22/01/2016

Projet de Refactoring

Baptiste CRESPIN - Nicolas CELLIER

baptiste-crespin@univ-tlse2.fr | nicolas-cellier@univ-tlse2.fr

Sommaire

Table des matières

[I – Refactoring 2](#_Toc441251348)

[1 - Mise en place des bonnes pratiques 2](#_Toc441251349)

[2 - Application.java 2](#_Toc441251350)

[3 – Projet.java 2](#_Toc441251351)

[4 – Task.java 2](#_Toc441251352)

[5 – ApplicationTest.java 2](#_Toc441251353)

[II – Ajout de fonctionnalités 2](#_Toc441251354)

[1 – Deadlines 2](#_Toc441251355)

[a) Date limite 2](#_Toc441251356)

[b) Today 3](#_Toc441251357)

[2 – Suppression 3](#_Toc441251358)

[3 – Visualisation 4](#_Toc441251359)

[a) View by date 4](#_Toc441251360)

[b) View by deadline 5](#_Toc441251361)

[c) View by project 5](#_Toc441251362)

[4 - Permettre qu’une tâche puisse « appartenir » à plusieurs projets en même temps 6](#_Toc441251363)

[5 - Faire en sorte que l’application maintienne en permanence 2 listes de tâches par projet 6](#_Toc441251364)

[6 - Faire en sorte qu’une tâche puisse être elle-même une liste de tâche 6](#_Toc441251365)

[III - Tests 6](#_Toc441251366)

# I – Refactoring

## 1 - Mise en place des bonnes pratiques

Nous avons fait le choix de débuter le projet par améliorer le plus possible le code déjà existant, afin de faciliter le travail ultérieur, et l’ajout de fonctionnalités futures.

