

4. tétel

Statikus modell. Osztálydiagramok és elemeik

Statikus modell:

A rendszer statikus struktúráját írják le objektum osztályok és relációk segítségével.

A rendszer adott állapotának rögzítése, így az alkalmazás csak egy adott időponthoz kötötten lehetséges. Felépítése a végrehajtható műveletek szerkezetét követi, a modell megegyezik a számítógépes program utasításainak sorrendjével.

Objektum: A rendszer különálló, egymástól megkülönböztethető alapelemei

Osztály: Az azonos jellegzetességű, egy csoportba tartozó objektumokat csoportosítja, azaz egy absztrakt objektumot határoz meg. Az osztály egy általános keretet határoz meg, amelyet az objektumok töltenek fel konkrét tartalommal. Az osztály és az objektum között típus-példány kapcsolat van.

Attribútum: Az objektum tulajdonságát írja le. Az osztály attribútumai tulajdonképpen egy „űrlapot” definiálnak, amelyet az osztályba tartozó objektumok töltenek fel konkrét értékkel.

Művelet: Az objektumok viselkedését írja le. Olyan eljárás, amellyel lekérdezhetők vagy módosíthatók az objektum tulajdonságai.

Metódus: Adott objektum esetén minden művelethez tartozik egy utasítássorozat, amelyet metódusnak nevezünk. A metódus a művelet konkretizálása. A művelet és a metódus között típus-példány kapcsolat van

Kapcsolat: Az objektumnak el kell érni a vele együttműködő objektumokat. Az objektumok közötti kommunikációs útvonalakat kapcsolatoknak nevezzük.

Asszociáció: Az osztály az objektumok közötti kapcsolatot strukturális módon is biztosíthatja. Ezt hívjuk asszociációnak. Az asszociáció és a kapcsolat között szintén típus-példány kapcsolat van.

Osztálydiagram

Az osztálydiagram egy statikus modell, mely a rendszerben előforduló összes osztályt és azok kapcsolatait ábrázolja. Az osztálydiagramon minden osztály pontosan egyszer szerepel. Az osztálydiagramon szereplő osztályok jellemzőit részletesebben ki kell fejteni. Az osztály dokumentációját osztályleírásnak nevezzük. Az osztálydiagramon csak annyi konténert és őst tüntetünk fel, amennyi az egyértelműséghez és a megértéshez szükséges.

Rögzíti az objektumok közötti kapcsolatok szabályait.

Nem a felhasználó hanem a tervező számára készül!

Az osztálydiagram tartalmazza az osztály nevét, az objektumok megvalósítására használt adatokat és típusokat, valamint a metódusokat. Az adatok az állapotok leírásának belső ábrázolásai, a metódusok pedig a műveletek vagy üzenetek. A metódusokat paraméterlista követheti.

Objektumdiagram

Az objektumdiagram tartalmazza az osztály nevét, amelyből példányosítottuk és az objektum pillanatnyi állapotát.

