

# Osztályok létrehozása, adattagok, tagfüggvények, konstruktorok

## 1. Személyek

Készítsünk egy Személy osztályt a következők szerint:

### Adattagok:

- ✎ Szöveg: nev
- ✎ Szöveg: nem
- ✎ Egész: szulev
- ✎ Egész: magassag

### Tagfüggvények:

- ✎ **Egész: Kor(év: Egész)** - A személy életkorát adja meg az adott évben, ha akkor már élt, különben -1 értéket ad vissza.
- ✎ **Szöveg: Termet()**  
A függvény a következő szöveges értékek valamelyikét adja vissza:

	<i>Nő</i>	<i>Férfi</i>
<b>alacsony</b>	<=155	<=165
<b>átlagos</b>	>155 és <170	>165 és <180
<b>magas</b>	>=170	>=180

- ✎ **Logikai: Osszeillo(Szemely s)** - Igaz értéket ad vissza, ha a paraméterként megadott személy különböző nemű, az életkoruk különbsége legfeljebb 5 év és azonos termetű.

Készítsünk programot, melyben létrehozunk két személyt, megadjuk az adataikat billentyűzetről, majd a program írja ki, hogy összeillő-e a két személy.

## 2. Iskola

Egy városi pályázatra iskolák nevezhetnek. A nyertes iskola 1 000 000 Ft nyer, amelyet informatikai eszközök vásárlására költethet. A pályázó iskolának meg kell adniuk a tanulók létszámát, az iskola általános tanulmányi átlagát, valamint az iskola tanulmányi átlagát informatika és matematika tantárgyakból.

A város önkormányzata a következőképpen számolja ki egy iskola pontszámát, ami alapján eldönti, hogy melyik iskola nyeri a pályázatot:

$$\text{pontszám} = \text{létszám}/100 + \text{iskolaátlag} + 2 * \text{matematikaátlag} + 3 * \text{informatikaátlag}$$

Hozz létre egy Iskola nevű osztályt, melyben az adattagok az iskola neve és a pályázatban kért adatok legyenek, valamint legyen egy Pontszám() tagfüggvénye, amely kiszámolja a képlet alapján az iskola pontszámát!

Budapesten a pályázatra 14 iskola jelentkezett. Készíts teszt osztályt, melyben a budapest.txt fájlból olvasd be az adatokat az iskolák nevű tömbbe, majd listázd ki az iskolák nevét és pontszámukat, és add meg a nyertes iskola/iskolák nevét!

## 3. Autók

Készítsünk egy Auto osztályt a következő tagokkal:

### Adattagok:

- ✎ Szöveg: rendszam
- ✎ Egész: gyartev
- ✎ Egész: teljesitmeny

### Tagfüggvények:

- ✎ **Egész: Gepjarmuado()** metódus, mely az alábbiak szerint visszaadja a fizetendő gépjárműadó éves mértékét:
  - ⇒ a jármű gyártási évében, illetve az azt követő három évben 345 forint kilowattónként;
  - ⇒ a gyártási évet követő 4.-7. évben 300 forint kilowattónként;
  - ⇒ a gyártási évet követő 8.-11. évben 230 forint kilowattónként;
  - ⇒ a gyártási évet követő 12.-15. évben 185 forint kilowattónként;
  - ⇒ a gyártási évet követő 16. évet követően 140 forint kilowattónként

Készítsünk egy teszt osztályt, amelyben a billentyűzetről bekérjük két autó adatait, és kiírjuk, hogy mennyi az autók után fizetendő gépjárműadó mértéke az idei évben!

#### 4. Téglalapok

Készítsünk egy `Teglalap` osztályt a következő tagokkal:

Adattagok:

- ↳ Egész: `width`
- ↳ Egész: `height`
- ↳ Egész: `left`
- ↳ Egész: `top`

Tagfüggvények:

- ↳ Egész: `Terulet()`
- ↳ Egész: `Kerulet()`
- ↳ `Kirajzol()`
- \*↳ Logikai: `Diszjunktak(Teglalap t)` – igaz, ha nincs közös része a paraméterként megadott téglalappal

Készítsünk egy teszt osztályt, melyben:

- a. véletlenszerű adatokkal generálunk 5 olyan téglalapot, amely kifer a consol-ablakra
- b. a legnagyobb kerületűt és a legnagyobb területűt kirajzoljuk a képernyőre
- c. a kirajzolt téglalapok középebe kiírja a kerületet, illetve a területet

#### 5. SIM kártya

Készítsünk egy `SimKartya` osztályt a következők szerint:

Adattagok:

- ↳ Szöveg: `PIN`; kezdőértéke `0000`
- ↳ Egész: `egyenleg`

Tagfüggvények:

