TP4\_Travail Pratique 4

Gestionnaire de projets

Table des matières

[1 Mise en situation 1](#_Toc1674185705)

[1.1 Informations générales 2](#_Toc953822080)

[2 DONNÉES ET INFORMATIONS 3](#_Toc917948278)

[2.1 Activité 3](#_Toc1355083945)

[2.2 Tâche 3](#_Toc1633510742)

[2.3 Projet 3](#_Toc2012670957)

[3 Consignes 3](#_Toc1138123801)

[3.1 Classe principale 4](#_Toc356422138)

[3.2 Activités 4](#_Toc861616856)

[3.3 Tâches 4](#_Toc355755131)

[3.4 Projet 6](#_Toc744280733)

[3.5 Programme principal 8](#_Toc1336450555)

[4 Remise 9](#_Toc1016051862)

**Consignes**

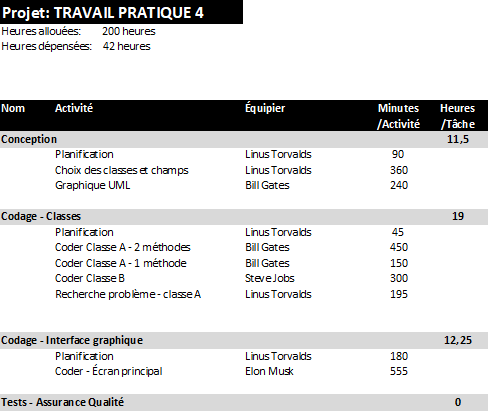
|  |
| --- |
| Toutes documentation permise.  Le code DOIT utiliser les notions vues en classe. |

# Mise en situation

## Informations générales

Vous devez modéliser un programme qui permet de gérer le temps de travail dépensé dans un projet, avec le temps passé par chaque équipier sur une activité relié à une tâche du projet.

Ci-dessous un exemple de rapport qui pourrait résulter de cette application.



Le **projet** est divisé en **tâches**, soit les objectifs principaux qui composent le projet.

Lorsqu’une personne travaille sur une tâche, elle réalise une **activité**, qui a une heure de début et une heure de fin.

La somme du temps passé pour chaque activité dans chaque tâche compose le temps total dépensé sur le projet.

Le temps sera conservé en minutes dans tout le programme, et sera parfois affiché en heures, ou en minutes ou une combinaison des deux.

# DONNÉES ET INFORMATIONS

## Activité

Une activité fait partie d’une tâche.

Lorsqu’une personne réalise une activité, elle entre:

* Son nom
* Une courte description du travail accompli
* La date et heure du début de l’activité
* La date et heure de la fin de l’activité

## Tâche

Une tâche est un objectif principal du projet.

Elle est composée de:

* Un nom
* Un total de temps dépensé, en minutes (conservé dans un entier de type **long**)
* La liste des activités soit un **tableau** d’objets **d’activités**

Lorsqu’une activité est entrée, elle est ajoutée au tableau de la tâche auquel elle correspond.

## Projet

Un objet Projet représente le projet au complet.

Il est composé de:

* Un nom
* Un nombre de minutes allouées au projet (un entier de type **long**).
* Un nombre de minutes dépensée jusqu’à maintenant, soit le total des minutes de toutes les activités de toutes les tâches (un entier de type **long**).
* La liste des tâches, soit un **tableau** d’objets de **tâches**.

# Consignes

## Classe principale

Dans une **package gestion\_projet**, faites une classe avec une méthode main, qui vous servira de programme principal.

## Activités

1. Dans un **package projets**, faites une classe pour les activités.
2. Ajoutez les champs à partir des descriptions précédentes.
3. Faites un constructeur:
   * Qui accepte un paramètre pour chacun des champs et qui initialise les champs à partir des paramètres.
4. Faites une méthode pour calculer les minutes dépensées dans la tâche:
   * Qui compte le nombre total de minutes à partir de la date/heure de début et de la date/heure de fin de l’activité.
   * Retourne un long, soit le nombre de minutes calculées
5. Faites une méthode pour afficher les informations de l’activité, sur une ligne. Cette méthode sera éventuellement utilisée pour produire les lignes d’activités dans le rapport du projet, respecter le format suggéré par l’exemple de la section **Mise en situation**.
6. Ajouter des méthodes d’accès et de mutation (getters et setters) seulement si vous en avez besoin.
7. Vous pouvez utiliser le programme principal pour tester le travail fait jusqu’à maintenant.

## Tâches

Les tâches sont modélisées dans une classe.

1. Dans le même **package projets**, faites une classe pour les tâches.
2. Ajouter un champ constant pour la taille du tableau d’activités et initialisez-le à **10**;
3. Ajoutez les autres champs à partir des descriptions précédentes.
4. Faites un constructeur:

* Qui accepte en paramètre seulement le nom de la tâche
* Qui initialise le nom de tâche à partir du paramètre
* Qui initialise le total de minutes dépensées à 0
* Qui initialise le tableau d’activités.
  + Pour le tableau d’activités, on décide qu’il sera rempli d’objets d’activité vides
  + On définit une **activité vide** comme une activité dont:
    - Le nom et la description sont des chaines vides
    - Le dates de début et de fin sont le 1 janvier 2000.
    - On pourra vérifier dans d’autres méthodes qu’une case du tableau ne contient pas d’activité si son nom est une chaine vide

1. Faire une méthode pour trouver le nombre d’activité présentes pour cette tâche:

* Qui parcoure le tableau d’activités pour trouver celles dont le nom n’est pas une chaine vide
* Qui retourne le nombre d’activités trouvées en tant que int.

