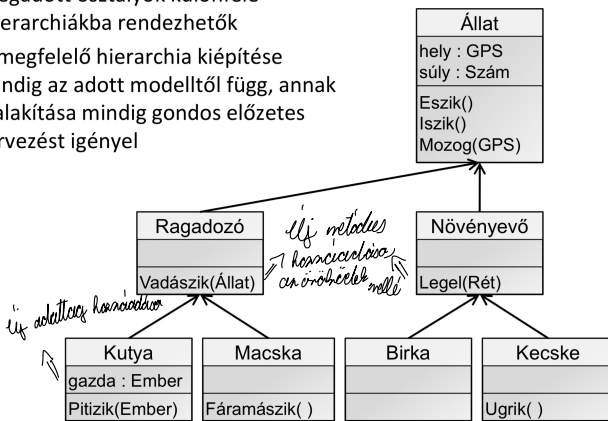
Shorvetten és => Egy szinttel magasabban lévő osztály

Ös => Osztályhierarchiában magasabban lévő osztály

Leszármazott => Osztályhierarchiában alacsonyabban lévő osztály

Osztályhierarchia felépítése

- Megadott osztályok különféle hierarchiákba rendezhetők
- A megfelelő hierarchia kiépítése mindig az adott modelttől függ, annak kialakítása mindig gondos előzetes tervezést igényel



Struktúra, destruktúra nem öröklődik, de kivárat az ésé

Polimorfizmus - többalakúság

Egy objektumnak több "típusa" is lehet, akkor mindegyik "típusként" használható

Egy kutya objektumot beszélgetésként állathént

Metódusátviteli

Több anons nevű, különböző szignatúrájú függvény, a függvényhíváskor megadott paraméterei hatására meg, melyik függvény fog elfolni, futásidőben dönt el => Statikus írtás

Egyesítlenítés

Absztraktnak és koncrnak tartalom műveletét elrejtjük, "megváltjuk"
Egy objektum belső állapota rejtve van, csak a public metódusokon keresztül érhető el, enélkül csak egyben keresztül módosítható \Rightarrow Getter, Setter

Absztraktszintű

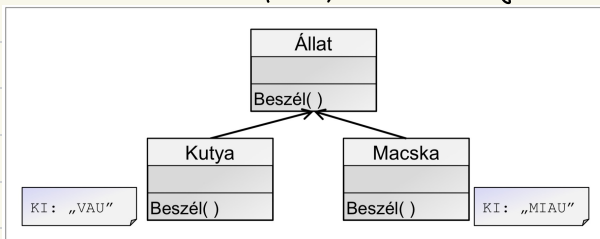
Egy osztály, melyben van legalább egy absztrakts metódus, mely deklarált, de nem megvalósított \Rightarrow Virtuális metódus
Nem példányosítható \Rightarrow célja az örökös hiánypótlása

- A leszármazott osztályok kötelessége az örökölt absztrakts metódusokat implementálni
- Ha nem implementálja, az is absztrakts

Altalános funkcionális inherit a hierarchia felől szintén

Note

Leírás metódus, ami nem írható felül, leírás osztály nem származékosítható



Interfész

Egy műveletkimenet deklaráció, melyben a metódusok nincsenek megvalósítva, amennyiben egy osztály implementál egy interfészt, köteles azokat megvalósítani, az objektum az interfészre, meghatározott viselkedéssel fog rendelkezni

Sokszor még pár dolog, de nincs elég idő

- Konstruktor/Destructor
- Függvény