Pour cela, nous avons commencé par

## 2 - Application.java

## 3 – Projet.java

## 4 – Task.java

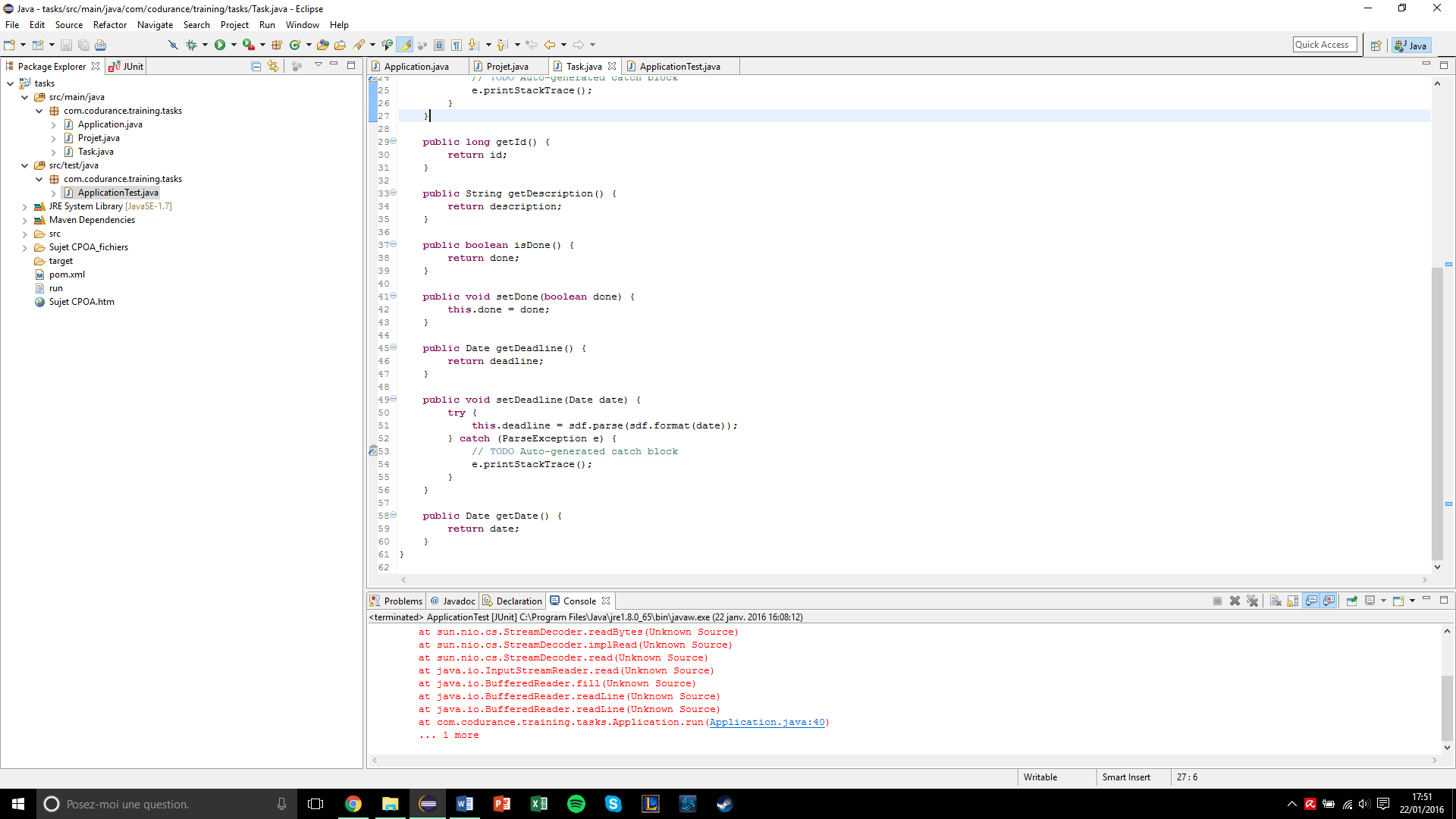
## 5 – ApplicationTest.java

# II – Ajout de fonctionnalités

## 1 – Deadlines

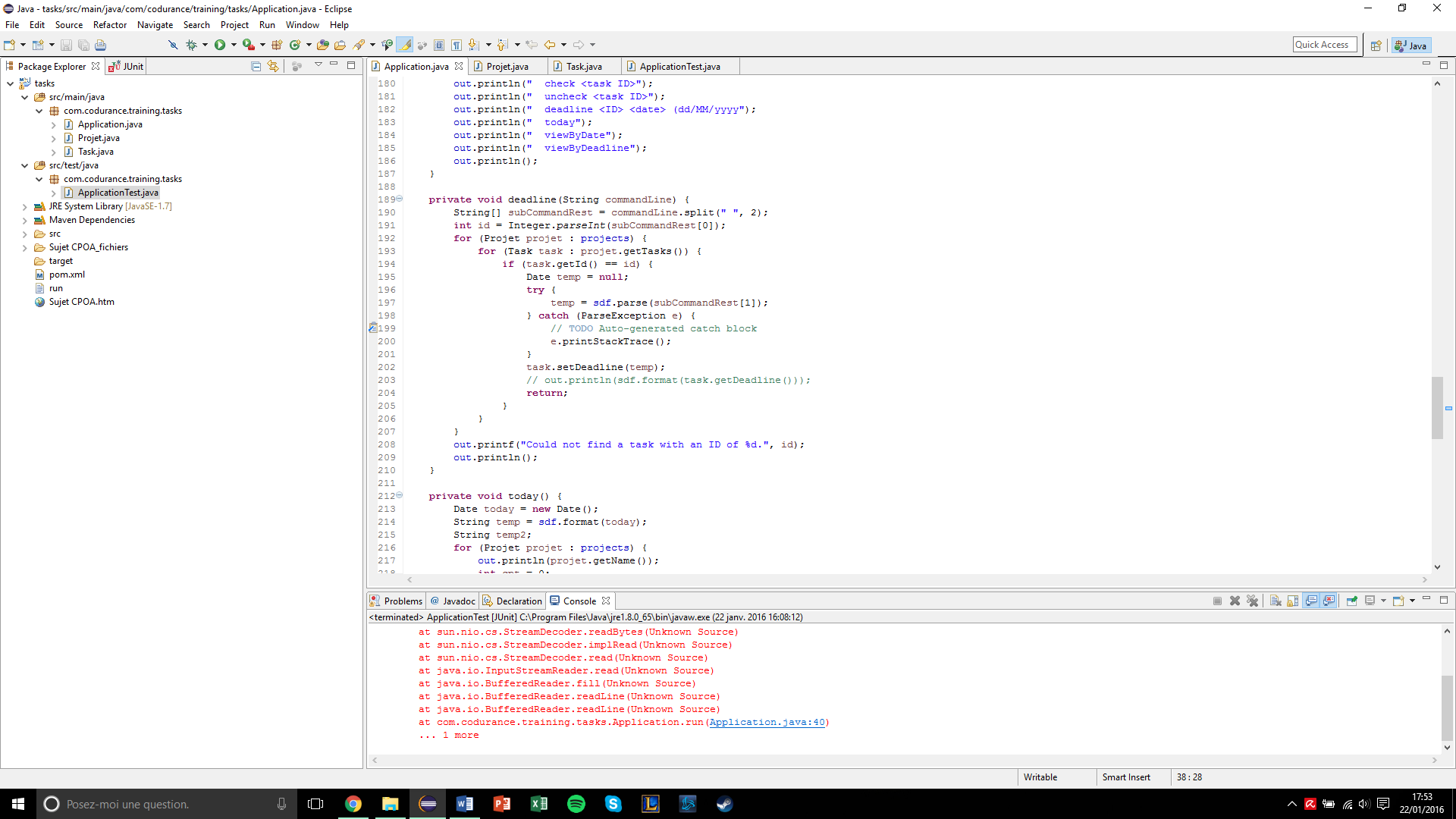