- ↳ Logikai: `PinEllenorzes(Szöveg: pinKod)` – igaz (true) értéket ad vissza, ha jó a pinkód, hamisat (false), ha nem jó
- ↳ `PinBeallitas(Szöveg: ujPin)` – megkérdezi a jelenlegi PIN-kódot, és csak akkor állítja át a PIN-t a megadott ujPin-re, ha helyesen adják meg a jelenlegi PIN-kódot – ebben az esetben igaz értéket ad vissza, ellenkező esetben false értéket
- ↳ `EgyenlegFeltolt(Egész: osszeg)` – a megadott összeggel növeli az egyenleget

A fenti osztály segítségével készítsünk programot, mely felkínál három menüpontot:

- P: PIN beállítás
- E: Egyenleg feltöltés
- L: Egyenleg lekérdezés

A megfelelő menüpont kiválasztása után lehetőséget ad a PIN beállítására, illetve egyenlegfeltöltésre vagy lekérdezésre, majd a megfelelő üzenet után (sikerült-e a kért művelet) bezáródik.

#### 6. Termékkészlet (Forrás: Nagy Attila, Óbudai Egyetem – Neumann János Informatikai Kar, 2010)

Adott egy képzeletbeli vállalkozás, mely szeretné elektronikusan nyilvántartani az általa végrehajtott termék eladásokat, és beszerzéseket. Ennek megvalósításaként készíts egy programot, ami a feladatot ellátja, azaz képes tárolni, hogy az egyes termékekből mennyi van a raktáron, mennyi az egységáruk ezeknek, és mi az eladások által szerzett összes bevétele a vállalatnak. A programot az alábbiak szerint készítsd el:

- a. Készíts egy `Termek` nevű osztályt. Az osztályban legyenek az alábbi adattagok:

- `Nev:` A termék neve
- `EgysegAr:` A termék ára
- `Kedvezmeny:` A termékre igénybe vehető kedvezmény százalékláb formájában (0-1 közti valós érték)
- `Keszlet:` Hány darab van a raktáron ebből a termékből?

- b. Készíts két konstruktort az osztályhoz. Az egyiknek lehessen megadni paraméterként egy nevet, és egy árat. A másinak lehessen megadni nevet, árat és raktárkészletet. Mindkét konstruktor inicializálja megfelelően az objektum mezőit a kapott paraméterek alapján. Annál a konstruktornál, ahol nem kellett raktárkészletet megadni, állítódjon be az 1 érték. Kedvezmény alapból ne legyen (0.0)!
- c. Készíts egy `Informacio` nevű metódust, ami szépen formázottan kiírja az osztály legfontosabb adatait (név, ár, raktárkészlet) az alábbi formában. Pl.:

Alma	100,00 Ft	(raktáron: 700 db)
Körte	110,00 Ft	(raktáron: 400 db)
Kenyér	220,00 Ft	(raktáron: 30 db)

- d. Készíts egy **Eladas** nevű metódust, ami paraméterében kapja, hogy hány darab termék eladása történik. A metódus csak akkor csökkentse az adott termék raktárkészletének értékét, ha a pozitív darabszámú terméket szeretnének vásárolni. Írjon ki hibáüzeneteket, ha az eladásra szánt termék már nincs a raktáron, illetve akkor is, ha a raktáron lévő készlet nem elég az eladás teljesítéséhez (raktárkészlet < eladásra szánt darabszám).
- e. Készíts egy **Beszerzes** nevű metódust is, ami paraméterben kapja, hogy hány darab termék beszerzése történik. A metódus a raktárkészlet mező tartalmát változtassa meg értelemszerűen a beszerzett mennyiséggel. Csak akkor módosítsa a raktárkészletet, ha a darabszám nem negatív.
- f. Végül vegyél fel egy statikus, osztály szintű adatmezőt is, ami az összes termék eladásából keletkezett összbevételt tárolja. Ezt az eladás és beszerzés metódus módosítsa megfelelően!
- g. Mindezek után a főprogramban hozz létre néhány példányt a termék osztályból, a konstruktorokat szabadon megválasztott paraméterekkel hívd meg, majd mindegyik termék példányra hívd meg az egy ciklusban ötször az eladás metódust úgy, hogy a darab paramétert véletlenszám generátorral állítsd elő. Írd ki a képernyőre a termékek információit az eladások előtt és után is.

A programnak a kimenete az alábbi példában illusztrálthoz hasonlítson:

A termékek listája:

Alma	100,00 Ft	(raktáron: 700 db)
Körte	110,00 Ft	(raktáron: 400 db)
Kenyér	220,00 Ft	(raktáron: 30 db)

Tranzakciók:

```

221 db alma sikeresen eladva! Raktáron van még: 479
46 db körte sikeresen eladva! Raktáron van még: 354
22 db kenyér sikeresen eladva! Raktáron van még: 8
123 db alma sikeresen eladva! Raktáron van még: 356
1 db körte sikeresen eladva! Raktáron van még: 353
0 db kenyér sikeresen eladva! Raktáron van még: 8
185 db alma sikeresen eladva! Raktáron van még: 171
4 db körte sikeresen eladva! Raktáron van még: 349
15 db kenyér eladása sikertelen! Nincs ennyi a raktáron! Készlet: 8
143 db alma sikeresen eladva! Raktáron van még: 28
18 db körte sikeresen eladva! Raktáron van még: 331
24 db kenyér eladása sikertelen! Nincs ennyi a raktáron! Készlet: 8
172 db alma eladása sikertelen! Nincs ennyi a raktáron! Készlet: 28
54 db körte sikeresen eladva! Raktáron van még: 277
31 db kenyér eladása sikertelen! Nincs ennyi a raktáron! Készlet: 8

