

## Exercice 1 Organisateur de tâches

Un élève de NSI veut se simplifier la vie au quotidien. Il décide de créer un petit **organisateur** (une « todo list » en français) pour l'aider à suivre ce qu'il a à faire : réviser, ranger sa chambre, préparer un exposé, etc.

Dans cet organisateur, **chaque élément à faire** sera représenté par un **objet** de la classe **Tache**. Autrement dit, **une tâche = un objet**.

Description d'une tâche :

Chaque tâche doit stocker les informations essentielles suivantes :

- **Titre** (*titre*) : une courte phrase qui nomme la tâche. Ex. : "Réviser NSI", "Ranger la chambre", "Envoyer le devoir".
- **Priorité** (*priorite*) : un nombre entier indiquant l'urgence/importance. Convention proposée : 1 = très urgent, 2 = urgent, 3 = normal, 4 = faible, 5 = très faible.
- **Durée** (*duree*) : temps estimé pour réaliser la tâche, en minutes.
- **État** (*\_\_terminee*) : booléen (**True/False**) indiquant si la tâche est faite (**True**) ou à faire (**False**). Par défaut, une nouvelle tâche n'est **pas** terminée.

En suivant son cahier des charges, il écrit le code python suivant :

```
1 class Tache:
2     def __init__(self, t, p, d):
3         # t -> titre (str), p -> priorite (int 1 .. 5), d -> duree en minutes (int)
4         self.titre = t
5         self.priorite = p
6         self.duree = d
7         self.__terminee = False
8
9     def __repr__(self):
10        """Retourne une courte description textuelle de la tâche."""
11        if self.__terminee:
12            etat = "Tâche terminée"
13        else:
14            etat = "Tâche en cours"
15        return f"{self.titre} - priorité {self.priorite} - Durée prévue : {self.duree} min. - \n - {etat}"
16
17    def est_urgente(self):
18        """Renvoie True si la tâche est considérée urgente (la tâche étant très urgente ou \n - urgente)."""
19        return self.priorite in (1, 2)
20
21    def marquer_terminee(self):
22        """Marque la tâche comme terminée."""
23        self.__terminee = True
24
25
26 # Instance d'exemple pour l'analyse :
27 t1 = Tache("Faire du sport", 3, d=30)
```

1. Lire et analyser le code
  - (a) Lister les attributs de la classe.
  - (b) Parmi les attributs de la classe lequel est privé.
  - (c) Lister les méthodes de la classe.
  - (d) Donner les noms des paramètres du constructeur.
2. Comprendre l'instance déjà créée
  - (a) Avec `t1 = Tache("Faire du sport", 3, d=30)`, quelles sont les valeurs de `t1.titre`, `t1.priorite`, `t1.duree`, `t1.terminee` juste après la création ?

- (b) Écrire une ligne qui **affiche uniquement le titre** de la tâche.
  - (c) Quel sera le retour de l'instruction `t1.est_urgente()`.
3. Créer et utiliser vos propres instances.
- (a) Créez une instance **t2** représentant «Réviser NSI», de priorité 1 et de durée 1h30min.
  - (b) Que va afficher `print(t2)`.
  - (c) La tâche **t2** est terminée. Comment mettre à jour **t2**?
-