### Date limite

Pour l’ajout de la fonctionnalité deadline, nous avons commencé par ajouter dans Task.java un attribut de type private Date deadline dans la classe Task, ainsi qu’un getter et un setter.



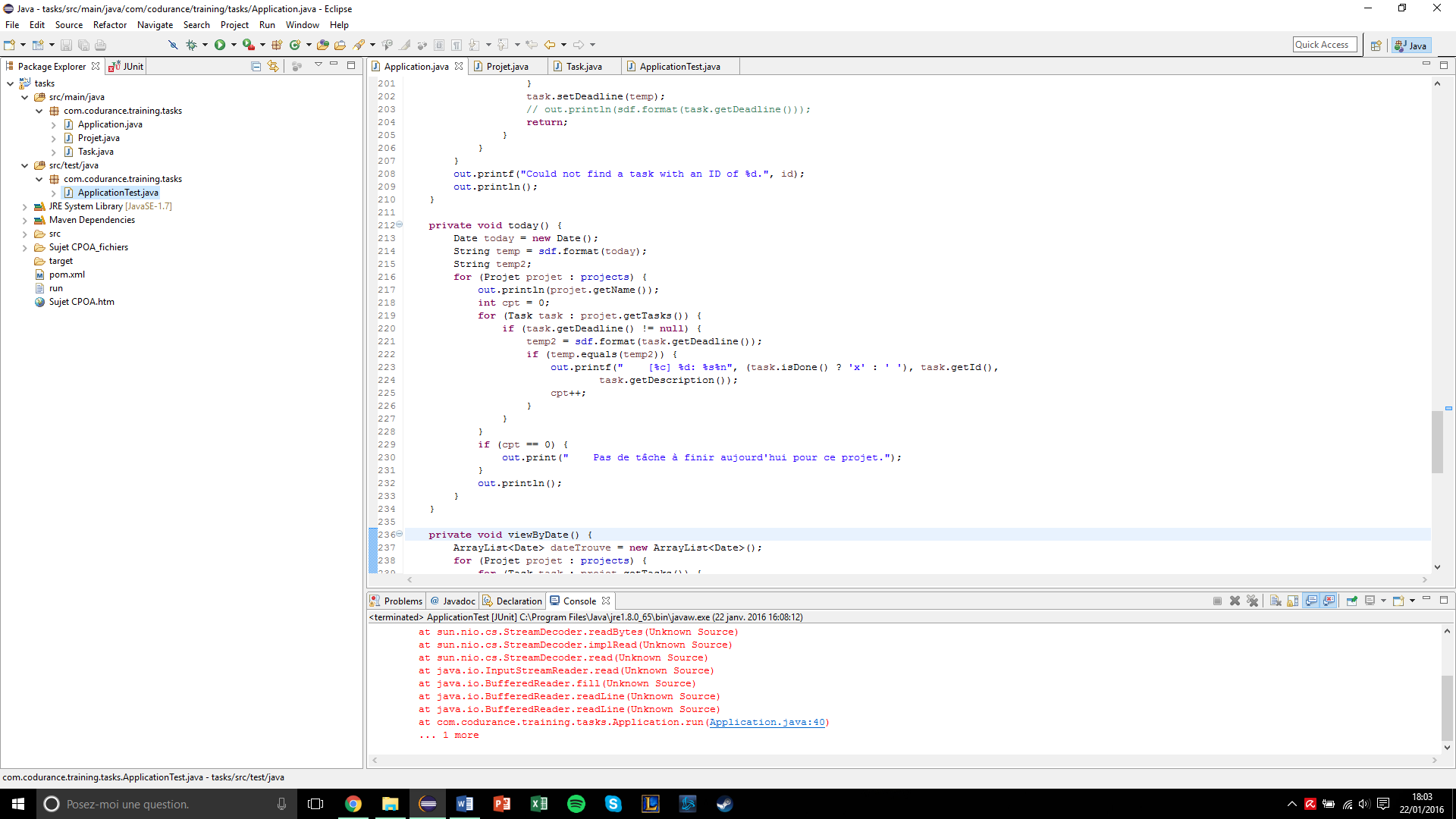
Puis nous avons initialisé celui-ci à null dans le constructeur par défaut.