1. Faites une méthode pour ajouter une activité

* Qui reçoit une activité en paramètre
* Qui ne retourne rien
* Qui ajoute l’activité au tableau d’activités
* Une activité est toujours ajoutée au tableau dans le prochain élément vide (activité vide)
* Vous pouvez utiliser la méthode écrite auparavant pour trouver le nombre d’activités du tableau et ainsi savoir à quel élément du tableau il faut ajouter l’activité.

1. Ajouter des méthodes d’accès et de mutation (getters et setters) seulement si vous en avez besoin.
2. Vous pouvez utiliser le programme principal pour tester le travail fait jusqu’à maintenant.

## Projet

Un projet est modélisé dans une classe.

1. Dans le même **package projets**, faites une classe pour les projets.
2. Ajouter un champ constant pour la taille du tableau de tâches et initialisez-le à **10**;
3. Ajoutez les autres champs à partir des descriptions précédentes.
4. Faites un constructeur:

* Qui accepte en paramètre le nom du projet et le nombre de minutes allouées au projet (un **long**).
* Qui initialise le nom du projet et les minutes allouées à partir des paramètres
* Qui initialise le nombre de minutes dépensées à 0
* Qui initialise le tableau de tâches.
  + Pour le tableau de tâches, on décide qu’il sera rempli d’objets de tâches vides
  + On définit une **tâche vide** comme une tâche dont:
    - Le nom de la tâche est une chaine vide
    - On pourra vérifier qu’une case du tableau ne contient pas d’activité si son nom est une chaine vide

1. Faire une méthode pour trouver le nombre de tâches dans le projet:

* Qui parcoure le tableau de tâches pour trouver celles dont le nom n’est pas une chaine vide
* Qui retourne le nombre de tâches trouvées en tant que int.

1. Faites une méthode pour ajouter une tâche

* Qui reçoit une tâche en paramètre
* Qui ne retourne rien
* Qui ajoute la tâche au tableau de tâches
* Une tâche est toujours ajoutée au tableau dans le prochain élément vide (tâche vide)
* Vous pouvez utiliser la méthode écrite auparavant pour trouver le nombre de tâches du tableau et ainsi savoir à quel élément du tableau il faut ajouter la tâche.

1. Faites une méthode pour trouver la position d’une tache dans la liste des taches à partir de son nom

* Qui reçoit un string en paramètre, soit le nom de la tâche à trouver
* Qui retourne un int, soit l’index dans la liste de la tâche qui correspond au nom reçu en paramètre
* Pour ce travail, on présume que le nom de tâche existe et qu’il est entré correctement

1. Faites une méthode pour ajouter une activité au projet

* Qui ne retourne rien
* Qui reçoit en paramètres le nom de la tâche à laquelle on ajoute une activité et un objet d’activité à ajouter à cette tâche
* Qui utilise la méthode précédente pour trouver l’index de la tâche en question à partir du nom de tâche reçu en paramètre
* Qui utilise la méthode pour ajouter une activité codée dans la classe de tâches

1. Ajouter une méthode pour mettre à jour les minutes dépensées:

* Qui ne retourne rien et n’accepte pas de paramètres
* Qui met à jour directement le champ des minutes dépensées pour le projet
* Qui parcours le tableau de tâches et pour chacune de ces tâches parcoure le tableau d’activités pour appeler la méthode de l’activité qui permet de calculer les minutes pour une activité.
* Qui compile le total de toutes les activités de toutes les tâches et met à jour directement le champ de nombre de minutes dépensées.
* Ajoutez un appel à cette méthode dans la méthode pour ajouter une tâche, pour que les minutes dépensées soient mises à jour à toutes les fois qu’on ajoute une tâche.
* Ajoutez un appel à cette méthode dans la méthode pour ajouter une activité, pour que les minutes dépensées soient mises à jour à toutes les fois qu’on ajoute une activité.

1. Ajoutez une méthode pour produire le rapport du projet, avec toutes les informations mentionnées dans l’exemple de la section Informations générales

* L’affichage est fait à la console
* Les temps alloué et temps dépensé sont affichés en heures

1. Ajouter des méthodes d’accès et de mutation (getters et setters) seulement si vous en avez besoin.
2. Vous pouvez utiliser le programme principal pour tester le travail fait jusqu’à maintenant.

## Programme principal

Dans le programme principal exécutez les actions suivantes, en utilisant les données fournies dans l’exemple du début de l’énoncé.

1. Créez un projet
2. Créez les tâches puis ajoutez-les au projet
3. Créez les activités, puis ajoutez-les aux tâches appropriées du projet
4. Produisez le rapport de projet

# Remise

Remettez le répertoire de projet **C34\_TP4** et tout son contenu sous format compressé (**zip**), sur **Léa** dans la section **travaux** sous **Travail Pratique 4**.