

# Az UML nyelv

## II. rész

### Diagramok

#### b.) Osztálydiagram (Class)

Olyan statikus modell, amely a rendszerben található összes osztályt és azok statikus (ismeretségi, tartalmazási és öröklési) kapcsolatait ábrázolja. Az összes többi modell erre épül. A használati esetből kikeressük a fontos főveket – melyek valós, modellezni szükséges entitásokat jelölnek –, és ezekből lesznek az osztályok.

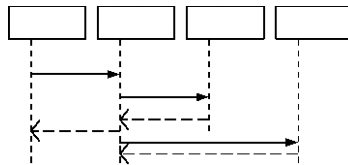
Az ábrázolási módokat már bemutattuk.

#### c.) Objektumdiagram (Object)

Az osztálydiagram példánya. Egy lehetséges rendszerkonfigurációt ad meg, a használt objektumok ábrázolásával és az ezek közötti relációkkal. Jelölésmódja hasonló az osztálydiagramhoz, csak konkrét objektumokról van szó.

#### d.) Szekvenciadiagram (Sequence)

Az objektumok közötti interakciókat követi nyomon, időrendi sorrendben. Hasonló az együttműködési diagramhoz, azonban a fő hangsúlyt az időre, az időrendi sorrendre helyezzük.



– A szekvenciadiagram –

A nyilak segítségével az időrendiséget és az üzenet típusát adhatjuk meg:

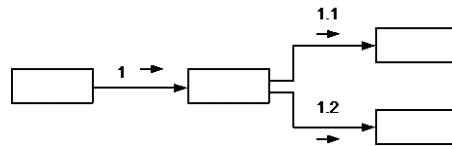
- Szinkron:
- Visszatérés:
- Egyszerű:
- Aszinkron:

A nyilakra ráírhatjuk az üzenet nevét és feltüntethetjük az időtartamot is.

#### e.) Együttműködési, kollaborációs diagram (Collaboration)

Olyan, konkrét objektumokat tartalmazó diagram, amely az objektumok közötti dinamikus kapcsolatokat ábrázolja. Tartalmazza az objektumok közötti ismeretségi, illetve tartalmazási (egész – rész) kapcsolatokat is. Az objektumok közötti információcsere üzenetekkel történik, ezeket olyan nyilakkal ábrázoljuk, amelyekre ráírjuk az üzenet nevét, és meg is számozhatjuk őket, ahhoz, hogy az eseményeket relatív rendezettséggel tüntessük fel.

Az együttműködés azon objektumok között valósul meg, amelyeket asszociációs kapcsolatok kötnek össze.



– Az együttműködési diagram –



Az ismeretségi, illetve a tartalmazási kapcsolatok kifejezésére, megvalósítási módszereire vonatkozóan, a következő információk írhatók a nyilak alá:

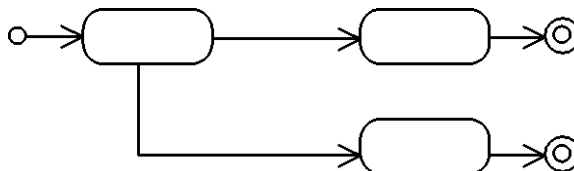
- `<<association>>`: a kliens objektum egy szerver típusú attribútumot tartalmaz.
- `<<global>>`: a kliens objektum egy globális változó által leírt mutatót tartalmaz a szerver felé.
- `<<local>>`: a kliens objektum egy lokális változó (valamelyik metódusának lokális változója) által leírt mutatót tartalmaz a szerver felé.
- `<<parameter>>`: a mutatót egy paraméter segítségével kapja meg a kliens.
- `<<self>>`: a kliens önmagára mutat.

A kis nyilak jelzik a kapcsolat irányát. A *szerver* objektum a nyíl jobboldalán, a *kliens* objektum a nyíl baloldalán található.

#### f.) Állapotdiagram (State)

Megadja, hogy egy adott objektum, egy bizonyos esemény hatására milyen állapotból milyen állapotba megy át. Tulajdonképpen a rendszer vagy az alrendszer életciklusának leírását tartalmazza a következő elemek segítségével:

- állapotok
- események
- integritás és más típusú feltételek
- akciók
- aktivitások
- egy kezdőállapot 
- több végállapot 



– Az állapotdiagram –

Az objektum állapotát az attribútumok (adatok) konkrét értékeivel jellemezzük. Ha egy objektumnak  $n$  attribútuma van, akkor minden egyes állapota  $n$  értéket jelent. Az objektumok állapotait az események változtatják meg.

