Tartalomjegyzék

- 1 "http://wiki.math.bme.huNagy"http://wiki.math.bme.hu Házi feladat
- 2 Esemény
  - 2.1 Tesztelés
  - 2.2 Beküldés
  - 2.3 Feladat
    - 2.3.1 Raeres osztaly [1 pont]
      - 2.3.1.1 konstruktor
        - 2.3.1.1.1 Segítség
      - 2.3.1.2 megis
      - 2.3.1.3 megse
    - 2.3.2 legjobb_datum függvény [1 pont]
      - 2.3.2.1 Segítség
    - 2.3.3 legjobb_fajlbol függvény [1 pont]
      - 2.3.3.1 Segítség
    - 2.3.4 Különálló futtatás [1 pont]
      - 2.3.4.1 Segítség
    - 2.3.5 Dokumentáció [1 pont]

"http://wiki.math.bme.huNagy"http://wiki.math.bme.hu Házi feladat

Határid? (mindenkinek): Április 7. 23:59

Külön fájlba kell írni, és emailben elküldeni nekem. Bizonyos részei CloudCoder-ben is fenn lesznek, de mivel olyan részt is tartalmaz, ami ott nem ellen?rizhet? (megfelel? modul forma, fájl írás/olvasás), nem minden.

8. heti labor (Márc. 30. és Ápr. 1. alkalmak) is ennek lesznek szánva, szval ha valakinek nincs python a gépén, és nem akar feltenni, az is opció ha már ott elkészül vele.

Esemény

Egy olyan programot kell írnunk, ami egy esemény megszervezésében segít, kicsit hasonlóan a közismert http://doodle.com/ honlaphoz. Minden ismer?sünk megadhatja azokat a napokat amik megfelel?ek neki, és a program ez alapján kiválasztja a legjobb dátumot.

Tesztelés

A feladatot az esemeny.py nev? fájlban kell megoldani. A teszteléshez összeállítottam egy csomagot, ami a feladat megoldását az "http://wiki.math.bme.huesemeny"http://wiki.math.bme.hu nev? modulban keresi, így csak ez a fájlnév m?ködik. A tesztel? csomag itt található. A tömörített fájl tartalmát csomagoljátok ki egy könyvtárba, és ott hozzátok létre az esemeny.py fájlt is. Ezután tesztelést a esemeny_teszt.py nev? fájl futtatásával tudtok végezni. Windows-on, ha fel van telepítve a python, akkor ehhez elég csak duplán klikkelni a fájlra. Linux-on ehhez futtathatóvá kell tenni el?bb a fájlt. Mindkét operációs rendszeren m?ködik ha spyder-ben megnyitjuk a fájlt és onnan futtatjuk, vagy ha a megfelel? könyvtárban azt írjuk a parancssorba hogy:

```
python esemeny_teszt.py
```

A tesztel? program a leteszteli a modul m?ködését, és olyan hibák esetén amikre gondoltam, specifikus hibaüzeneteket is ad. Ha valamikor olyan hiba történik benne, amit nem értetek, akkor szóljatok/írjatok emailt.

Beküldés

Miel?tt elkülditek a feladatot, nevezzétek át a fájlt "http://wiki.math.bme.huesemeny_<CC_user>.py"http://wiki.math.bme.hu-re, amibe a saját CloudCoder felhasználóneveteket írjátok be. Tehát pl. aki "http://wiki.math.bme.hutestuser"http://wiki.math.bme.hu a CloudCoder-en, az "http://wiki.math.bme.huesemeny_testuser.py"http://wiki.math.bme.hu fájlnévvel küldje el a megoldását. Ezután küldjétek el az így átnevezett fájlt az email címemre, "http://wiki.math.bme.huInfo2 HF"http://wiki.math.bme.hu tárggyal.

Feladat

A feladat különböz? részeihez ebben az esetben megadom a pontértéket is a végs? 5 pontból.

Raeres osztaly [1 pont]

Legyen a modulban egy **Raeres** nevu osztaly, ami annak tárolására van, hogy egy ismer?sünk melyik napokon ér rá. Az osztálynak két tagváltozója legyen:

- **nev**, ami az ismer?sünk neve egy karakterláncként
- **datumok**, ami egy lista datetime.date objektumokkal, amik id?rendi sorrendben vannak letárolva.

konstruktor

A konstruktor is két paramétert kérjen ennek megfelel?en. Az els? a név, a második legyen a dátumokat tartalmazó lista, és ezt akkor is id?rendi sorrendbe tárolja le a *datumok* tagváltozóba, ha a paramétert nem úgy kapja!

Az osztálynak a konstruktoron kívül két metódusa legyen:

Segítség

- Ha *datetime.date* típusú objektumokat összehasonlítunk (pl. kacsacs?rök használatával), akkor id?rend szerint mondja kisebbnek/nagyobbnak ?ket. Így a különféle beépített *sort()* és *sorted()* függvények is id?rendbe rendezik maguktól sorba ?ket.

megis

A **megis()** metódus egy paramétert kérjen, ami egy dátum, amikor mégis ráér az adott ismer?s. Ez változtassa meg az objektum *datumok* tagváltozóját, hogy ez a dátum is szerepeljen a listában az eddigiek mellett. Fontos, hogy a lista továbbra is legyen id?rendi sorrendbe rendezve.

megse

A **megse()** metódus is egy paramétert kapjon, ami egy olyan dátum lesz, ami szerepel a dátumok jelenlegi listáján. Ezen a dátumon mégse ér rá az adott ismer?s, úgyhogy töröljük a listából. Továbbra is legyen a *datumok* lista id?rendi sorrendbe rendezve.

legjobb_datum függvény [1 pont]

Legyen a modulban, de a Raeres osztályon kívül, egy "http://wiki.math.bme.hulegjobb_datum"http://wiki.math.bme.hu nev? függvény. A függvénynek egy paramétere legyen:

- **raeresek**, egy lista *Raeres* objektumokkal.