Ensuite, après avoir ajouté les choix correspondant dans le menu, ainsi que l’aide, nous avons ajouté la méthode deadline, permettant de modifier la deadline d’une tache en donnant en paramètre l’ID de la tache ainsi que la date de la nouvelle deadline. Pour cela, elle parcourt toutes les taches de tous les projets jusqu’à rencontrer une tache à l’ID correspondant au paramètre. Puis une fois cette tâche trouvée, elle modifie sa deadline.



### Today

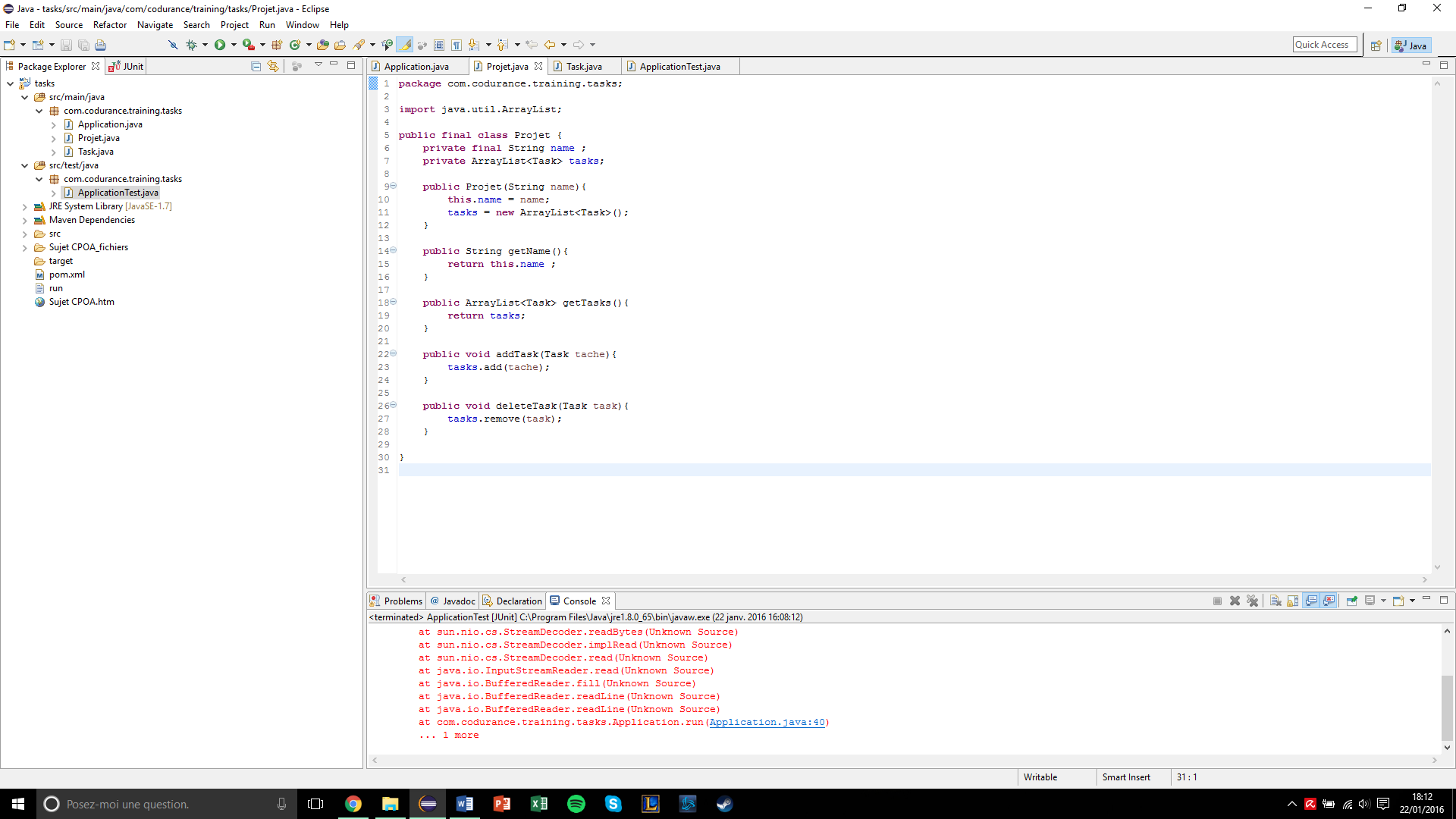
La commande today permet d’afficher toutes les taches dont la deadline est située aujourd’hui. Pour cela, elle va parcourir toutes les taches de tous les projets en affichant la tache au passage si la deadline de celle-ci est égale à celle d’aujourd’hui.



