

| | | | | | | | | | |
|------|--|----------------|--|--|--|--|--|--|--------------|
| NÉV: | | NEPTUN KÓD: | | | | | | | X CSOPORT |
|------|--|----------------|--|--|--|--|--|--|--------------|

| | |
|---|--|
| ! | A beadandó solution elnevezési mintája: NEPTUNKOD_X |
| | A beadott dolgozat fordítási hiba esetén nem értékelhető. Kommentezett részre pont nem szerezhető. |
| | A feltöltendő file-t a megfelelő kurzushoz tölts fel, egyéb esetben a ZH nem értékelhető. |

ZÁRTHELYI DOLGOZAT

A Covenant űrhajó lezuhanását követően, a legénység nyomozásba kezdett. Az Ön feladata, mint egyetlen életben maradt mérnökinformatikus, hogy a feladatok elvégzésével a nyomozást segítse.

ALIENS.txt szöveges file, amelyben az űrlényekről szóló adatok vannak:

“ Facehugger : 2030-9-7 : 6 : 8 { 9936 ; 3398 ; 3613 ; 3506 ; 3613 ; 6227 ; ”

a sablon: *típus : kikelés_ideje : xPoz : yPoz { megölt_emberek_azonosítói_felsorolva*

PEOPLE.txt szöveges file, amelyben az űrlényekről szóló adatok vannak:

“ 1728 : KLAUDIA : 2027-1-17 : 0 : 1 ”

a sablon: *ID : név : hajóraszállás_ideje : férfi_nő_logikai_értékkel : él_e_még_logikai_értékkel*

Alien osztály (10 p)

- A osztály rendelkezzen a következő adattagokkal, melyekhez tartozik olvasható és írható tulajdonság is:
 - (string) azonosító id, (string) típus, (DateTime) kikelés ideje, (int) x pozíció, (int) y pozíció, (string) ember azonosítók tömbje, (Ember) megölt emberek tömbje (maximum 7 db)
- *Alien()* konstruktor, amely megkapja a típus, kikelés ideje, x valamint y pozíciókat!
- *IDGeneral()* metódus, amely az alábbi algoritmus alapján generál azonosítót az űrlénynek:
 - [kikelés idejének számjegyeinek összege] - [típusának első két karaktere]
 - ahol az egyes [...] helyekre a megfelelő adatokat állítsa elő (a [és] karakterek NEM kelljenek)
 - A metódus minden esetben, az adott példány létrehozásakor hívódjon meg, és biztosítsa, hogy csak osztályon belül lehessen meghívni a metódust.
- *Jelolo()* metódus, amely a szörny típusának kezdőbetűjét adja vissza, nagybetűs formában.

Ember osztály (5 p)

- (int) ID, (string) név, (DateTime) hajóraszállás dátuma, (enumeráció Férfi/Nő) nem, (bool) él-e még adattagokkal dolgozzon, melyeknél biztosítsa, hogy az egyes adattagok írható és olvasható tulajdonságokkal egyaránt rendelkezzenek.
- *Ember()* konstruktor, amely minden paramétert fogad, amely az osztály adattagai között megtalálható. Biztosítsa továbbá, hogy csak ID és név esetén is lehessen embert létrehozni.
- *Ember()* konstruktor, amely az EMBER.txt fájlban található sorok egyikét kapja meg bemenetként, dolgozza fel és állítja elő a megfelelő értékeket az adattagok számára.

folytatás a túloldalon

Eszkoztar osztály (25 p)

- Az osztály rendelkezzen egy darab Alien-ek tárolására alkalmas tömbbel, melyhez készítsen írható és olvasható tulajdonságot. Készítsen továbbá, egy osztályszintű szintén Alien-ek tárolására alkalmas másik tömböt, azonban ebbe majd csak a „Chestbuster” típusokat fogja tárolni.
- *Eszkoztar()* konstruktor, amely első körben az *Import* belső metódust hívja meg, majd a *ChestbusterKivalogat* metódust.
- *Import()* belső láthatósággal rendelkező metódus, amely a(z) ALIENS.txt egyes sorait bejárva, létrehozza a szükséges objektumokat és eltárolja azokat. Figyeljen arra, hogy a tömb mérete dinamikusan változzon a beolvasott fájl alapján. Egyelőre csak az emberek azonosítóit dolgozza fel, a konkrét emberek meghatározását a következő metódus végzi. Az Alien-ek által megölt emberek számossága maximum 10.
- *EmberVisszaallit()* metódus, amely célja, hogy az Alien-ek létrehozásakor az adatbázis csak az ember azonosítókat tárolta el. A mentőakció célja, hogy kiderítsük, az egyes azonosítók mögött milyen valós személyek állnak. Ennek megfelelően a rendszerbe már beimportált Alien-ek gyűjteményét dolgozza fel és az emberek azonosítóit alapján, az PEOPLE.txt megfelelő embereit keresse meg, majd Ember objektumként tárolja el a megfelelő Alien-hez.
- *ChestbusterKivalogat()* metódus, amely a már beimportált Alien-eket tartalmazó tömböt feldolgozza és azokból kiválogatja a „Chestbuster” típusokat a számukra létrehozott osztályszintű tömbbe. Figyeljen arra, hogy a tömb mérete dinamikusan legyen, azaz csak azokat az elemeket tároljuk benne, amelyek valóban relevánsak.
- *AlienMegjelenit()* metódus, amely konzolra kirajzolja az Alien-ek legutolsó ismert tartózkodási helyét, (x,y) koordináták segítségével. Az elhelyezéshez használja a kurzor pozícionálizálását valamint a *Jelolo* függvényt. Az ismert terület 50*50-es méretű, így ekkora tartományban dolgozzon. Azt követően, a kurzort állítsa vissza a terület alá, bal oldalra.
- *ChestbusterKikeltek()* metódus, amely a chestbursterek közül válogatja ki azokat, akik 2020 előtt keltek ki.
- *MegoltEmberek()* metódus, amelyek az Alien-ek által megölt embereket sorolja fel és írja ki a konzolra a következő formában:
 - [ALIEN_TÍPUSA]:
 - [egy tabulátornyi hely]: [megölt_ember_1]
 - [egy tabulátornyi hely]: [megölt_ember_2], stb...
- *Log()* metódus, amely az Alien-eket tartalmazó tömböt feldolgozza és fájlba írja az egyes elemeket. Fájlba íráskor az egyes Alien-ek azonosítóját kövesse egy kettőspont, majd felsorolás szerűen az emberek nevei. Az embereket kötőjellel tagolja.
- *Deaconok()* metódus, amely egy kimeneti tömbbe kigyűjti, hogy hány Deacon típusú Alien található a térkép felső részén. Felső rész alatt a [0 - 9]-ig terjedő sorokat értjük.

Minden részfeladat elvégzését a megadott, külön metódussal és osztállyal valósítsa meg! Az egyes metódusok megvalósítása során – ahol lehetséges – a tanult programozási tételeket és rendezési algoritmusokat használja!

ZÁRTHELYI DOLGOZAT BEADÁSA

A kész feladatot **teljes egészében** (egész solution mappa), .zip (nem .rar, .7z stb.) formába tömörítse be, majd a <http://zh.nik.lan> leadó felületen, a **megfelelő kurzust kiválasztva** töltsse fel!

Megfelelő kurzus a Szoftvertervezés és –Fejlesztés I. (nappali / NIXSF1HBNE), azon belül pedig az az alkalom, amelyre Ön jár. Rossz kurzusra való feltöltés esetén a zárthelyi nem értékelhető. Fordítási hibával rendelkező program nem értékelhető.