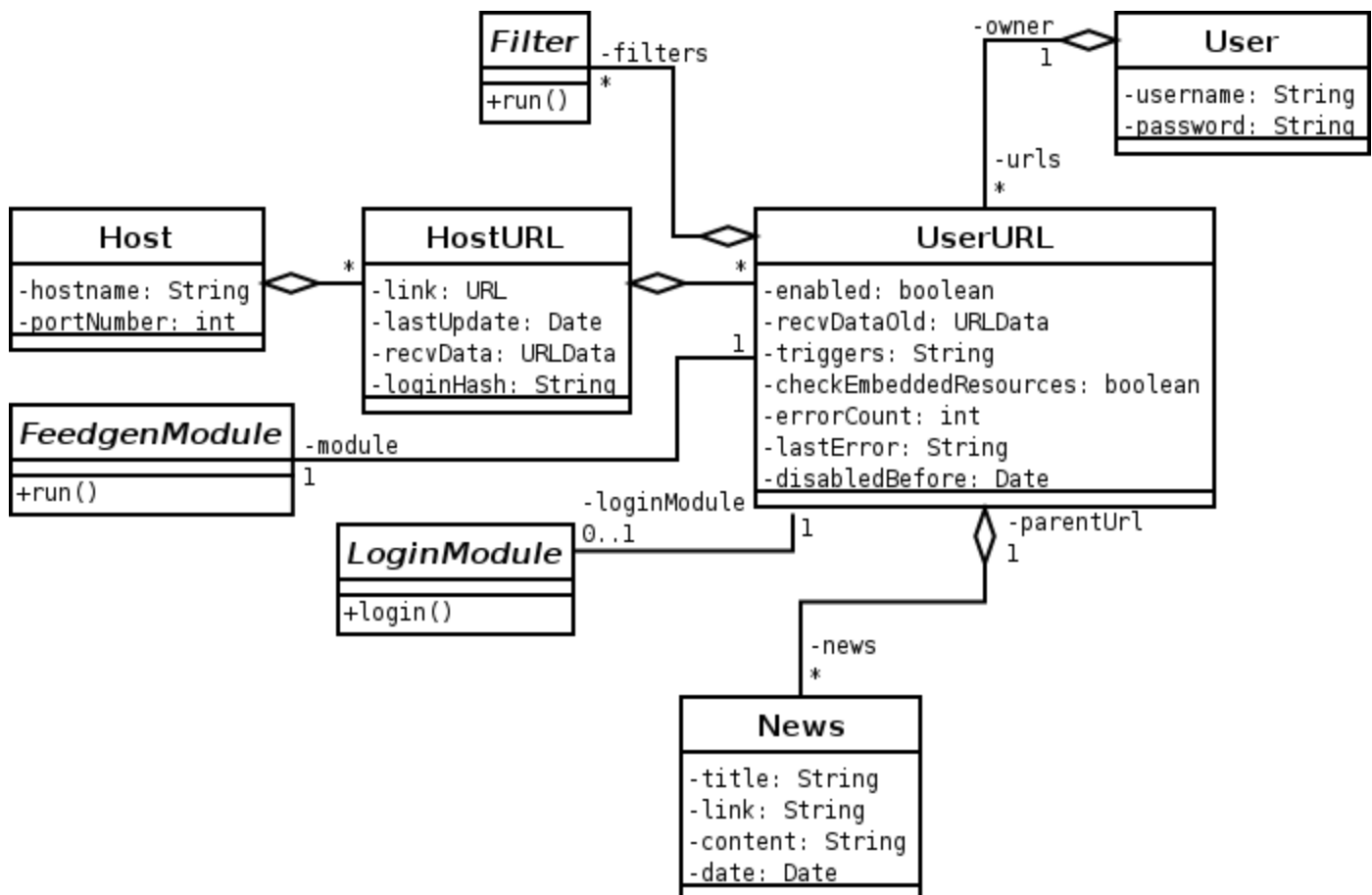
Olyan osztálydiagram, amelyben szerepel legalább egy objektum

Segítségével szemléltethetjük az alkalmazás adott időpillanatban felvett állapotának bizonyos részleteit. Jelölhetjük az attribútum értékeket, a közöttük fennálló kapcsolatokat.

Az osztálydiagram kidolgozásának ajánlott lépései

- **Osztályok azonosítása**
- Nyelvi elemzés → Tartalommal rendelkező főnevek kiemelése
- Osztályozás
- Elhagyandó osztályok kiválasztása:
 - Redundáns osztályok
 - Nem lényeges osztályok
- Attribútumok
- Műveletek
- Szerepek
- **Asszociációk azonosítása**

- Nyelvi elemzés ⇒ igék, igei kifejezések kiemelése, amelyek két objektum közti viszonyt írnak le.
- Például:
 - Fizikai elhelyezkedés, birtoklás (alkotja, része, hozzá tartozik, van neki stb.)
 - Tárgyas igékkel kapcsolatos cselekvés (vezeti, felveszi stb.)
 - Előírt feltételeknek való megfelelés (rokona, tagja stb.)
- Kiemelt asszociációk felülvizsgálata
- Akciók elhagyása
- Magasabbrendű relációk felbontása binárisakra
- Származtatott asszociációk elhagyása
- **Attribútumok azonosítása**
- Nyelvi elemzés ⇒ Objektumot vagy osztályt jelölő főnévhez kapcsolt jelzős vagy birtokos szerkezetek keresése
- Csak azok, amelyek az alkalmazás szempontjából jelentőséggel bírnak
- Attribútum által felvehető értékek halmazának meghatározása
- Kerüljük a származtatott attribútumokat



Ábrázolás

Az osztály jele téglalap, melyben az osztály neve felül szerepel.

Attribútumok megadása: *attribútumnév: típus = kezdeti érték*

típus: a lehetséges értékeket határozza meg.

kezdeti érték: megadása nem kötelező

Az osztálydiagramban megadhatjuk az attribútumok és a műveletek láthatóságát, elérhetőségét a következő jelekkel: `+ public`, `# protected`, `- private`.

