# 3. előadás

## Osztályok jelölése UML diagramokban

|  |
| --- |
|  |
|  |
|  |

Felsoroljuk az attribútumokat és műveleteket. (El lehet hagyni.)

Osztály neve mindig nagybetűvel kezdődik + CamelCase. Ékezetet szabad.

Osztály neve félkövéren szedett. Absztrakt osztály félkövér + dőlt.

### Példa osztály

Tetszőlegesen elrejthetőek részletek:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| Kerékpár |
|  |
|  |

 |

|  |
| --- |
| Kerékpár |
|  |

 |

|  |
| --- |
| Kerékpár |

 |

### Példa forráskódja



### Generikus / sablon osztályok

Szaggatott vonal jobb felső sarokba

### Annotáció

Megjegyzés (szemantikus kiegészítés, implementációs kérdéskörbe tartozó elemek jelölésére szolgál)

### Megszorítás

Az attribútumok lehetséges értékeire megszorításokat tehetünk:

|  |
| --- |
| Számla |
|  |

Nem csak attribútumokra tehető megszorítás, UML diagramokban máshol is szerepelhetnek, ám minden esetben kapcsos zárójelek között kell megadni.

## Az osztálydiagram definíciója

Az osztálydiagram a problématérben a megoldás szerkezetét leíró összefüggő gráf, amelynek

* csomópontjaihoz az osztályokat,
* éleihez pedig az osztályok közötti relációkat rendeljük

Osztályok között a következő relációk állhatnak fenn:

* asszociáció (a részt vevő osztályok objektumait kapcsolja össze)
	+ aggregáció (Speciális változata)
		- kompozíció (Még speciálisabb változata)
* öröklődés (osztályok közötti kapcsolat)

### Asszociáció

* Két vagy több osztály objektumainak valamilyen relációval történő összekapcsolása
* Lehet reflexív, azaz azonos osztályú objektumok összekapcsolása is lehetséges
* Az asszociációhoz társulhat annak neve, azonosítója. Nem muszáj elnevezni.
* Iránya is lehet: az aktív objektumtól a passzív objektum felé mutat
* Az összekapcsolt objektumoknak lehet multiplicitása is, szerepe is, és az összekapcsoláshoz minősítő is társulhat
* Navigálhatóság is megadható, amellyel kifejezhető, hogy a társított osztályok objektumai ismerik-e egymást. Elhagyása esetén kölcsönös elérhetőséget tételezünk fel

Egyirányú navigálhatóság, Multiplicitás, Irányított társítás, Szereppel ellátott társítás

Többszörös multiplicitás – mindig gyűjtemény JAVA-ban.

Csúcsok és élek nem tudják, melyik gráfhoz tartoznak.

## Aggregáció

* Speciális asszociáció, fenti tulajdonságok érvényesek rá
* Az általános asszociációnál erősebb kapcsolat, pl.:
	+ Egész és annak részei,
	+ Felépítmény és annak komponensei
* Azt fejezi ki, hogy az egyik osztály objektumai részét képezik egy másik osztály objektumainak
* Az aggregáció tranzitív (ELTE⊇Karok⊇Tanszékek⇒ELTE⊇Tanszékek), antiszimmetrikus (ELTE⊇Karok⤃Kar⊇ELTE), lehet reflexív (Felépítmény: Ha az alkatrész absztrakt osztály, akkor az alkatrészek további alkatrészekből állhatnak.)

Jelölés: kulcsszó helyett aggregáció jel.

## Kompozíció

* Speciális aggregáció, fenti tulajdonságok érvényesek rá
* Azt fejezi ki, hogy az egyik osztály objektumai a másik osztály objektumait fizikailag tartalmazzák
* A kompozíciós kapcsolat és az attribútum jellegű kapcsolat jelentése ugyanaz, csupán a diagramokban jelenik meg másképp (Például fent a Kerékpár osztály tartalmaz egy Szín objektumot.)
* Egy komponens objektum legfeljebb 1 gazdaobjektumhoz tartozhat
* Egy gazdaobjektumnak tetszőleges számú komponense lehet
* A gazdaobjektum és annak komponensei azonos életciklusúak, azaz egyszerre jönnek létre, és egyszerre szűnnek meg

## Specializáció, származtatás és öröklődés

* A specializáció egy általános és egy speciális osztály kapcsolata
* Azt fejezi ki, hogy a speciális osztály az általános osztályból származtatással jön létre
* A speciális osztály átveszi az általános osztály nevét, attribútumait, műveleteit és asszociációit. Ezeket újakkal kiegészítheti, az átvetteket újrafogalmazhatja, de úgy, hogy a speciális objektumok az általános objektumok helyére is behelyettesíthetőek legyenek
* Öröklődéssel valósul meg:
	+ Specifikációs öröklődés: a speciális osztály az általános osztály absztrakt tulajdonságait veszi át
	+ Implementációs öröklődés: a származtatott osztály az általános osztály konkrét tulajdonságait veszi át az absztraktakkal együtt
* Nem szimmetrikus, nem lehet reflexív
* A specializáció és az általánosítás is lehet többszörös
	+ (JAVA-ban nem teljesen: implementációs öröklődés csak egyszeres lehet. A többi kell, legyen.)

Jelölés: A nyíl az általános osztály felé mutat.