| Nom: | | | |
|------|------|------|--|

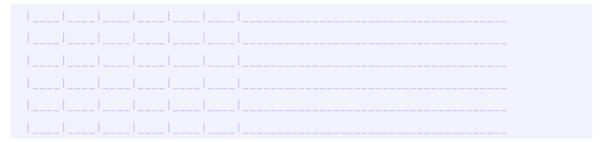
Exercice 1 5 points

1. Construire une classe Smartphone où chaque instance sera un objet disposant des attributs marque (de type String), modele (de type String) et ram (de type Int), demandés en paramètres dans la méthode constructeur.

2. Instancier dans une variable mon_tel le smartphone Samsung modèle GalaxyS22, disposant de 8 giga-octets de ram.

1____1___

3. Écrire une fonction prenant en paramètres deux téléphones de la classe Smartphone et affichant la marque et le modèle du téléphone ayant le plus de mémoire ram (sans se soucier du cas d'égalité).



Exemple d'utilisation de la classe :

Exercice 2 5 points

Écrire une classe Player qui :

- ne prendra aucun argument lors de son instanciation.
- affectera à chaque objet créé un attribut energie valant 3 par défaut.
- affectera à chaque objet créé un attribut alive valant True par défaut.
- fournira à chaque objet une méthode blessure qui diminue l'attribut energie de 1. Si l'attribut energie passe à 0, l'attribut alive doit passer à False et l'attribut energie ne doit plus pouvoir évoluer.
- fournira à chaque objet une méthode soin qui augmente l'attribut energie de 1.
- fournira à chaque objet une méthode miracle qui, si le personnage est mort, le rend à nouveau vivant avec un attribut energie valant 2.

>>> mario = Player()
>>> mario.blessure()
>>> mario.blessure()
>>> mario.blessure()
>>> mario.alive
False
>>> mario.soin()
>>> mario.alive
False
>>> mario.alive
False
>>> mario.energie
0
>>> mario.miracle()
>>> mario.soin()
>>> mario.energie
3

| | | | _ | l |
|---|--|--|----|---|
| | | | | |
| _ | | | _ | |
| | | | _ | |
| | | | _l | |
| _ | | | _ | |
| _ | | | _ | |
| | | | _ | |
| | | | _l | |
| | | | _ | |
| | | | _ | |
| | | | _l | l |

Exercice 3 10 points

On souhaite gérer les livres d'une bibliothèque, à l'aide de deux classes, la classe Livre et la classe Bibliothèque. Le cahier des charges ci-dessous comporte la description des deux classes, ainsi qu'un exemple concret de leur utilisation.

▶ classe Livre :

Elle comporte 3 attributs:

- titre : le titre du livre (type String)
- auteur : le nom de l'auteur (type String)
- etat : un nombre entier entre 0 et 5. Si l'état vaut 0, le livre est trop abîmé et doit être retiré de la bibliothèque. L'état d'un livre ne peut pas être négatif. (type Int)

Elle comporte 2 méthodes :

- degrade : ne prend aucun paramètre. Enlève 1 à l'état du livre.
- description : ne prend aucun paramètre. Affiche les renseignements sur le livre.

\triangleright classe Bibliotheque :

Elle comporte 1 attribut :

• livres : une liste, vide à la création de l'objet (type List)

Elle comporte 3 méthodes:

- ajoute : prend un objet de type Livre en paramètre et l'ajoute à la bibliothèque.
- supprime_livres_abimes : ne prend aucun paramètre. Enlève de la bibliothèque les livres trop abîmés.
- inventaire : ne prend aucun paramètre. Affiche le contenu de la bibliothèque.

► Exemple d'utilisation des classes

```
>>> ma_bibli = Bibliotheque()
>>> livre1 = Livre("Les Misérables", "Victor Hugo", 3)
>>> livre2 = Livre("Les fleurs du mal", "Charles Baudelaire", 1)
>>> ma_bibli.ajoute(livre1)
>>> ma_bibli.ajoute(livre2)
>>> livre1.description()
Titre : Les Misérables
Auteur : Victor Hugo
Etat: 3
>>> livre2.degrade()
>>> livre2.description()
Titre : Les fleurs du mal
Auteur : Charles Baudelaire
Etat: 0
>>> ma_bibli.inventaire()
Titre : Les Misérables
```

| Auteur : Victor Hugo |
|---------------------------------------|
| Etat : 3 |
| |
| Titre : Les fleurs du mal |
| Auteur : Charles Baudelaire |
| Etat : 0 |
| |
| >>> ma_bibli.supprime_livres_abimes() |
| >>> ma_bibli.inventaire() |
| Titre : Les Misérables |
| Auteur : Victor Hugo |
| Etat : 3 |

Écrire les classes Livre et Bibliothèque.

| '' |
|----|

Exercice 1 5 points

```
class Smartphone:
    def __init__(self, marque, modele, ram):
        self.marque = marque
        self.modele = modele
        self.ram = ram

mon_tel = Smartphone("Samsung", "GalaxyS22", 8)

def compare(tel1, tel2):
    if tel1.ram > tel2.ram:
        print(tel1.marque, tel1.modele)
    else:
        print(tel2.marque, tel2.modele)
```

Exercice 2 5 points

```
class Player:
    def __init__(self):
        self.energie = 3
        self.alive = True

def blessure(self):
        self.energie -= 1
        if self.energie == 0:
            self.alive = False

def soin(self):
        if self.alive:
            self.energie += 1

def miracle(self):
        if not self.alive:
        self.alive = True
        self.energie = 2
```

```
class Livre:
    def __init__(self, titre, auteur, etat):
        self.titre = titre
        self.auteur = auteur
        self.etat = etat
    def degrade(self):
        if self.etat > 0:
            self.etat -= 1
    def description(self):
        print("Titre :", self.titre)
        print("Auteur :",self.auteur)
        print("Etat :", self.etat)
class Bibliotheque:
    def __init__(self):
        self.livres = []
    def ajoute(self, livre):
        self.livres.append(livre)
    def supprime_livres_abimes(self):
        new_livres = []
        for livre in self.livres:
            if livre.etat > 0:
                new_livres.append(livre)
        self.livres = new_livres
        # autre méthode :
    def supprime_livres_abimes(self):
        for livre in self.livres:
            if livre.etat == 0:
                self.livres.remove(livre)
    def inventaire(self):
        for livre in self.livres:
            livre.description()
            print()
```