```

A termékek listája az eladások után:

Alma	100,00 Ft	(raktáron: 28 db)
Körte	110,00 Ft	(raktáron: 277 db)
Kenyér	220,00 Ft	(raktáron: 8 db)

Az összes bevételünk: 85 570,00 Ft

## 7. Számok (2013. május emelt idegen nyelvű informatika érettségi programozás feladata)

A *Szereti Ön a számokat?* internetes vetélkedőben a versenyzők olyan kérdéseket kapnak, amelyekre egy egész számmal kell válaszolniuk, melyek mindegyike 0 és 1 milliárd közé esik. A kérdések különböző témakörökből származnak (pl. magyar, matematika, történelem, kémia), és nehézségüktől függően 1-től 3-ig terjedő pontszámot érnek.

A feladatokat a verseny szervezői egy adatfájlban tárolják. A fájlban minden feladat két sorban helyezkedik el. Az első sor tartalmazza a kérdést, a második pedig – egy-egy szóközzel elválasztva – a helyes választ, a helyes válaszért adható pontszámot és a témakör megnevezését. A fájlban ékezetes betűk nem szerepelnek, pl. a „gyümölcsízű” szó helyett a „gyumolcsizu” szót írták be.

Például:

Mikor volt a mohacsi vesz?

1526 1 tortenelem

A példában szereplő kérdés: Mikor volt a mohacsi vesz? A helyes válasz: 1526. A helyes válasz 1 pontot ér, és a kérdés a történelem témakörbe tartozik. Az adatfájl még csak részben készült el. Az Ön feladata ennek a félkész adatfájlnak a tesztelése.

A fájl legfeljebb 100 kérdést tartalmaz. Biztosan van benne matematika, történelem és földrajz feladat, de más témakörök is előfordulnak.

Készítsen programot, amely a `felszam.txt` állomány adatait felhasználva az alábbi kérdésekre válaszol! A program forráskódját mentse `szamok` néven! (A beolvasott fájl adatait és a felhasználó válaszainak az érvényességét nem kell ellenőriznie.)

- a) Olvassa be a `felszam.txt` állományban talált adatokat, és azok felhasználásával oldja meg a következő feladatokat!
- b) Hány feladat van az adatfájlban? A választ írassa ki a képernyőre!
- c) Határozza meg, hogy hány matematika feladat van az adatfájlban, és ezek közül hány feladat ér 1, 2, illetve 3 pontot! A választ egész mondatban írassa ki a képernyőre!

Például:

Az adatfájlban 20 matematika feladat van, 1 pontot ér 10 feladat, 2 pontot ér 6 feladat, 3 pontot ér 4 feladat.

- d) Mekkora a fájlban található válaszok számértéke? A választ egész mondatban írja ki a képernyőre!
- e) Milyen témakörök szerepelnek ténylegesen az adatfájlban? Írassa ki a témakörök nevét a képernyőre úgy, hogy minden előforduló témakör pontosan egyszer jelenjen meg!
- f) Kérje be egy témakör nevét, és véletlenszerűen sorsoljon ki egy kérdést ebből a témakörből! Sorsoláskor ügyeljen arra, hogy az adott témakörbe eső valamennyi feladatnak legyen esélye! (Feltételezheti, hogy a felhasználó helyesen adta meg egy létező témakör nevét.) Írassa ki a kérdést, kérje be a felhasználó válaszát, majd adja meg a válaszáért járó pontszámot! (Helytelen válaszáért 0 pont jár.) Ha a válasz helytelen volt, a helyes választ is közölje! A párbeszéd az alábbi formában jelenjen meg:

Például:

Milyen témakörből szeretne kérdést kapni? történelem  
Mikor volt a mohacsi vesz? 1514  
A válasz 0 pontot ér.  
A helyes válasz: 1526

- g) Generáljon egy 10 kérdésből álló feladatsort véletlenszerűen úgy, hogy egyetlen feladat se szerepeljen benne kétszer! (Ügyeljen azonban arra, hogy minden beolvasott feladatnak legyen esélye a kiválasztásra!) A feladatsort írassa ki a `tesztfel.txt` állományba az alábbi formátumban! (Az első szám a helyes megoldásért járó pontszám, ezt követi a helyes válasz, majd a kérdés egy-egy szóközzel elválasztva.) Az állomány végére írassa ki a feladatsorra összesen adható pontszámot is!

Például:

...  
1 1526 Mikor volt a mohacsi vesz?  
...  
A feladatsorra összesen 20 pont adható.