

Szkriptnyelvek - Python ismertető

- A programot Python nyelven kell megírni.
- A benyújtandó fájl neve: `feladat.py`
 - Egy Python nyelven írt, szöveges fájl (nem zip, rar, stb.)
 - Ez csak a feladatban kért dolgokat tartalmazza! Amennyiben saját inputtal teszteléd a kódot lokálisan, úgy feltöltés előtt a tesztelő kódrészletet kommenteld ki!
- A megoldást Bíró webes felületén (<https://biro.inf.u-szeged.hu>) keresztül kell benyújtani és 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 **Python 3.10.1** interpreterrel fogja futtatni, és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 2 másodpercnél és hiba nélkül fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- Ha 3 teszteset futási ideje túllépi a fenti időkorlátot, a tesztelés befejeződik, a pontszám az addig szerzett pontszám lesz.
- A riport.txt megtekinthető az alábbi módon:
 1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
 2. A kapott url formátuma:
https://biro.inf.u-szeged.hu/Hallg/IB370G/FELADAT_SZAMA/hXXXXXX/4/riport.txt
 3. Az url-ből visszatörölve a 4-esig (riport.txt törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 25 alkalommal lehet benyújtani, a megadott határidőig.
- A munkád során figyelj arra, hogy pontosan kövesd a feladatban leírtakat, az elnevezéseket!
- A fájl elejére kommentbe írd be a neved, Neptun és h-s azonosítód az alábbi formában:

```
# Nev: Vezeteknev Keresztnev  
# Neptun: NEP4LF  
# h: h123456
```

Szkriptnyelvek 1. ZH

Borfesztivál Szegeden (10 pont)

Szegeden megrendezik a híres Borfesztivált, ahol minden évben több ezer ember érkezik, hogy különböző borokat kóstoljon meg. A Szegeden eltöltött este során rögzítik a vásárlók neveit és az általuk vásárolt borok mennyiségét.

Készíts egy `borfesztival` nevű függvényt, amely egy szöveget kap paraméterül. A függvény feladata, hogy a paraméterként kapott szöveg alapján számolja ki, hogy mennyi bor és milyen típusú borokat vásároltak az emberek a Borfesztiválon, valamint hogy mennyi volt az összérték.

A szöveg tartalmazza a vásárlók neveit, amelyeket `;` karakterrel választanak el. A nevek vegyesen tartalmazhatnak kis- és nagybetűket (akár lehetnek felesleges szóközők). A függvénynek a következő borokat kell számolnia:

- Vörösbor: Gábor minden vásárlás alkalmával 12 üveg vörös bort vett.
- Fehérbor: Marci minden vásárlás alkalmával 6 üveg fehér bort vett, míg István 8 üveggel, de minden 5. fehérbor ingyenes volt az este alatt.
- Szóda: Gábor minden vásárlás alkalmával 1 üveg szódát vett, míg István minden vásárlás alkalmával 2 üveg szódát vett.

Az borok árai a következők:

Bor	Ár
Vörösbor	4500
Fehérbor	3000
Szóda	350

A függvény számolja ki az összes vásárolt bor mennyiségét típusonként lebontva, és számolja ki az összértéket. Az eredményt egy dictionary-ben adja vissza, ahol a kulcsok a bor típusok (Vörösbor, Fehérbor, Szóda), az értékek pedig a vásárolt mennyiségek.

Amennyiben valamelyik paraméter nem szöveg vagy a szöveg hossza 0, adjunk vissza `-1`-et a függvényből.

Példa:

```
borfesztival("GÁBOR; MARCI;István;István;Gábor ;  
Marci;Marci;gábor;Gábor;ISTVÁN")
```

Eredmény:

```
{  
  'Vörösbor': 48,  
  'Fehérbor': 42,  
  'Szóda': 10,  
  'Összérték': 321500  
}
```

Harci Valorant Ügynökök (30 pont)

A Valorant világában a harci ügynököknek mindig magas támadókéességük van, és különböző képességeikkel rendelkeznek.

Készíts egy `ValorantUgynok` nevű osztályt, amelynek legyen 4 adattagja: `nev`, `_tamadas`, `hp` (életerő) és egy lista, `kepessegek` néven.

Az osztály konstruktorában a nevet és támadást kapja paraméterül (ilyen sorrendben), és ezekkel inicializálja a `nev`, `_tamadas` adattagokat. A `tamadas`-t nem kötelező megadni, alapértelmezett értéke legyen 10. A `kepessegek` adattag egy üres lista. A `hp` kezdetben legyen 100.

Készíts get és set property-t az `_tamadas` adattaghoz, `tamadas` néven. A getter adja vissza az adattag értékét, a setter pedig állítsa be az adattag értékét a paraméterben kapott értékre, de figyeljen, hogy a támadás mindig pozitív és maximum 100 lehet.

Írj egy `kepesség_hozzaadas` metódust, amely egy képesség nevét várja paraméterül (string). A metódus a következőképpen működik:

- Ellenőrizze, hogy a képesség neve string típusú-e. Ha nem, dobjon egy `ValueError` típusú kivételt a "Érvénytelen képesség" hibaüzenettel.
- Amennyiben a paraméter string, adjuk hozzá a `kepessegek` listához, de mivel az ügynökeink neve titkos így csak minden második betűt tároljuk el.

Írj `kepesség_torles` metódust mely a hozzáadáshoz hasonlóan működik, csak nem töröl hanem eltávolítja a képességek közül a paraméterben érkező ügynök nevét.

Definiálj egy `__lt__` metódust, amely meghatározza, hogy az aktuális ValorantUgynok támadóképessége kisebb-e, mint a paraméterként érkező ValorantUgynok támadóképessége. Ha a paraméterben kapott objektum nem ValorantUgynok típusú, dobjunk egy `ValueError` kivételt `Nem ugynok` szöveggel.

Definiáld felül a szöveggé alakítást megvalósító függvényt (`__str__`), úgy, hogy az osztály adatait a következő formában adja vissza: "Az ügynök neve: {nev}, támadása: {tamadas}, HP: {hp}, és {k_db} képessége van.", ahol `{nev}`, `{tamadas}`, `{hp}` és `{k_db}` a megfelelő adatokat helyettesíti be.

Definiálj egy `__add__` metódust, amely két ValorantUgynok objektumot kap paraméterként, és egy új ValorantUgynok objektummal tér vissza, amely, megtartja a magasabb támadóértéket és életerőt, emellett összegzi a képességeiket is. Ha a paraméterben kapott objektum nem ValorantUgynok típusú, dobjunk egy `TypeError` kivételt `Nem ügynököt adtál meg!` szöveggel.

Jó munkát!