

Solutions

Exercice 1

Voir fichier Task.java pour les questions 1. et 2.

Identifier l'élément manquant

L'élément manquant pour que la classe soit complète est un constructeur. Afin de pouvoir être instanciée, une classe nécessite un constructeur.

```
public Task() {  
  
}  
  
public Task(int priority, int time) {  
    this.priority = priority;  
    this.time = time;  
}
```

On pourra définir un constructeur vide, ainsi qu'un constructeur prenant les valeurs de priorité et de temps à attribuer.

Exercice 2

L'important ici est d'être cohérent entre l'ajout et le retrait. Dans le fichier **Queue.java** fournit en solution, nous utilisons la méthode **addFirst** pour ajouter un élément et la méthode **removeLast** pour retirer un élément. Ainsi, nous créons, en contraignant l'ajout et le retrait, une structure de file : les nouveaux éléments sont ajoutés en début de liste et les plus anciens (en fin de liste) sont les premiers à être retirés.

Exercice 3

Nous avons optés pour une classe avec 3 attributs (1 par priorité). Il est possible d'envisager une solution avec un tableau de files. Voici dans ce cas à quoi ressemblerait la classe.

```
public class PriorityQueue {  
    private Queue[] priorities;
```

```

public PriorityQueue() {
    this.priorities = new Queue[3];
}

public void initFromTasks(Queue tasks) {
    Task task;
    int priority;

    while ( !tasks.isEmpty() ){
        task = tasks.remove();
        priority = task.getPriority();

        this.priorities[priority].add( task );
    }
}

public Task getTask() throws EmptyQueueException {
    Task task;
    if ( !this.priorities[0].isEmpty() ) {
        task = this.priorities[0].remove();
    } else if ( !this.priorities[1].isEmpty() ) {
        task = this.priorities[1].remove();
    } else if ( !this.priorities[2].isEmpty() ) {
        task = this.priorities[2].remove();
    } else {
        throw new EmptyQueueException("Empty priority queue.");
    }

    return task;
}

public boolean isEmpty() {
    return this.priorities[0].isEmpty() && this.priorities[1].isEmpty()
        && this.priorities[2].isEmpty();
}
}

```