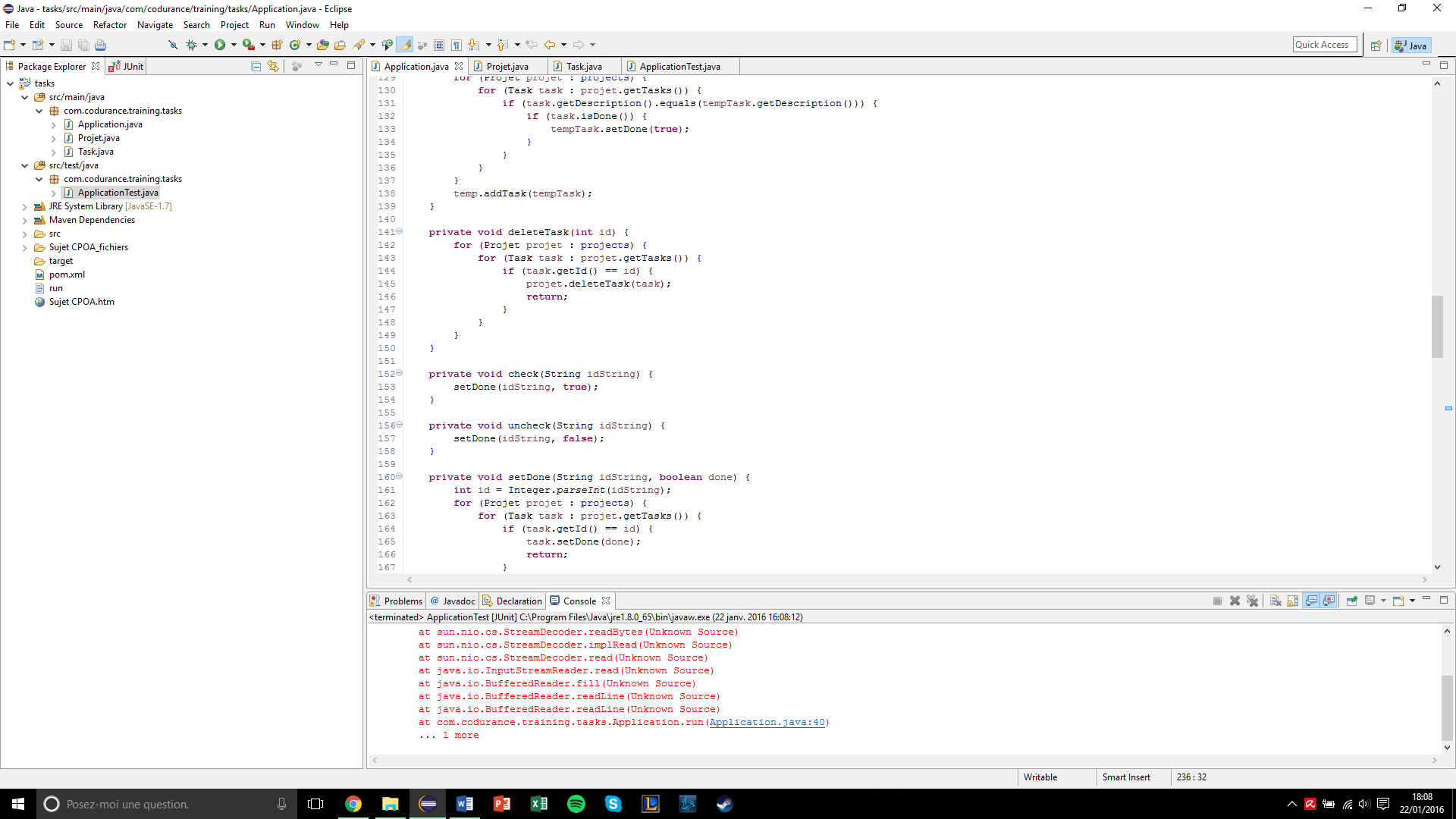
## 2 – Suppression

Pour implémenter la suppression d’une tâche, nous avons, après avoir ajouté les choix correspondant dans le menu, ainsi que l’aide, ajouter la méthode deleteTask, qui parcours toutes les taches de tous les projets jusqu’à trouver une tache ayant un ID correspondant. Si l’ID correspond, la tâche est supprimée du projet à l’aide de la fonction deleteTask de projet :

deleteTask de projet :



deleteTask de l’application :