Műveletek megadása: *műveletnév(paraméterek) : típus*

műveletnév:

♦Egy objektumon belül létezhethet ugyanolyan nevű függvény, ha más a paraméterezése. A név és a paraméterek együttesen azonosítják a műveletet, a kettő együttesét szignatúrának (signature) nevezzük.

♦Ha ugyanolyan nevű műveletek vannak az objektumban, akkor azt a művelet túlterhelésének (overloading) nevezzük.

Paraméterek megadása: *jelleget név: típus = alapért. érték*

ahol a jelleget lehet:

in: bemenő (ez az alapértelmezett érték)

out: kimenő

inout: be és kimenő

Az attribútum számossága: azt határozza meg, hogy az adott objektumhoz hány érték tartozhat.

Pl.: útvonal[2..*]: String

A leggyakrabban használt számosságok:

1: pontosan 1

*: tetszőleges, 0 vagy több

0..1: opcionális, 0 vagy 1

n: pontosan megadott számérték

0..n: legfeljebb egy adott számérték

Az alapértelmezett számosság attribútumok esetében az 1.

Származtatott attribútum: az osztály attribútuma az osztály egyéb attribútumaiból számítható valamilyen művelet

segítségével. Pl.: $\text{Áfa} \{ \text{Áfa} = \text{Ár} * \text{Áfakulcs} \}$

A származtatott attribútum a programban úgy jelenik meg, mint egy lekérdező művelet, amelynek nincs parameter

Tetszőleges objektum ábrázolása esetén a kettőspont előtti helyet üresen hagyjuk.

Elemek és relációk

A rendszer egyes nézeteinek statikus és dinamikus sajátosságait különböző diagramokkal fejezhetjük ki, amelyek a rendszer elemei közötti relációkat írják le, több különböző szempontból nézve.

Az UML négy relációtípust különböztet meg:

1. függőség (dependency)

Két elem között akkor áll fenn, ha az egyik (a független) elem változása hatással van a másik (a függő) elemre. Kölcsönös a függőség akkor, ha mindegyik elem hatással van a másikra. Grafikus ábrázolásában a szaggatott nyíl a független elem felé mutat.

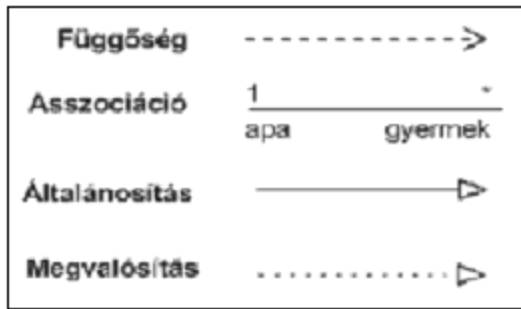
2. asszociáció (association)

Az objektumok kapcsolatát, ezek struktúráját határozza meg. Speciális esete a rész-egész viszony, amely kétféle lehet: aggregáció vagy kompozíció. Aggregáció esetén a rész az egészhez tartozik, de önmagában is létező entitás, míg kompozíció esetén a rész önmagában nem létezhet, csak az egész elemeként. A szemléltető nyílon jelöljük az asszociáció irányát, multiplicitását. Rész-egész viszony esetén az egésznél lévő vonalvég egy csúcsára állított, aggregációnál „lyukas”, kompozíciónál tömött rombusz.

3. általánosítás és specializáció (generalization/ specification)

Az objektumok speciális viszonya, gyermek-szülő kapcsolat, amelyben a főlérendelt elem az általános, az alárendelt a specializált. Ábrázolása egy „lyukas” nyíl, amely a szülő felé mutat.

4. megvalósítás (realization) Annak kifejezése, hogy egy osztály biztosít egy másikat arról, hogy elvégez számára egy bizonyos feladatot. Grafikus szimbóluma egy szaggatott, „lyukas” fejű nyíl.



A relációk UML szimbólumai

Az osztály speciális fajtái

Absztrakt osztály: olyan osztály, amelynek nem lehetnek objektumai, a leszármazottainak viszont igen. Az absztrakt osztály közös jellegetességeket foglal össze, keretet határoz meg.

Interface: speciális absztrakt osztály, amely kizárólag látható műveleteket tartalmaz, attribútumokat nem.

Konkrét osztály: az absztrakt osztály ellentéte, tehát olyan osztály, amelynek lehetnek objektumai

Metaosztály: Olyan osztály, amelynek példányai osztályok.