Szoftverfejlesztés gyakorlat – 05. óra

September 30

2019

Osztály, Osztálykapcsolat, Öröklődés

Feladatlap

1. Feladat – Felsorolás típus - Enum

Az objectből származik, de nem közvetlenül, hanem az Enum osztályból öröklődik. Mivel ezek is osztályok lesznek, ezért megadható tagváltozó is és megadható neki konstruktor is. Az enum felsorolás típus mindig úgy kezdődik, hogy felsorolom az egyes értékeit. Nem lehet származtatni másik osztályból, hiszen ez már származik az Enum osztályból, a javaban pedig nincs többszörös öröklődés.

Hozz létre egy bolygo package-t, amiben legyen egy új Bolygo felsorolás típus. Az egyes bolygókat, a bolygók tömegeit és sugarait kell tárolni a felsorolásban. Mivel az értékek állandóak, így az egyes elnevezésük is csupa nagybetű lesz.

bolygó	tömeg	sugár
MERKÚR	3,30e+23	2,44 0e6
VÉNUSZ	4,87e+24	6,052e6
FÖLD	5,97e+24	6,378e6
MARS	6,42e+23	3,397e6
JUPITER	1,90e+27	7,1492e7
SZATURNUSZ	5,68e+26	2,5559e7
URÁNUSZ	8,68e+25	6,0268e7
NEPTUNUSZ	1,02e+26	2,4766e7
PLUTÓ	5,98e+24	1,37e6

Legyen minden bolygónak egy tömege és egy sugara, ezért hozz létre egy-egy adattagot számukra. Mivel a felsorolás típus értékei állandóak, így az adattagok értékeit el kell látni final módosítóval. Ahhoz, hogy az egyes értékekhez megadható legyen a tömeg is és a sugár is, szükséges egy konstruktor is, ami egy tömeg és egy sugár paraméterrel fog rendelkezni és beállítja az egyes adattagok értékét.

Készíts egy getTomeg() és egy getSugar() metódust, ami az egyes adattagok értékét adj vissza.

Hozz létre egy publikus statikus változót, amely a gravitációs állandót tárolja. Az értéke: 6,67433e-11; Készíts metódust getGravitacio() néven, ami a bolygóhoz tartozó gravitációt határozza meg, kiszámítása a G*(bolygó tömeg)*(bolygó sugár)^2

Készíts getSuly(int tomeg) metódust, ami a paraméterben kapott tárgy tömegéből kiszámítja a bolygóhoz tartozó súlyát. A számítás a gravitáció és tömeg szorzata lesz.

Számítsd ki, hogy mennyi lenne a súlyod az egyes bolygókon. Célszerű akkor egy Bolygók tömböt létrehozni a Bolygo.values() segítségével.

Objektumorientált tervezés

A tervezés során a valós világ elemeire vonatkozó feladat objektummodelljét állítjuk elő. Az egyes elemeket objektumokkal modellezzük. Ezt a folyamatot gyakran megszemélyesítésnek vagy antropomorf tervezésnek nevezzük.

Általánosságban az Objektum Orientált Programozás tervezés lépései a következőek:

- A feladat pontos specifikálása
- A feladat elvégzéséhez szükséges objektumok meghatározása

- Az objektumok tulajdonságainak (adattagok), tevékenységeinek (metódusok) felmérése.
- A közös tulajdonságok és tevékenységek kiemelése. (Öröklődés, Polimorfizmus)
- Az osztályhierarchia kialakítása általánosítással és specializációval
- Az osztályok/objektumok kapcsolatának, együttműködésének kiépítése.

A feladat megoldásához szükséges objektumok azonosításában segítenek a specifikációban szereplő főnevek, illetve a tevékenységeket pedig igék segítségével a legegyszerűbb meghatározni.