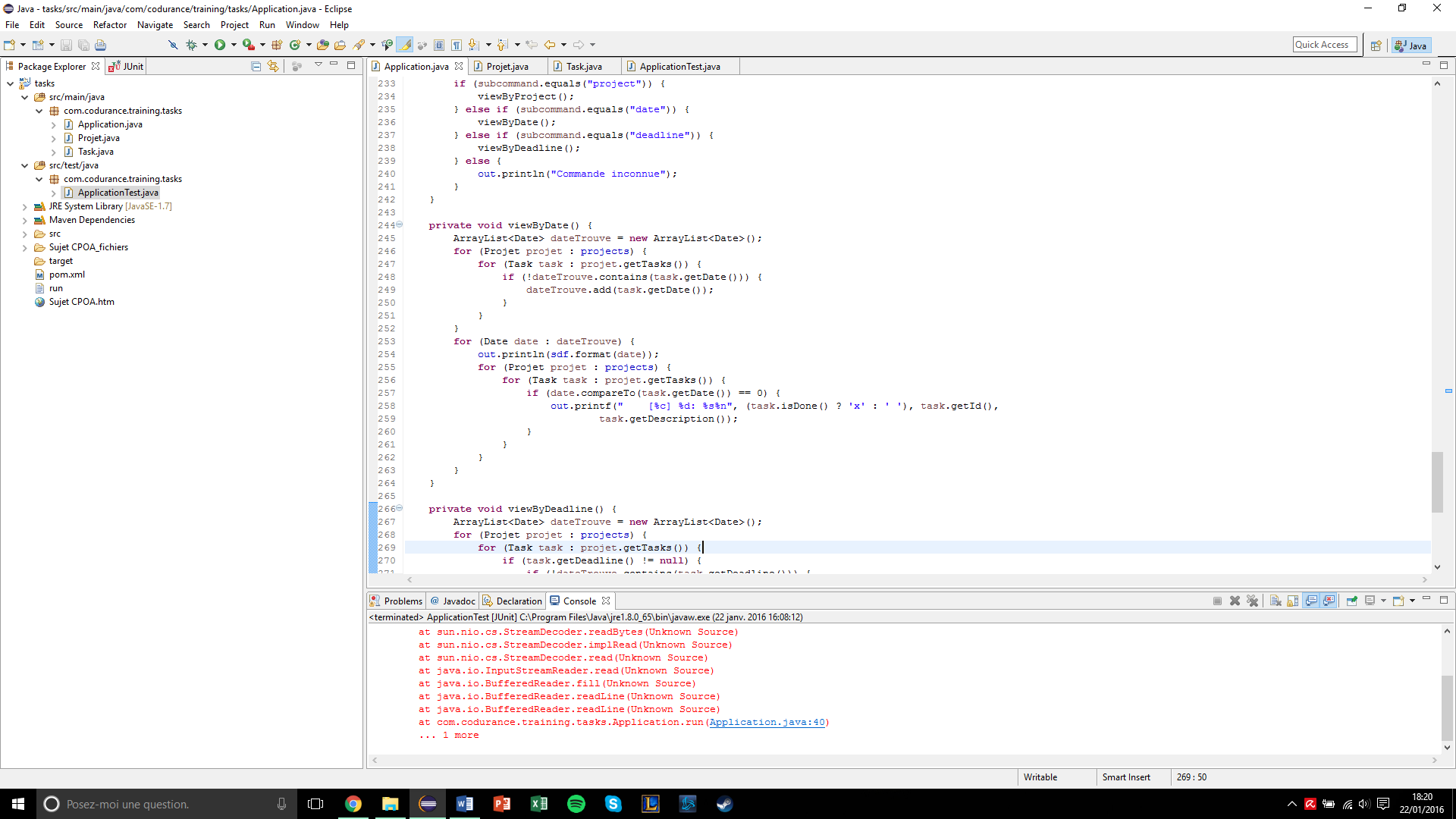


## 3 – Visualisation

### View by date

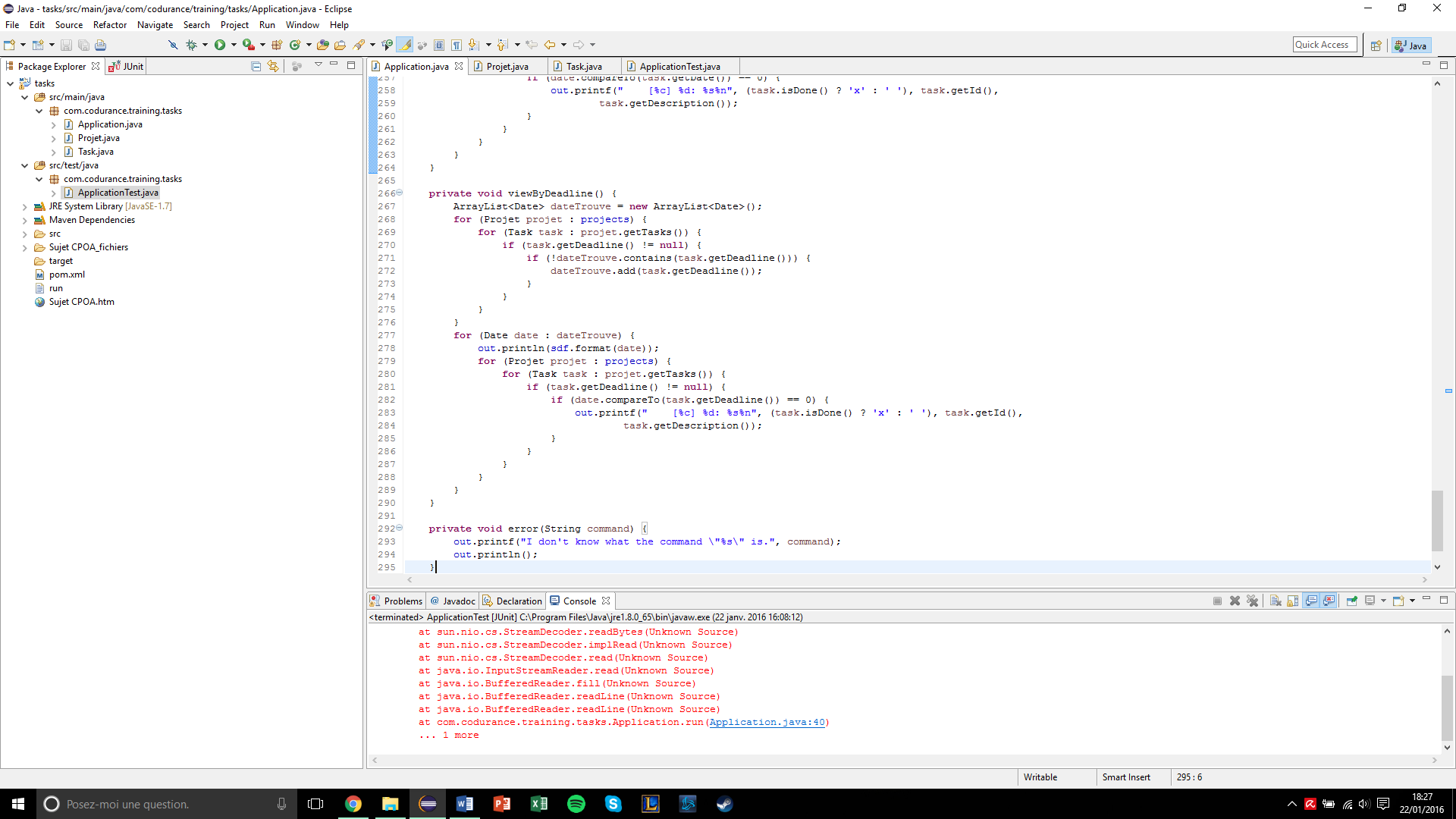
Pour l’ajout de la fonctionnalité view by date, nous avons commencé par ajouter dans Task.java un attribut de type private Date date dans la classe Task, ainsi qu’un getter. Cette date est automatiquement initialisée à la date d’aujourd’hui au moment de la création d’une tâche.

Ensuite, après avoir ajouté les choix correspondant dans le menu, ainsi que l’aide, nous avons ajouté la méthode viewByDate, permettant d’afficher toutes les taches, en fonction de leur date de création. Pour cela, la méthode parcourt toutes les taches de tous les projets en ajoutant leur date dans une liste, dans leur ordre de création. Puis, on parcourt la liste des dates, en affichant chacune des taches correspondantes.