A függvény keresse meg a legjobb dátumot az alapján hogy ki mikor ér rá. A legjobb dátum a következ?képpen van meghatározva:

1. El?ször is vegyük azt a dátumot, ami a legtöbb embernek jó. Ha több dátum is ugyanannyi embernek jó, akkor ezek közül a dátumok közül, amik a legtöbb embernek jók, válogatunk a továbbiakban.
2. Ha a legtöbb embernek jó dátumok között van hétvégi, akkor mindenképpen hétvégit választunk.
3. Az így még felmerül? dátumok közül a legkorábbit választjuk.

A függvény térjen vissza az így megtalált legjobb dátummal.

Segítség

- A *datetime.date* osztálynak van weekday() metódusa, ami segíthet megállapítani hogy melyik nap hétvége.
- Opcionálisan a *collections* beépített könyvtár Counter osztálya hasznos lehet megszámolni hogy melyik dátum hányszor szerepel összesen, kicsit egyszer?bbé teheti a kódot. Azonban hátránya, hogy a CloudCoder-en lev? változatával a python-nak nem m?ködik, úgyhogy ha ezt használjátok, akkor csak a saját gépen lehet tesztelni, a CloudCoder-en nem.

legjobb_fajlbol függvény [1 pont]

Ez a függvény ugyanazt csinálja mint az el?z?, csak a különböz? emberek ráérését nem függvényparaméterként kapja, hanem egy fájlból olvassa be. A **legjobb_fajlbol** függvénynek egy paramétere legyen:

- **fajlnev**, a fájl neve, amib?l olvassa be az adatokat.

A fájloknak olyan lesz a formátuma, hogy minden sor egyvalakinek a ráérését írja le, ilyen formátumban:

```
Eszter: 2015.2.4, 2015.2.3
```

Tehát el?ször is egy kett?spont elválasztja a nevet a dátumok listájától, majd a dátumok listáján belül vessz?k választják el a dátumokat, majd a dátumon belül pontok választják el az évet, hónapot és napot.

A függvény a *legjobb_datum()*-hoz hasonlóan a megtalált dátummal térjen vissza, *datetime.date* objektum formában.

Segítség

- Miután beolvastuk a fájlból az adatokat, elég csak meghívni a *legjobb_datum()* függvényt az adatokkal, nem kell átmásolni a kódját vagy ilyesmi.
- Fájl olvasásról az el?z? el?adáson volt szó.
- A tesztel? csomagban láthattok példát arra hogy a fájlok hogy néznek ki, ha megnézitek a *t1.txt*, *t2.txt* vagy *t3.txt* nev? fájlokat.

- Az egy sor adat megfelel? feldarabolásához a karakterlánc split() metódusát lehet használni. Pl. ha az egész sor a *sor* nev? változóban van, akkor

```
sor.split("http://wiki.math.bme.hu:"http://wiki.math.bme.hu)
```

egy két elem? listát ad vissza, amib?l az els? a név, a második a dátumok teljes listája. Hasonlóan aztán a dátumok listáját a vessz?knél szét lehet vágni, majd a dátumot magát a pontoknál.

Különálló futtatás [1 pont]

Azt szeretnénk, ha az esemeny.py fájl akkor is csinálna valami hasznosat, ha magában futtatjuk. A 3. el?adáson volt róla szó, hogy a kódot amit csak akkor szeretnénk futtatni ha ez a fájl fut (és nem importálva van), azt egy

```
if __name__ == "__main__":
```

blokkon belülre kell tenni.

Ennél a python programnál a következ?t szeretnénk ha csinálná ilyenkor a fájl:

- Olvassa be a "http://wiki.math.bme.huraeresek.txt"http://wiki.math.bme.hu fájlból az adatokat
- Keresse meg ez alapján a legjobb dátumot
- A megtalált legjobb dátumot írja ki a "http://wiki.math.bme.hulegjobb_datum.txt"http://wiki.math.bme.hu nev? fájlba. A dátumot úgy formázza, ahogy a *datetime.date* osztály alapértelmezett formázása van, tehát ilyen formátumú legyen a "http://wiki.math.bme.hulegjobb_datum.txt"http://wiki.math.bme.hu tartalma:

```
2015-06-14
```

Segítség

- Fájl írásról is el?z? el?adáson volt szó.
- A dátum szöveggé konvertálásához az str() beépített függvényt használhatjuk, ez mindjárt a megfelel? formátumot fogja adni.
- Természetesen a korábban megírt *legjobb_fajlbol()* függvény használata jelent?sen segít itt.

Dokumentáció [1 pont]

A python-ban, mint már 3. el?adáson tárgyaltuk, a docstringek használata a szabványos megoldás a dokumentációra. Legyen a megoldásunkban minden fontosabb dolognak legalább egy soros docstring-je, tehát:

- Magának a *Raeres* osztálynak
- A *Raeres* osztály *megis()* és *megse()* metódusának (a konstruktornak nem kell)
- A *legjobb_datum()* és a *legjobb_fajlbol()* függvényeknek.

(Ha valaki ezek közül valamelyeket nem oldotta meg, elég ha minden dokumentálva van amit megoldott.)