

Exercice 1 *8 points*

1.

```
class Bim:
    def __init__(self, nature, surface, prix_moy):
        self.nt = nature
        self.sf = surface
        self.pm = prix_moy

    def estim_prix(self):
        return self.sf * self.pm
```

2. Cette instruction renvoie la valeur 140000. Cette valeur sera de type Float.

3.

```
def estim_prix(self):
    if self.nt == 'maison':
        return self.sf * self.pm * 1.1
    elif self.nt == 'bureau':
        return self.sf * self.pm * 0.8
    else:
        return self.sf * self.pm
```

4.

```
def nb_maison(lst):
    nb = 0
    for bien in lst:
        if bien.nt == 'maison':
            nb += 1
    return nb
```

ou en une ligne pour épater la galerie :

```
def nb_maison(lst):
    return sum(bien.nt == 'maison' for bien in lst)
```

```
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()
```