
Fonaments de Programació

Pràctica 2: Hotels (Part 1)

Notes importants:

1. Si no us dona temps d'acabar tots els exercicis durant la sessió de pràctiques **els heu d'acabar després** pel vostre compte.
2. **S'ha d'utilitzar:**
 - 2.1. **El mateix nom que us proposem per a les classes, funcions i procediments**
 - 2.2. **Els mateixos missatges**

Objectiu:

L'objectiu d'aquesta pràctica és implementar un petit gestor de dades dels hotels d'una ciutat. L'aplicació final haurà de permetre als usuaris poder obtenir diferent informació sobre els hotels de la ciutat, la seva distribució per barris, la seva localització, etc.

Les dades dels hotels, així com les dades dels barris i districtes de la ciutat, s'obtidran a partir d'uns fitxers de dades CSV amb un format determinat. A cada línia dels fitxers hi ha les dades d'un hotel, barri o districte. Per cada línia, les dades disponibles estan separades pel caràcter ';' i tenen un salt de línia ('\n') al final.

Per aquesta pràctica us donem tres fitxers CSV amb les dades dels hotels, barris i districtes de la ciutat de Barcelona. Abans de començar amb la pràctica, reviseu els fitxers per tenir clar el seu format i les dades que contenen.

Per implementar el gestor plantejat, se us proposen uns exercicis on es demanen les classes, funcions i procediments necessaris i que posteriorment s'han d'utilitzar per a fer el programa principal.

Exercici 1: Crear una classe que s'anomeni `Hotel` per guardar la informació d'un hotel. Els atributs d'aquesta classe seran: `nom`, `codi_hotel`, `carrer`, `numero`, `codi_barri`, `codi_postal`, `telefon`, `latitud`, `longitud` i `estrelles`. La classe haurà de tenir els següents mètodes:

- `__init__(self, nom, codi_hotel, carrer, numero, codi_barri, codi_postal, telefon, latitud, longitud, estrelles)`: Constructor per inicialitzar els atributs d'un objecte de la classe `Hotel`.

Aquest mètode haurà de fer algunes comprovacions sobre els arguments passats per alguns dels atributs:

- Els valors passats pel número, el codi de barri i el número d'estrelles han de ser enters (int) positius. En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "<camp> ha de ser un valor enter positiu" on <camp> serà el nom del camp (`numero`, `codi_barri` o `estrelles`) pel qual s'ha passat un valor erroni.
- Els valors passats per la latitud i la longitud han de ser reals (float). En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "<camp> ha de ser un valor real" on <camp> serà el nom del camp (`latitud` o `longitud`) pel qual s'ha passat un valor no real.
- El valor passat pel número d'estrelles ha d'estar entre 1 i 5 (ambdós inclosos). En cas contrari, s'ha de generar una excepció de tipus `ValueError` amb el missatge "estrelles ha de ser un valor entre 1 i 5".

- `__str__(self)`: Retornarà les dades d'un hotel com una cadena de caràcters en una línia amb el format:

```
<nom> (<codi_hotel> <carrer>,<numero> <codi_postal> (barri: <codi_barri>)
<telèfon> (<latitud>,<longitud>) <estrelles> estrelles
```

Exemple:

```
Hotel H10 Itaca (HB-004151) Roma,22 08015 (barri: 9) 932265594
(41.381193,2.145467) 4 estrelles
```

- `__gt__(self, altre_hotel)`: Permetrà utilitzar l'operador `>` entre dos hotels retornant `True` si l'hotel té més estrelles que `altre_hotel` i `False` en cas contrari.
- `distancia(self, latitud, longitud)`: Calcularà i retornarà la distància entre l'hotel i les coordenades passades com a paràmetre segons la fórmula:

$$dist = \arccos(\sin(lat_1) \cdot \sin(lat_2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \cos(lon_2 - lon_1)) \cdot RADI_TERRA$$
on `RADI_TERRA` és igual a 6378.137, i les latituds i longituds s'han de convertir a radians multiplicant per $\pi/180$ abans d'aplicar la fórmula.

