

# Programozás II. Házi Feladat

SZTE Szoftverfejlesztés Tanszék

2022. ősz

## Ismertető

- A programot C++ nyelven kell megírni.
- **A benyújtandó fájl neve kötelezően feladat.cpp.**
- A megoldást a *Bíró* fogja kiértékelni.
  - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a  
-std=c++1y -static -O2 -DTEST\_BIRO=1  
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
  - Futási hiba 6: Memória- vagy időkorlát túllépés.
  - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
  - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túindexelés, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
  1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
  2. A kapott url formátuma:  
`https://biro2.inf.u-szeged.hu/Hallg/IBL302g-1/1/hXXXXXX/4/riport.txt`
  3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.
- A megvalósított függvények semmit se írnak ki a standard outputra!

# Feladat

## 1. feladat (3 pont)

Készíts egy `Telepes` nevű osztályt, mely a különféle bolygókon letelepedett emberek adatait tárolja. A telepesek adattagjait az 1. táblázat foglalja össze. Az adattagok legyenek privát láthatóságúak. Mindegyik rendelkezzen a táblázat szerinti getterrel és setterrel.

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
<code>nev</code>	<code>std::string</code>	A telepes neve	<code>get_nev</code>	<code>set_nev</code>
<code>szul_bolygo</code>	<code>std::string</code>	Születési bolygó	<code>get_szul_bolygo</code>	<code>set_szul_bolygo</code>
<code>bolygo</code>	<code>std::string</code>	Jelenlegi bolygó	<code>get_bolygo</code>	<code>set_bolygo</code>
<code>ero</code>	<code>unsigned</code>	Munkavégző képesség	<code>get_ero</code>	<code>set_ero</code>

1. táblázat. Telepes adattagok

## 2. feladat (8 pont)

Készítsd el a `Telepes` osztály konstruktorait!

- Legyen egy konstruktora, mely négy paraméterrel rendelkezik és mind a négy adattagot a paramétereknek megfelelően állítja be. A paraméterek sorrendje az 1. táblázat sorainak sorrendje legyen, tehát elől a sztringek majd utolsónak a szám.
- Rendelkezzen egy default konstruktorral, amely a `nev`, `szul_bolygo` és `bolygo` adattagokat üres sztringre inicializálja, az `ero`-t pedig 1-re.
- Legyen egy konstruktora, amely két sztringet és egy `unsigned` paramétert vár. Az első sztring paraméter alapján a telepes neve legyen inicializálva, míg a második sztring paraméter a telepes jelenlegi bolygójának a neve. Ez alapján legyen inicializálva a `bolygo` és a `szul_bolygo` adattag is. Az `ero` érték a harmadik, `unsigned` paraméter alapján legyen beállítva.
- Lehessen úgy is lepdányosítani az osztályt, hogy nem adjuk meg a konstruktorban a harmadik, `unsigned` értéket, csak a két sztringet (név, bolygó). Ilyenkor az `ero` érték legyen 1-re állítva.

## 3. feladat (4 pont)

Készíts egy `kivandorlo()` metódust a `Telepes` osztályba, mely segít eldönteni, hogy a telepes kivandorlónak minősül-e. A visszatérési értéke akkor legyen igaz, ha a `szul_bolygo` és `bolygo` adattagok értéke nem egyenlő egymással. Két bolygó neve egyenlőnek számít, ha csak kisbetű/nagybetű eltérés van a nevükben (pl. „Mars” és „mars” egyenlőek). A bolygók nevei ascii karakterek (tehát nem tartalmazznak pl. ékezetes betűket). Ha mindkét adattag értéke üres sztring egyenlőnek tekinthetőek.

**Tipp:** Módosít ez a metódus az objektum adattagjain? Ennek megfelelően hozd létre.

## 4. Feladat (7 pont)

Készíts egy `dolgozik(std::string munkak)` metódust a `Telepes` osztályban, amely naplózza a telepes munkavégzését. Egy `std::string`-et adjon vissza.

A paraméterként kapott sztring reprezentálja az egymás után következő feladatokat pontosvesszővel elválasztva. A feladatok alfanumerikus kódokkal vannak reprezentálva, pl.: `b11` - kapálás, `100` - ültetés, `qx` - napelemek tisztítása. A telepesek minden feladatot el tudnak végezni, azonban csak annyi darabot, amennyi az `ero` értékük, utána kifáradnak. A metódus célja, hogy levágja azokat a munkákat az input sztring elejéről, amiket a telepes elvégzett. Az így módosított sztringgel térjen vissza, azaz a visszaadott sztring már csak azokat a munkákat tartalmazza, amiket az aktuális telepes nem bírt elvégezni és egy másik telepesre kell hagyni. Üres sztringet adjon vissza, ha nincs több elvégzendő munka. Ha csak egyetlen elvégzendő munka van, akkor nem található pontosvessző a munkasorozat sztringben. A telepes objektum ne módosuljon!

### Példa működés:

Kezdetben: `munkak - „b2;b11;c3;x823”`, `ero - 2`

A visszaadott érték: `„c3;x823”`, `ero - 2`