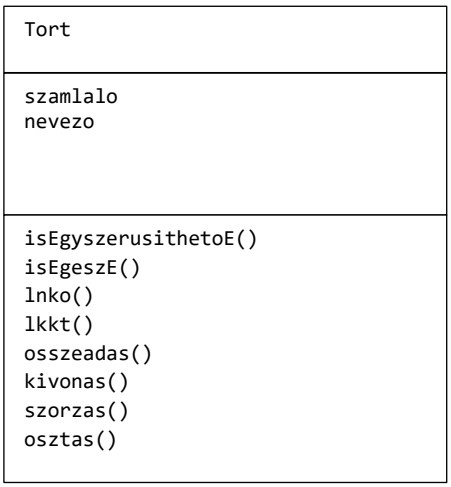
2. Feladat – Törtek

Készíts programot, amely közönséges törtekkel kapcsolatos feladatokat oldja meg. A program forráskódját TortOOP néven mentse.

A törteket úgy tudjuk a legegyszerűbb alakra hozni, ha a számlálóját és a nevezőjét elosztjuk a két szám legnagyobb közös osztójával, és az így kapott szám lesz az új számláló, illetve nevező. A LNKO megvalósításához használja az eukledészi algoritmust.

Két közönséges tört összeadásához a következő lépésekre van szükség.

- Mindkét számot bővíteni kell, azaz mind a számlálóját, mind a nevezőjét ugyanazzal a számmal kell megszorozni. Ezt a bővítést úgy célszerű elvégezni, hogy a közös nevező a két eredeti nevező legkisebb közös többszöröse legyen. Ez lesz az összeg nevezője.
- A két bővített alakú tört számlálóját össze kell adni, ez lesz az eredmény számlálója.

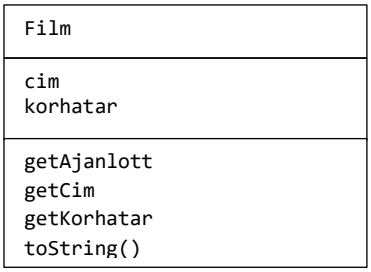


3. Feladat – Filmes alkalmazás

Készíts alkalmazást, amely különböző hordozón tárolt filmek karbantartását végzi az alábbiak szerint:

- Minden filmről tudjuk a címét és korhatárát.
- A konstruktor létrehoz egy példányt a fenti adatokkal.
- Az getAjanlott() metódus igaz értékkel tér vissza, ha a bemenő paraméter értéke nem kisebb, mint a korhatár.
- A getCim() és a getKorhatar() metódusok visszaadják a megfelelő mezők tartalmát.

- A toString() space-szel tagolt stringbe összefűzve adja vissza a mezőket.



Készíts származtatott osztályokat.

- A VHS osztály a szalaghosszt tárolja pluszban.
- a DVD a nyelveket tárolja (egy tömbben).
- A toString() mindig a megfelelő osztály mezőit szolgáltassa.

Teszteld az alkalmazást:

- egy Film típusú elemekből álló tömbben helyezd el különböző hordozón tárolt példányokat.
- Jelenítsd meg egy ciklussal a filmek adatait.
- Keresd ki a leghosszabb VHS filmet.
- Melyik film lett a legtöbb nyelvre lefordítva, ha minden film csak egyszer szerepel az archívumban?

Többalakúság (polimorfizmus)

Metóduspolimorfizmus

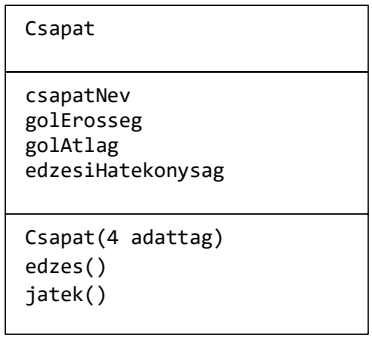
Egy leszármazott osztály egy örökölt metódust újra megvalósíthat (lásd toString()).