- A la llibreria `math` disposeu de la constant `pi` i de les funcions trigonomètriques.
- Els valors passats per la latitud i la longitud han de ser reals (`float`). En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "`<par>` ha de ser un valor real" on `<par>` serà el nom del paràmetre (`latitud` o `longitud`) pel qual s'ha passat un valor no real.

Exercici 2: Fer la funció `codi_in_llista_hotels` que rebrà com a paràmetres una llista amb els hotels d'una ciutat (un objecte de la classe `Hotel` a cada posició) i una cadena de caràcters amb el codi d'un hotel. La funció retornarà `True` si la llista conté un hotel amb el codi passat com a paràmetre o `False` en cas contrari.

Exercici 3: Fer un funció anomenada `importar_hotels` que rebi com a paràmetre el nom d'un fitxer (`string`) amb la informació dels hotels d'una ciutat i un caràcter separador (per exemple `','`), i retorni una llista on cada posició de la llista contingui un objecte de la classe `Hotel` amb les dades d'un hotel de la ciutat. En acabar de llegir el fitxer, la funció ha de mostrar el missatge "S'han importat correctament `<N>` hotels" on `<N>` és el nombre d'hotels que s'han guardat correctament a la llista d'hotels.

Tingueu en compte els següents punts:

- S'ha de controlar que el fitxer es pot obrir. Si es produeix un error, s'ha de generar una excepció de tipus `FileNotFoundError` amb el missatge "fitxer no trobat".
- Si teniu algun problema amb la codificació dels caràcters al llegir el fitxer, obrir-lo amb:

```
open('nom_fitxer', 'r', encoding='utf-8')
```
- La primera fila del fitxer conté l'encapçalament de les dades i no s'ha de tenir en compte ja que no conté dades de cap hotel.
- Al final de cada línia hi ha un salt de línia (`"\n"`). Elimineu-lo abans de processar les dades de la línia.
- Al fitxer, el codi de l'hotel està junt amb el nom de l'hotel a la primera columna. El codi de l'hotel està format per dos lletres, un guió i 6 números (ex.- HB-004151). El nom de l'hotel i el codi s'han de separar eliminant els caràcters addicionals (`' - '`) que hi ha entre ells.
- El fitxer pot contenir hotels repetits i cal controlar que no es creïn entrades repetides a la llista. per a fer-ho, utilitzeu la funció `codi_in_llista_hotels`.
- Al crear els objectes de la classe `Hotel` amb el seu constructor, heu de passar les dades de l'hotel en el tipus (`'str'`, `'int'`, `'float'` ...) que correspongui.

- Al fitxer, els valors de la ubicació dels hotels (latituds i longituds) tenen 6 decimals però estan guardats com un valor sense separació entre la part entera i els decimals. Per obtenir els valors correctes, cal dividir aquests valors per 1.000.000 després de fer les conversions de tipus corresponents.
- No us oblideu de tancar el fitxer a l'acabar de fer la lectura.

Exercici 4: Crear una classe que s'anomeni `Barri` per guardar les dades d'un barri de la ciutat. Els atributs d'aquesta classe seran: `nom` i `codi_districte`. Aquesta classe haurà de tenir els següents mètodes:

- `__init__(self, nom, codi_districte)`: Constructor per inicialitzar els atributs d'un objecte `Barri`.

Aquest mètode haurà de comprovar que el valor passat pel codi de districte sigui un valor enter (int) i positiu. En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "codi_districte ha de ser un valor enter positiu".

