Rádió

Készítette: Veres Z. Benedek

Selye János Gimnázium, Komárom

InfoProg 2021 levelező forduló, programozói B kategória

Tartalom:

Főbb modulok:

- Universal_tools.py
- Playlist.py
- Management.py
- Gui.py

universal_tools.py

-Ebben a modulban összefoglaltam a főbb, funkciókat, amelyeket, szinte az összes egyéb modulban használok.

Funkciók:

- Ellenörző:
  - check_file()
  - check_if_length_right()
- Kereső:
  - show_header()
  - return_column()
  - search_for()
- Fájl szerkesztő:
  - write_full_csv()
  - add_row()
  - edit_row()
  - delete_row()

playlist.py

-Ez a modul felelős azért, hogy a zenékből lejátszási listákat lehessen készíteni.

Klasszok és metódusai:

- Playlist()
  - random_music()

management.py

-Ebben a modulban van a fő klassz, ami a gerince a programnak

Klasszok:

- RadioManager
- EmptyObject

gui.py

- A modul foglalkozik a program vizuális részével.

Főbb funkciói:

- add_interface()
- validate()
- print_row()
- make_box()

universal_tools.py

UTF-8 kódolás helyett ISO-8859-2 kódolást használtam, mivel az UTF-8 nem ismerte fel az **ő** és **ű** betűket

check_file(file,tag)

parametéterek:

- file – a fájl neve, amit ellenőrizni szeretnénk, ha nem létezik akkor készít egyet specifikus fejléccel.
- tags – ha fájlt készít, milyen címkék legyenek a fejlécen.

-Nem ad vissza semmilyen értéket, csak ha a fájl nem létezik, akkor egy specifikus fejléccel(amit megadtunk a tags parameternél) készít egyet.

check_if_length_right(string,max_length)

paraméterek:

- string – az a string típusú adat, aminek megszeretnék nézni a hosszát.
- max_length – mennyi lehet a maximum hossza a stringnek
- A funkció vagy True-t vagy False-t add vissza értékként, attól függően, hogy len(string) <= int(max_length).

show_header(file)

paraméterek:

- file – fájl, aminek ki fogja írni, a fejlécét.

return_column(file,column)

paraméterek:

- file – fájl, amelynek vissza fogja adni az egyik oszlopát.
- column – ez egy string, ami lehetőleg az egyik címke a fejlécek között.
- A funkció visszaad egy listát, amiben megtalálható a specifikus oszlop elemeit. Ha a címkét nem találja, vagy az oszlop üres, üres listát ad vissza.

search_for(file,search_tag,specify)

paraméterek:

- file - az a fájl, amelyet végig keres
- search_tag – mit keressen
- specify – alapból egy üres string, ilyenkor az egész fájlt átkeresi, ha valamit beírsz akkor abban a specifikus oszlopban fog keresni.
- Visszaadja azokat a sorokat, amelyekben a search_tag, megtalálható. Lista formában adja vissza.

write_full_csv(file,content)

paraméterek:

- file – a neve az új fájlnak, amelyet létrehoz.
- content – mit írjon a sorba. Figyelem: a content egy lista típus kér be, és a sorban a listában lévő elemeket, mindig új sorba fogja írni.

Példa:

```
.>data = [['name','id', interest'],['Bob Ross','1','Painting'],['Isaac Asimov','2','writing']
.> write_full_csv(file,data)
.>'name','id','interest'
.'Bob Ross','1','Painting'
.'Isaac Asimov','2','writing'
```

add_row(file,content)

paraméterek:

- file – fájl, amelybe írni fogja az új sort.
- content – ez egy lista, és a listában lévő elemeket vesszővel elválasztva fogja beírni.

-Mindig az utolsó sorba ír

edit_row(file,id,tag,content)

paraméterek:

- file – fájl, amelyben azt a specifikus sort fogja átírni
- id – a sornak az idja (a programban szükséges, hogy a soroknak idjük legyen), amit át fog majd írni
- tag – megadja, hogy melyik címke alatt írja át.
- content – mire írja át

-Van pár dolog, amit a fájl éppsége és átláthatósága érdekében, amit nem lehet át írni:

1. A 0.ID, ami a fejléc nem lehet átírni,
2. Az ID-kat sem lehet átírni, így minden ID egy adott fájlban egyedi

delete_row(file,id)

paraméterek:

- file – megadja, hogy melyik fájlban töröljük ki az egyik sort.
- id – megadja, hogy melyik idjűt sort törölje ki
- Nem lehet kitörölni a fejléce, azaz a 0 ID-jű elemet.

playlist.py

Class – Playlist()

- az objektum elkészítésekor meg kell adni, hogy melyik fájból, szedje ki a zeneszámokat és később dolgozhasson vele. (2. feladat)

Metódusai:

random_music()

- kimutatja a fájlban talált zenék címkéit, mint választható lehetőség, utána kiválaszthatjuk, hogy mennyi zenét választjon ki véletlen szerűen(maximum annyi, amennyi abból a címkéből van a fájlban), és rakja bele egy új fájlba, aminek a neve [címke]-[zenék száma]-[év-hónap-nap-óra-perc] jelölést követi.

management.py

- Ez a modul foglalkozik, hogy az összes modult összekovácsolja. Itt alakítjuk ki a megfelelő objektumokat.

Class – RadioManager(name,file,rules,tags,linked_file,linked_file_tags)

- Ez a klassz rakja össze az objektumokat a program működéséhez (műsorvezetők,műsorok,műsor idők)
- paraméterek:
  - name – megadjuk hogy majd az objektumnak mi a neve, valamilyen szinten ez egy azonosítóként működik
  - file – megadja az objektum fájljának helyét
  - rules – az objektum kritériumai, amiket majd tudunk ellenőrizni (pl. a név az maximum 30 karakterből állhat)
  - tags – a fájl fejlécének címkéi (pl. id,név,leírás,…)
  - linked_file – egy blockláncot tudunk ezzel létre hozni (például a műsorok objektum össze van kötve a műsorvezetők objektumával, hogy fel tudja ismerni milyen műsorvezetők vannak)
  - linked_file_tags – a hozzárendel fájl, címkéi

- EmptyObject():
  - Üres objektum

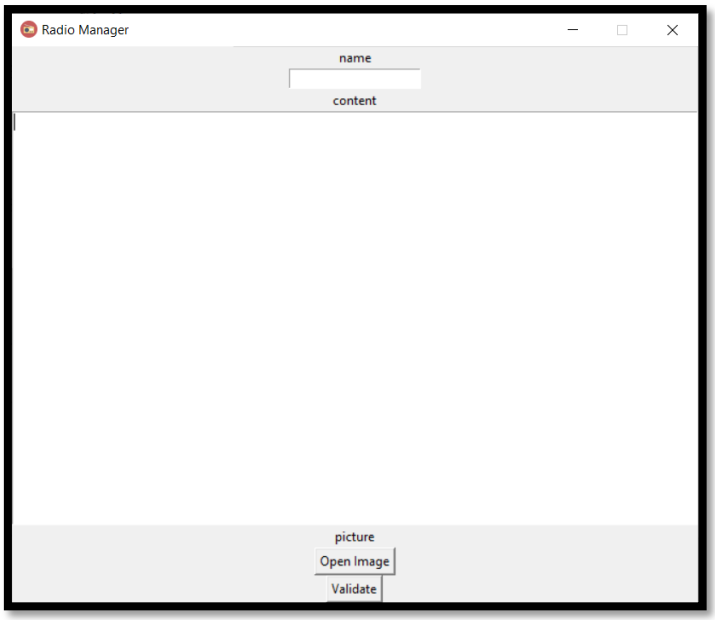
gui.py

A program elindításakor kinyílik a kezelőfelület (FIGYELEM: a programot a saját virtuális környezetében kell elindítani, mivel a kellet egy külső modul (Pillow) amely segített a képekkel való munkálatokban, ha a számítógépen megtalálható a Pillow modul, akkor önmagában is elindítható a program) A program felülete:

add_interface(event,object,mode='add')

paraméterek:

- Event – ez csak ahhoz szükséges, hogy egy gombhoz köthessük a funkciót
- Object – megnevezzük, hogy melyik objektumhoz van köze a funkciónak
- Mode – „add" vagy „edit" módja van a funkciónak, „add" módban a program egy új sort fog hozzá adni a fájlhoz, „edit" módban pedig már egy létező sort írhatunk át.