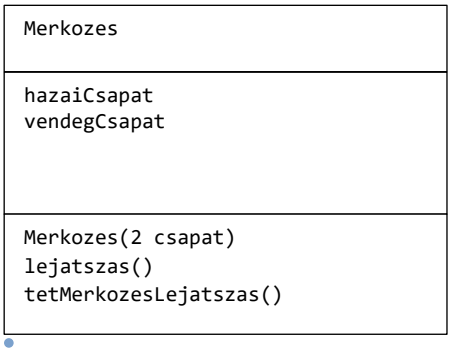
Objektumpolimorfizmus

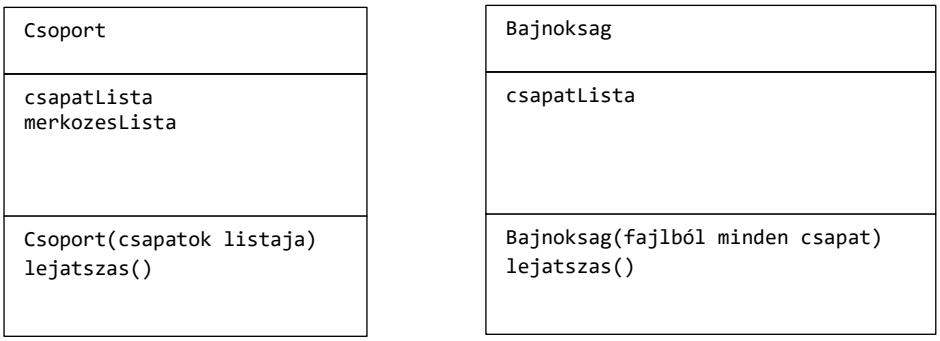
Minden egyes objektum szerepelhet minden olyan szituációban, ahol az ősosztály objektuma szerepelhet, (nem csak a saját osztálya példányaként használható) => A Film ősosztály típusának megfelelő tömbben az utódosztályok példányait is elhelyezhetjük és kényelmesen kezelhetjük. A csak utódokban létező publikus jellemzőket (pl. metódusokat) típuskényszerítéssel (cast-olással) érhetjük el

4. Feladat – Bajnokság szimulátor

Készíts egy bajnokság szimulátor alkalmazást, az alábbiak szerint:







A metódustörzsek meghatározása

Csapat.edzes()

- golErosseg = edzesiHatekonysag* (-0.1,0.1 között)
- golAtlag = edzesiHatekonysag * (-1,1 között)

Csapat.jatek()

- lottGolok = 0..1 * golErosseg * golAtlag

Merkozes.lejatszas()

- csapat.edzes()
- csapat.jatek()
- eredmeny = nyertes csapat (null ha döntetlen)

Merkozes.tetMerkozesLejatszas()

- nem lehet döntetlen, valakinek nyernie kell!

Csoport.lejatszas()

- mindenki játszon mindenkivel

Bajnoksag.lejatszas()

- 4 csoport létrehozása véletlenszerűen, majd a mérkőzések lejátszása
- ezt követően 2 elődöntő mérkőzés a tét mérkőzésnek megfelelően.
- majd bronzmeccs és döntő a tét mérkőzésnek megfelelően.

A csapatok fájl szerkezete, formátuma

```
Csapatnév       golE  golA  edzesH
Magyarország    0.1   1     0.5
Anglia          0.8   3     0.8
Hollandia       0.7   2     0.6
```

Készíts származtatott osztályokat

Csapatból származtatás
Focicsapat, Kézilabdacsapat (30-40 gól), Vízilabdacsapat (5-15 gól)
- játék metódusok túlterhelése

Bajnokságból származtatás
Focibajnokság, Kézilabdabajnokság, Vízilabdabajnokság
- konstruktorok paraméterei a megfelelő csapattípussal hívódnak meg

5. Feladat

Készítsd el az alábbi Osztály diagram alapján a Gerenciseket szimuláló programot.