- `__str__(self)`: Retornarà les dades d'un barri com una cadena de caràcters en una línia amb el format:

```
<nom> (districte: <codi_districte>)
```

Exemple:

```
la Maternitat i Sant Ramon (districte: 4)
```

Exercici 5: Fer un funció anomenada `importar_barris` que rebi com a paràmetre el nom d'un fitxer (string) amb la informació dels barris d'una ciutat i un caràcter separador (per exemple ';'), i retorni un diccionari on la clau sigui el codi de barri (valor enter) i el valor sigui un objecte de la classe `Barri` amb les dades d'un barri de la ciutat. En acabar de llegir el fitxer, la funció ha de mostrar el missatge "S'han importat correctament <N> barris" on <N> és el nombre de barris que s'han guardat correctament al diccionari de barris.

Tingueu en compte els següents punts:

- S'ha de controlar que el fitxer es pot obrir. Si es produeix un error, s'ha de generar una excepció de tipus `FileNotFoundError` amb el missatge "fitxer no trobat".
- La primera fila del fitxer conté l'encapçalament de les dades i no s'ha de tenir en compte ja que no conté dades de cap barri.
- Al final de cada línia hi ha un salt de línia ('\n'). Elimineu-lo abans de processar les dades de la línia.
- El codi de barri utilitzat com a clau del diccionari de barris ha de ser un valor enter.
- Al crear els objectes de la classe `Barri` amb el seu constructor, heu de passar les dades del barri en el tipus ('str', 'int', 'float' ...) que correspongui.
- No us oblideu de tancar el fitxer a l'acabar de fer la lectura.

Exercici 6: Crear una classe que s'anomeni `Districte` per guardar les dades d'un districte de la ciutat. Els atributs d'aquesta classe seran: `nom`, `extensio`, `poblacio` i `llista_barris`. Aquesta classe haurà de tenir els següents mètodes:

- `__init__(self, nom, extensio, poblacio)`: Constructor per inicialitzar les propietats d'un objecte `Districte`. Els camps `nom`, `extensio` i `poblacio` s'inicialitzaran amb els valors passats com arguments. El camp `llista_barris` s'inicialitzarà amb una llista buida.

Aquest mètode haurà de fer algunes comprovacions sobre els arguments passats per alguns dels atributs:

- El valor passats per la població del districte ha de ser enter (int) i positiu. En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "poblacio ha de ser un valor enter positiu".
- El valor passat per l'extensió del districte ha de ser real (float) i positiu. En cas contrari, s'ha de generar una excepció de tipus `TypeError` amb el missatge "extensio ha de ser un valor real positiu".
- `__str__(self)`: Retornarà les dades d'un districte com una cadena de caràcters en una línia amb el format:

```
<nom> (<extensio> kms2, <poblacio> habitants) barris: <barri_1>, <barri_2>, ..., <barri_n>
```

Si la llista de barris és buida, s'imprimirà N/D per indicar que els barris del districte no estan disponibles.

Exemples:

```
Les Corts (6.02 kms2, 81576 habitants) barris: les Corts, la  
Maternitat i Sant Ramon, Pedralbes
```

```
Eixample (7.46 kms2, 269349 habitants) barris: N/D
```

- `densitat(self)`: Retornarà la densitat de població (habitants per km²) del districte.

Exercici 7: Fer un funció anomenada `importar_districtes` que rebi com a paràmetre el nom d'un fitxer (string) amb la informació dels districtes d'una ciutat i un caràcter separador (per exemple ';'), i retorni un diccionari on la clau sigui el codi de districte (valor enter) i el valor sigui un objecte de la classe `Districte` amb les dades d'un districte de la ciutat. En acabar de llegir el fitxer, la funció ha de mostrar el missatge "S'han importat correctament <N> districtes" on <N> és el nombre de districtes que s'han guardat correctament al diccionari de districtes.

