Pràctica 3: Gestió de la Xarxa Social iTICApp: Ampliació

Tecnologia de la Programació — iTIC

Marta I. Tarrés Puertas

February 5, 2021

Contents

1 Organització

1.1 Objectius

L'objectiu d'aquesta pràctica és prendre contacte amb els programes estructurats en base a classes d'objectes. El context del projecte se sitúa en una xarxa social, en el marc d'un projecte de recerca sobre els seus usuaris i els posts que publiquen a la xarxa.

Després d'haver lliurat la pràctica 2, en que es construïa una aplicació per a la gestió d'usuaris i posts, en aquesta ampliació ens proposem els següents objectius:

- Millorar lleugerament l'intèrpret d'ordres.
- Afegir ordres per desar les dades en un fitxer i recuperar-les més endavant.
- Simplificar la lectura i gravació de dades.

Les tasques cal fer-les durant aquesta sessió de laboratori i no haurien de suposar més temps.

Aquesta pràctica també us la podeu prendre com una auto-avaluació. Hauríeu de veure-us capaços de poder-la fer individualment si heu treballat convenientment la pràctica 2.

1.2 Condicions

- La pràctica cal fer-la en equips de dues persones.
- El model de desenvolupament que es demana que utilitzeu és test driven programming. Si no en recordeu els detalls del curs passat, pregunteu!

1.3 Lliuraments

- Caldrà lliurar el codi resultant del projecte la documentació escrita amb `Sphinx` que ha d'incloure també una taula de dedicacions de cada persona del grup a les diferents tasques de la pràctica.
- La durada de la pràctica és d'una setmana.

2 Introducció

Per desenvolupar la pràctica es requereixen coneixements previs sobre:

- Fitxers de text.
- Cadenes de caràcters.

Seria interessant fer-hi un cop d'ull abans de venir a la sessió de laboratori.

> TASCA PRÈVIA 1 Repasseu-vos el tractament de fitxers de text i de cadenes de caràcters de l'assignatura d'INF.

3 Millora de l'intèrpret

L'objectiu d'aquesta part és fer l'intèrpret d'ordres una mica més interessant.

TASCA 2 Afegiu una nova ordre fixa a totes les instàncies de la classe Interpret modificant la classe convenientment. Aquesta nova ordre és:

ajuda Escriu per la pantalla la llista de totes les ordres conegudes per l'interpret en ordre alfabètic.

A tal efecte heu de modificar convenientment el mètode run de la classe Interpret.

TASCA 3 Finalment cal millorar el tractament dels errors. Quan l'usuari s'equivoca escrivint una ordre, en comptes d'avortar el programa amb una excepció cal escriure un missatge indicant quin ha estat el problema i continuar com si res hagués passat.

Per exemple, si l'usuari escriu:

```
** print posts-user
```

i oblida dir quin nick d'usuari, l'aplicació hauria de dir quelcom semblant a:

```
** print posts-user
error: manca el nick de l'usuari.
```

fins i tot, si posem un nom d'una ordre no existent, hauríem d'observar un comportament com el següent:

```
** print posts i hashtags
error: comanda desconeguda.
```

Això és fàcil d'aconseguir millorant convenientment les funcions associades a cada ordre, de forma que comprovin els paràmetres que reben i actuïn en conseqüència.

4 Gravació i lectura en fitxer

L'objectiu d'aquesta tasca és dotar l'aplicació de dues comades més. Una per desar en un fitxer de text les dades i una altra per llegir-les. Les noves ordres han de ser:

desa ⟨`nomf`⟩ Desa les dades en el fitxer de text de nom ⟨`nomf`⟩.

recupera ⟨`nomf`⟩ Recupera les dades del fitxer de text de nom ⟨`nomf`⟩. En cas que el fitxer contingui productes o receptes que ja existien, no les incorpora de nou i les ignora.

A tal efecte seguiu els següent esquema de treball:

Tasca 4 Afegiu a la classe iTICApp dos nous mètodes. Aquests mètodes han de permetre gravar i recuperar un la xarxa social des d'un fitxer de text. Els mètodes han de ser els següents:

```
def desa(self, f):
    """
    Desa la xarxa 'self' en el fitxer de text 'f'. El fitxer
    'f' ha d'estar obert en mode escriptura.
    """
    pass

def obre(self, f):
    """
    Afegeix a la xarxa 'self' els usuaris i posts que hi
    ha en el fitxer de text 'f'. El fitxer 'f' ha d'estar obert
    en mode lectura.
    """
    pass
```

Ara cal afegir les ordres desa i recupera al nostre intèrpret d'ordres modificant convenientment el mòdul `main.py`.

Si tot va bé, ara podeu emmagatzemar la xarxa i tornar-la a recuperar per continuar treballant-hi. Fantàstic!

5 Millora de la gestió de la gravació

Tal i com ha quedat l'aplicació després de la tasca anterior, a l'usuari li cal gestionar la gravació/recuperació de dades des del fitxer. Això és una mica inconvenient per l'usuari corrent. En aquesta tasca modificarem l'aplicació per tal que:

- Carregui automàticament el fitxer de dades en començar la sessió.

- Es gravi automàticament al sortir de la sessió.
- Hi hagi una ordre per gravar la informació a petició de l'usuari.

A tal efecte seguirem el següent guió:

TASCA 5 Cal modificar la classe Interpret afegint una nova funcionalitat. Tota instància d'aquesta classe haurà de conèixer dues funcions especials que s'invocaran, si és el cas, a l'engegar l'interpret i al finalitzar (al principi i al final de run).

A tal efecte, afegir dos nous atributs a la classe:

alpha És una funció sense paràmetres que es cridada pel mètode run just abans de començar.

omega És una funció sense paràmetres que es cridada pel mètode run just abans d'acabar.

També afegirem dos mètodes més a la classe que serviran per gestionar aquests atributs:

```
def set_begin(self, f):
    """
    Fixa la funció 'f' com l'inicialitzador que es cridarà just
    abans d'arrencar l'interpret. 'f' és una funció sense
    paràmetres.
    """
    pass

def set_end(self, f):
    """
    Fixa la funció 'f' com el finalitzador que es cridarà just
    abans d'acabar l'execució de l'intèrpret. 'f' és una funció sense
    paràmetres.
    """
    pass
```

TASCA 6 Usant les funcionalitats implementades en el punt anterior, feu que el programa carregui les dades d'un fitxer de nom fix, per exemple `dades.dat` i les desi en el mateix fitxer just abans d'acabar l'execució. Haureu de modificar únicament el mòdul `main.py`.

TASCA 7 Substituïu les ordres afegides en la tasca 2 per una única ordre:

desa Desa les dades de la xarxa en el fitxer.