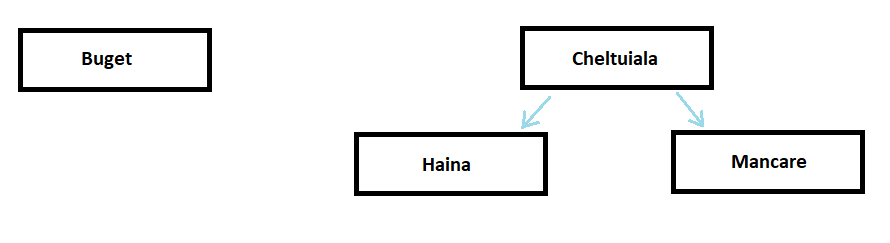
***Python Fundamentals Webex***

Implementati urmatoarea structura de clase.



Se vor implementa in total 4 clase.

1. ***Buget*** -> Clasa de sine statatoare ce va interactiona cu celelalte clase prin metodele ei
   * ***Atribute***:
     1. ***bani\_disponibili*** (int): valoarea bugetului, introdusa in momentul instantierii unui obiect buget
     2. ***cheltuieli*** (Lista de obiecte de tip ***Cheltuiala***, optional): Implicit are valoarea None.
     3. ***bani***\_***cheltuiti*** (int): valoare totala a atributelor suma din lista de obiecte cheltuiala (nu se instantiaza “hardcoded”; trebuie gasita o metoda programatica pentru a face asta)
   * ***Metode***:
     1. \_\_***init***\_\_: Seteaza valorile pentru toate atributele de mai sus
     2. \_\_***repr***\_\_: apelarea acestei functii prin intermediul ***str(obiect\_buget)*** sau prin ***print(obiect\_buget)*** va *returna* un sir de caractere cu valorile atributelor: ***[bani\_cheltuiti, bani\_disponibili, lungimea listei self.cheltuieli]*** intr-un f-string literal usor lizibil (eg: ‘S-au cheltuit 500 de lei. Au ramas 4500 de lei. In total s-au facut 10 achizitii’)
     3. \_\_***len***\_\_: returneaza lungimea listei self.cheltuieli
     4. ***verifica\_bani***(***self, cheltuiala***): verifica daca banii disponibili minus valoarea cheltuielii sunt sau nu > 0. Daca valoare este mai mare decat 0 ***returneaza True***. Altfel ***returneaza False***.
     5. ***adauga\_intrare***(***self, cheltuiala***): daca verifica\_bani returneaza true, modifica toate atributele clasei buget in mod corespunzator.
     6. ***elimina\_intrare***(***self, cheltuiala***): elimina cheltuiala din lista ***self.cheltuieli***.
     7. ***save\_budget***(***self, filepath, filename***): In urma apelarii functiei acesteia se va creea calea filepath si se va genera un fisier in calea respectiva care va contine, pe cate o linie fiecare cheltuiala din obiectul buget curent. (atentie, programul trebuie sa poata rula de 2 ori consecutiv fara nici o modificare (hint: tratarea exceptiilor/verificarea existentei unui path))
2. ***Cheltuiala*** -> clasa parinte
   * ***Atribute***:
     1. ***suma*** (int): costul cheltuielii curente
     2. ***descriere*** (str, optional): implicit ia valoarea ‘’. Este o scurta descriere optionala a cheltuielii.
   * ***Metode***:
     1. \_\_***init***\_\_: instantiaza atributele de mai sus
     2. \_\_***repr***\_\_: apelarea acestei functii prin intermediul ***str(obiect\_cheltuiala)*** sau prin ***print(obiect\_cheltuiala)*** va returna un sir de caractere cu valorile atributelor: [***suma, descriere***] intr-un f-string literal usor lizibil (eg: ‘S-au cheltuit 500 lei. {descriere}’)
3. ***Haina*** -> clasa mostenitoare a clasei Cheltuiala
   * ***Atribute***:
     1. ***suma*** (int): costul hainei
     2. ***descriere*** (str, optional): Implicit ia valoarea “”. Este o scurta descriere optionala a cheltuielii.
     3. ***firma*** (str, optional): Implicit ia valoarea “”.
   * ***Metode***:
     1. \_\_***init***\_\_: instantiaza atributele de mai sus. Se implementeaza folosind ***super***()
     2. \_\_***repr***\_\_: apelarea acestei functii prin intermediul ***str(obiect\_cheltuiala)*** sau prin ***print(obiect\_cheltuiala)*** va returna un sir de caractere cu valorile atributelor: [***suma, descriere, firma***] intr-un f-string literal usor lizibil (eg: HAINE: 100 Bluza Tommy’)
     3. ***returneaza\_produs***(***self, buget***): apeleaza metoda ***elimina\_intrare***() din ***buget*** si elimina ***obiectul cheltuiala*** curent (datorita mostenirii si obiectele de tip Haina se comporta ca cele de tip Cheltuiala) din lista de obiecte cheltuiala din buget.
4. ***Mancare*** -> clasa mostenitoare a clasei Cheltuiala
   * ***Atribute***:
     1. ***suma*** (int): costul hainei
     2. ***descriere*** (str, optional): Implicit ia valoarea “”. Este o scurta descriere optionala a cheltuielii.
     3. ***bio*** (str, optional): Implicit ia valoarea True.
   * Metode:
     1. \_\_***init***\_\_: instantiaza atributele de mai sus. Se implementeaza folosind super()
     2. \_\_***repr***\_\_: apelarea acestei functii prin intermediul ***str(obiect\_cheltuiala)*** sau prin ***print(obiect\_cheltuiala)*** va returna un sir de caractere cu valorile atributelor: [***suma, descriere, bio***] intr-un f-string literal usor lizibil (eg: MANCARE: 5 imi era pofta de un mar. Bio: True)

***Important****:*

*Dupa crearea claselor se cere instantierea unui obiect al clasei Buget, si a cate 2 obiecte ale claselor Haina respectiv Mancare si apelarea tuturor metodelor implementate pentru fiecare tip de obiect.*

Punctajul va fi acordat astfel:

● 10% - crearea claselor

● 60% - implementarea corecta a fiecarei metode

● 10% - crearea instantei acestei clase si apelarea fiecarei metode din clasa cu parametri corespunzatori.

● 10% - prezenta comentariilor, respectarea standardelor de scriere a codului in Python (Pep8)

● 10% - lizibilitatea codului

Pentru predarea temei, trebuie să creați o arhivă cu numele următor: $nume\_complet\_cursant.zip, care va conține fișierul python scris de voi si toate fisierele si folderele folosite in timpul executiei programului, fără acel folder venv. Arhiva va fi trimisă pe mail la adresa [office@telacad.ro](mailto:office@telacad.ro), specificând clasa din care faceți parte.

Spor!