Tingueu en compte els següents punts:

- S'ha de controlar que el fitxer es pot obrir. Si es produeix un error, s'ha de generar una excepció de tipus `FileNotFoundError` amb el missatge "fitxer no trobat".
- La primera fila del fitxer conté l'encapçalament de les dades i no s'ha de tenir en compte ja que no conté dades de cap districte.
- Al final de cada línia hi ha un salt de línia ('\n'). Elimineu-lo abans de processar les dades de la línia.
- El codi de districte utilitzat com a clau del diccionari de districtes ha de ser un valor enter.
- Al crear els objectes de la classe `Districte` amb el seu constructor, heu de passar les dades del districte en el tipus ('str', 'int', 'float' ...) que correspongui.
- No us oblideu de tancar el fitxer a l'acabar de fer la lectura.

Exercici 8: Fer el procediment `omplir_llista_barris` que rep com a paràmetres un diccionari de districtes (clau: codi de districte, valor: objecte de la classe `Districte`) i un diccionari de barris (clau: codi de barri, valor: objecte de la classe `Barri`). El procediment comprovarà que tots els districtes del diccionari de districtes tinguin el camp `llista_barris` buit. Si és així, omplirà el camp `llista_barris` de tots els districtes del diccionari de districtes a partir de la informació del diccionari de barris. En cas contrari, imprimirà per pantalla el missatge "El diccionari de districtes ja conté informació dels barris"

Exercici 9: Fer el procediment `mostrar_hotels` que rep com a paràmetre una llista d'hotels on cada posició conté un objecte de la classe `Hotel`. Aquest procediment ha d'imprimir la informació de tots els hotels de la llista, un a cada línia. Si la llista no conté hotels, s'ha de mostrar el missatge "No hi ha hotels".

Exercici 10: Fer el procediment `mostrar_menu` que imprimirà per pantalla el menú del programa amb una línia en blanc abans del text. De moment, només es mostraran dues opcions:

```
--- MENÚ PRINCIPAL ---  
1 - Veure hotels  
S - Sortir del programa
```

Exercici 11: Fer el programa principal que segueixi els passos que teniu a continuació. **Utilitzeu les funcions i procediments dels exercicis anteriors.**

1. Inicialitzacions de variables i constants.
2. Fer la lectura dels fitxers amb les funcions dels exercicis 2, 4 i 6, per crear la llista amb la informació dels hotels i els diccionaris amb la informació dels districtes i els barris de la ciutat. La crida a les funcions es farà dins una estructura `try-except-else-finally` per capturar i tractar les possibles excepcions que es produeixin en fer la lectura i procés dels fitxers.
 - 2.1. Si es produeix una excepció de tipus `FileNotFoundError`, es mostrarà el missatge "Error llegint fitxers: " i a continuació el missatge retornat per l'excepció.
 - 2.2. Si es produeix una excepció d'un altre tipus, es mostrarà el missatge "Error processant els fitxers: " i a continuació el missatge retornat per l'excepció.
 - 2.3. Si no es produeix cap excepció, s'executaran els passos 3 a 8 que teniu més avall.
 - 2.4. Finalment, tant si es produeix alguna excepció com si no, es mostrarà un missatge amb el següent format: "© <nom autors>" on <nom autors> seran els noms i cognoms dels integrants del grup de pràctiques.
3. Omplir la llista de barris de cada districte de la ciutat que tindreu guardats al diccionari de districtes a partir de la informació continguda al diccionari de barris. Per a fer aquest punt, utilitzeu el procediment `omplir_llista_barris`.
4. Mostrar per pantalla el menú principal del programa (procediment `mostrar_menu`) i demanar a l'usuari que introdueixi una de les opcions del menú.
5. Si l'opció és '1', mostrar la informació dels hotels continguts a la llista d'hotels amb el procediment `mostrar_hotels`.
6. Si l'opció és 'S' o 's', mostrar el missatge "Sortint del programa".
7. Per qualsevol altra opció, es mostrarà el missatge "Opció no permesa".
8. Repetir els passos del 3 al 7 fins que l'opció escollida sigui la de sortir ('S' o 's').