Az állapotok a következő jellemzőkkel rendelkezhetnek:

- *állapotazonosító* (state): azonosítja az állapotot
- *megjegyzés* (comment): az állapot rövid magyarázó leírása
- *előzetes események* (pre-events): az állapotot előidéző események
- *invariáns* (invariant): ami nem változik
- *utólagos események* (post-events): az állapot megszűnéséhez kötődő események

Az események a következő jellemzőkkel rendelkezhetnek:

- *eseményazonosító* (event): azonosítja az eseményt
- *megjegyzés* (comment): az esemény rövid magyarázó leírása
- *előzetes események* (pre-events): az eseményt megelőző események
- *paraméterek* (parameter): az esemény paraméterei
- *előfeltétel* (pre-condition): az esemény bekövetkezését szükségszerűen megelőző állítás
- *utólagos események* (post-events): az eseményt követő események
- *utófeltétel* (post-condition): az esemény befejezését szükségszerűen előidéző állítás

Az eseményeknek három fázisuk van:

- *belépés* (entry): létrejön egy állapot
- *lefolysis* (event): egy adott állapothoz kötődik
- *vége* (exit): az állapot elhagyása

g.) *Aktivitásdiagram (Activity)*

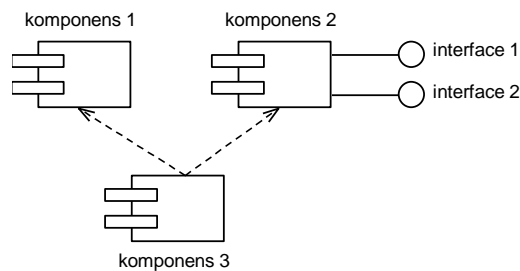
Az állapotdiagram egy formája, a tevékenység implementációját adja meg, a felhasznált műveletek kifejtésével. Tulajdonképpen egy véges állapotú automata.

Az állapotdiagram állapotainak helyére a végrehajtható tevékenységeket tesszük.

Tulajdonképpen a hagyományos adatfolyam diagram egy módosított változata.

h.) *Komponensdiagram (Component)*

Az osztálydiagram osztályainak és egyéb elemeinek fizikai csoportosítását ábrázoló diagram. Szoftverkomponenseket, interfészeket és ezek közötti kapcsolatokat tartalmaz. A szoftverkomponensek *programállományok* (forráskódok), bináris *állományok* és *végrehajtható programok* lehetnek.

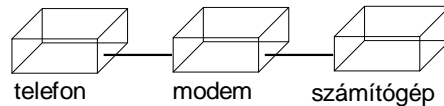


– A komponensdiagram –

*i.) Telepítési diagram (Deployment)*

A rendszerben található fizikai eszközöket (hálózati eszközök, számítógépek, processzorok, nyomtató stb.) és azok kapcsolatait ábrázolja.

Az összekötő egyenesekre felírhatjuk a speciális protokollokat.



– A telepítési diagram –

Kovács Lehel

## Tények, érdekességek az informatika világából

### Az Xbox One konzol

- ☒ Az Xbox One a Microsoft új, nyolcadik generációs videojáték-konzolja, melyet 2013. május 21-én mutattak be a nagyközönségnek. Az Xbox 360-as konzol utódja, és az Xbox konzolcsalád harmadik tagja. Legfőbb vetélytársai a Sony PlayStation 4 és a Nintendo Wii U.
- ☒ Az Xbox One egyik legfontosabb tulajdonsága, hogy fenntartja az x86-64 utasításkészlettel való kompatibilitást, amitől több fejlesztő megjelenését, végső soron a játékok számának növekedését várják; és nem tartja fenn a korábbi Xbox 360-as konzollal való kompatibilitást.
- ☒ CPU: nyolcmagos AMD x86-64 processzor (Jaguar alapú), 28 nm technológia, órajele 1,75 GHz. A processzor két Jaguar egységet tartalmaz, ezekben négy-négy független magot és egy közös osztott  $2 \times 2$  MB L2 gyorsítótárat.
- ☒ GPU: egyedi D3D11.1 osztályú 800 MHz-es órajelű grafikai processzor, 768 GPU-magot tartalmaz 12 számítási egységben (compute unit, CU), AMD GCN architektúra, 28 nm technológia, teljesítménye 1,23 TFLOPS (csúcs shader átviteli sebesség);
- ☒ 32 MiB belső eSRAM memória, ennek sávszélessége 102 GB/s.
- ☒ A CPU, GPU és az eSRAM memória egyetlen egységbe (egylapkás rendszerbe) van összevonva.
- ☒ Rendszermemória: összesen 8 GiB 2133 MHz órajelű DDR3 memória, 256 bites memóriainterfész, sávszélessége 68,3 GB/s. A főmemóriából 3 GiB az operációs rendszer számára van fenntartva, a maradék 5 GiB használható a játékok számára;
- ☒ HDD: 500 GB-os beépített, a felhasználó által nem cserélhető merevlemez, típusa SAMSUNG Spinpoint M8 ST500LM012 (5400 U/min, 8 MiB cache, SATA 3 GB/s); a játékok kötelező jelleggel a merevlemezre installálódnak. A gép külső meghajtókkal lesz bővíthető.
- ☒ Blu-ray/DVD kombinált meghajtó.