

2026 sujet 9

Question 1

```
def distance(self, sommet):  
    return math.sqrt((self.x-sommet.x)**2 + (self.y-sommet.y)**2  
        + (self.z-sommet.z)**2)
```

Question 2

[La notion de sommet adjacent n'est pas donnée et semble étrange lorsqu'on regarde la méthode `est_adjacent`. Mais cela ne nous empêche pas de répondre à la question 2.]

```
def trouver_sommets_adjacents(self):  
    # comme un sommet n'est pas adjacent à lui-même  
    # d'après est_adjacent, on peut parcourir deux fois  
    # la liste sans précaution  
    for sommet1 in self.sommets:  
        for sommet2 in self.sommets:  
            if sommet1.est_adjacent(sommet2):  
                return sommet1, sommet2
```

Question 3

```
def estimation_impression(volume, param_imp):  
    volume_impression = volume * param_imp['remplissage'] / 100  
    temps = volume_impression / param_imp['vitesse_extrusion']  
    return temps
```

Question 4

```
objet = Objet3D()  
objet.ajouter_sommet(0, 0, 0)  
objet.ajouter_sommet(0, 2, 0)  
objet.ajouter_sommet(2, 2, 0)  
objet.ajouter_sommet(2, 0, 0)  
objet.ajouter_sommet(1, 1, 2)  
objet.ajouter_face([1, 2, 3, 4]) # Base  
objet.ajouter_face([1, 2, 5]) # Face  
objet.ajouter_face([2, 3, 5]) # Face  
objet.ajouter_face([3, 4, 5]) # Face  
objet.ajouter_face([4, 1, 5]) # Face  
objet.afficher()
```

Question 5

[Il y a une erreur dans le code de la fonction transformer, il manque un "s" à sommets dans la dernière ligne : `self.sommets = sommets`]

```
def transformer(self, rapport):
    """
    Applique une transformation d'échelle à l'objet 3D en
    renvoyant un nouvel objet
    """
    sommets = []
    for sommet in self.sommets:
        sommets.append(
            Sommet(sommet.x*rapport, sommet.y*rapport, sommet.z*rapport))
    nouvel_objet = Objet3D()
    nouvel_objet.sommets = sommets
    nouvel_objet.faces = self.faces
    return nouvel_objet
```

Et pour le tester :

```
objet = Objet3D()
objet.ajouter_sommet(0, 0, 0)
objet.ajouter_sommet(0, 2, 0)
objet.ajouter_sommet(2, 2, 0)
objet.ajouter_sommet(2, 0, 0)
objet.ajouter_sommet(1, 1, 2)
objet.ajouter_face([1, 2, 3, 4]) # Base
objet.ajouter_face([1, 2, 5]) # Face
objet.ajouter_face([2, 3, 5]) # Face
objet.ajouter_face([3, 4, 5]) # Face
objet.ajouter_face([4, 1, 5]) # Face
nouveau = objet.transformer(0.5)
nouveau.afficher()
```