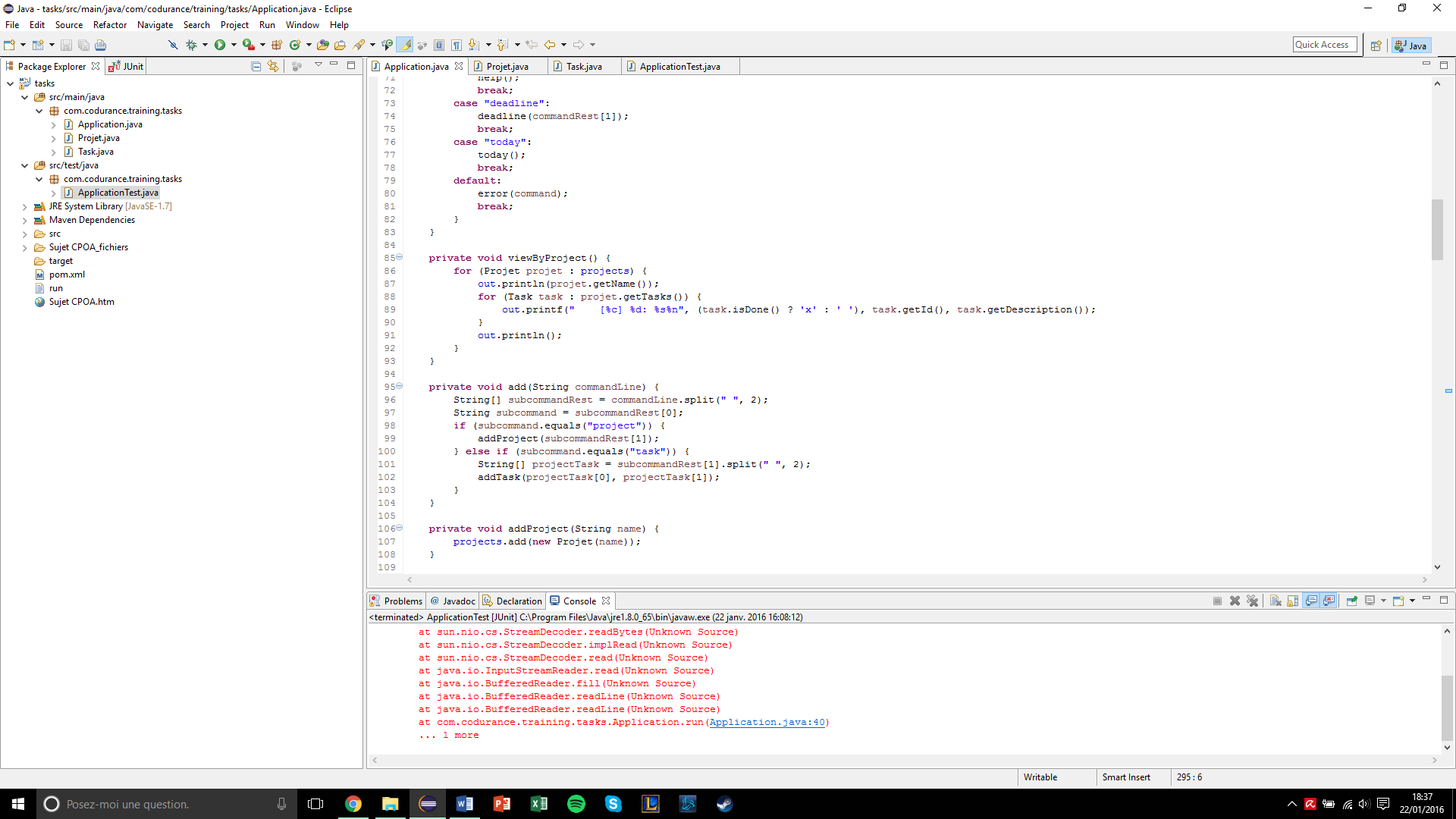
### View by deadline

View by deadline reprends le même principe que view by date, mais en utilisant cette fois ci les deadlines des projets. Cependant, toutes les tâches n’ont pas forcement de deadline, il a donc fallu ajouter quelques conditions afin de vérifier que deadline soit différent de null.



### View by project

La méthode view by project existait déjà sous le nom show. Il n’y a donc pas eu de modification à apporter hormis dans le menu et l’aide.



## 4 - Permettre qu’une tâche puisse « appartenir » à plusieurs projets en même temps

## 5 - Faire en sorte que l’application maintienne en permanence 2 listes de tâches par projet

## 6 - Faire en sorte qu’une tâche puisse être elle-même une liste de tâche

# III – Tests

Les tests ne fonctionnent pas à cause de machin donc on ne peut pas utiliser les tests.