Az object-nek van egy kritériumai (object.rules), amely meghatározza, hogy a felületen milyen bemenetelek szereplejenek(a képen, a név, leírás és a kép helye szerepelnek)

validate(event,object,mode)

paraméterek:

- Event – ez csak ahhoz szükséges, hogy egy gombhoz köthessük a funkciót
- Object – megadja, hogy melyik objektumohz van köze

- Mode – hasonlóan az add_interfacehez, itt is két módja van „add” és „edit”

Az add interface-ben megadott dolgokat ellenőrzi a benne lévő szűrők alapján (fő szűrője, amely ellenőrzi az object.rules-ban található kritériumokat)

Validate_time(start,end)

paraméterek:

- Start – a kezdő időpont
- End – a végső időpont

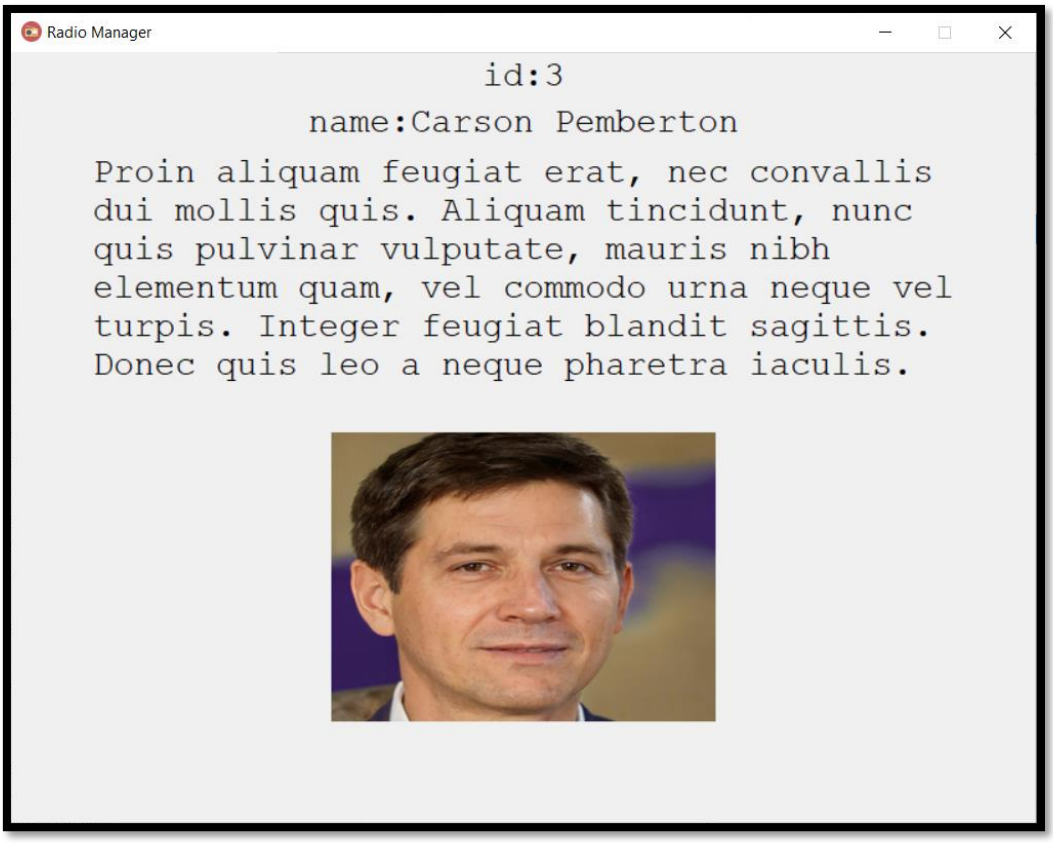
A funkció megnézi, hogy az idő megfelelő formában van-e (óra:perc)

print_row(event,obj)

paraméterek:

- Event – ez csak ahhoz szükséges, hogy egy gombhoz köthessük a funkciót
- Obj – megnevezzük, hogy melyik funkciónak van köze a programhoz

Ez a funkció a make_box funkcióval működik együtt (csak azután működik, hogy a make_box segítségével csináltunk egy global box-ot), ha létezik a doboz (amiben van egy Listbox), megnézi mire kattintottunk és az elmenti a chosen változóban. Ezután megnézi annak az objektum fájljának az elemeit és kiírja egy új ablakban. Ha a képet nem találja akkor lecseréli az InfoProg logójára.



Make_box(event,obj)

paraméterek:

- Event - ez csak ahhoz szükséges, hogy egy gombhoz köthessük a funkciót
- Obj – melyik objektumhoz van köze, alapból a CHOSEN_OBJECT globális változó az obj. Amikor megnyomjuk a gombot átírja a CHOSEN_OBJECT-ot

A funkció egy megadott Frame-be egy ListBoxot készít el.