

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet de programmation et génie logiciel

2017-2018

Création de fond de planning

Tuteur académique

Christophe LENTÉ

Étudiants

Hanyuan PENG (DI4)

Stéphane DELUCE (DI4)

Liste des intervenants

Nom	Email	Qualité
Hanyuan PENG	hanyuan.peng@etu.univ-tours.fr	Étudiant DI4
Stéphane DELUCE	stephane.deluce@etu.univ-tours.fr	Étudiant DI4
Christophe LENTÉ	christophe.lente@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Hanyuan Peng et Stéphane Deluce susnommés les auteurs.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Christophe Lenté susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

Les auteurs reconnaissent assumer l'entièr responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

Les auteurs attestent que les propos du document sont sincères et assument l'entièr responsabilité de la véracité des propos.

Les auteurs attestent ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

Les auteurs attestent que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

Les auteurs reconnaissent qu'ils ne peuvent diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

Les auteurs autorisent l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Hanyuan Peng et Stéphane Deluce, *Création de fond de planning*, Projet de programmation et génie logiciel, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{  
    author={Peng, Hanyuan and Deluce, Stéphane},  
    title={Création de fond de planning},  
    type={Projet de programmation et génie logiciel},  
    school={Ecole Polytechnique de l'Université François Rabelais de Tours},  
    address={Tours, France},  
    year={2017-2018}  
}
```



Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iii
1 Introduction	1
1 Présentation du projet	1
2 Environnement de développement	2
1 Langage de programmation	2
2 Outils de gestion de projet	2
3 Étude de projet	4
1 Analyse des besoins	4
2 Spécifications Fonctionnelles	4
3 Modélisation	5
3.1 UML	6
3.2 Définitions des classes	7
3.2.1 Modèle	7
3.2.2 Contrôleur.....	9
3.2.3 Vue	9

TABLE DES MATIÈRES

4 Planification	10
1 Diagramme de Gantt	10
2 Répartition des tâches	11
5 Implémentation	12
1 Choix de bibliothèques.....	12
2 Organisation	12
2.1 Organisation des fichiers	13
3 Fichiers d'entrées	13
3.1 Maquette	13
3.2 Calendrier	13
4 Analyse des données.....	13
5 Fichier de sortie	14
6 Documentation	14
7 Tests et validation	14
7.1 Résultat d'exécution	15
6 Conclusion	16
1 Améliorations & Évolutions	16
7 Annexes	17
1 Manuel d'utilisation	17



Table des figures

2 Environnement de développement

1	Tableau Trello	2
---	----------------------	---

3 Étude de projet

1	Diagramme de classes - UML	6
---	----------------------------------	---

4 Planification

1	Diagramme de Gantt	10
2	Listing des tâches	11

5 Implémentation

1	Organisation des dossiers.....	13
2	Exemple de documentation.....	14
3	Résultat d'exécution	15

1

Introduction

1 Présentation du projet

L'objectif de ce projet est de développer un programme qui devra créer un fond de planning en Excel afin de faciliter la création des emplois du temps. Grossièrement, le fond de planning se présente comme un quadrillage avec en ligne les cours (CM, TD, TP) et en colonne les semaines. Il est utilisé pour répartir les heures de CM, TD, TP par semestres et par enseignants. Des formules de calculs sont ajoutées à certaines cases afin de comptabiliser les heures placées.

Le programme prendra en entrée un fichier Excel contenant une maquette et les enseignants affectés pour les cours d'un semestre. Il fournira en sortie un fichier Excel. Il sera possible d'ajouter un fichier d'entrée pour les éléments manquants, comme les dates de rentrées, de vacances ...

2

Environnement de développement

1 Langage de programmation

Java nous a été imposé comme langage de programmation. Nous avons choisi comme IDE **Eclipse** pour développer le programme, car c'est un outil que nous avions tous les deux déjà utilisé.

2 Outils de gestion de projet

Nous avons fait le choix d'utiliser **Trello** afin de gérer les tâches de notre projet. Trello est un *Kanban* en ligne, il nous permet de visualiser les tâches qu'il nous reste à faire ainsi que celles sur lesquelles travail notre binôme.

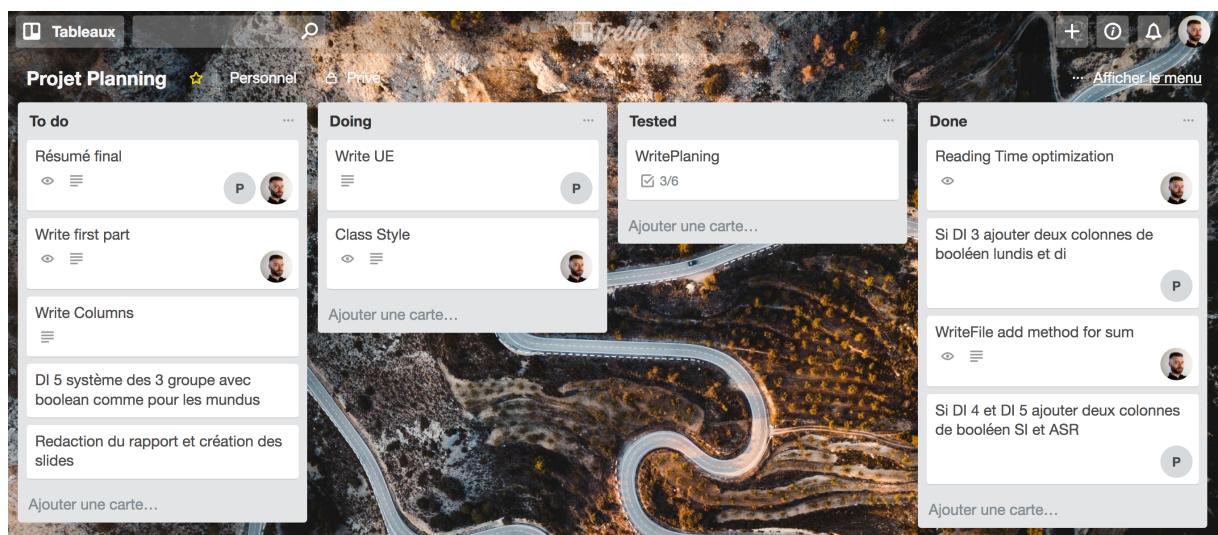


Figure 1 – Tableau Trello

Nous avons mis en place un **Git** comme système de versionning, car c'est un système avec lequel nous sommes déjà familiers. De plus, nous avons fait le choix d'héberger notre code à l'aide de **GitLab** qui nous permet de faire des projets privés. **GitLab** nous permet aussi de centraliser notre code à un même endroit.

Notre projet utilisera des bibliothèques, pour simplifier la gestion des dépendances, nous avons fait le choix d'utiliser **Maven** qui est un gestionnaire de dépendances pour Java.

3

Étude de projet

1 Analyse des besoins

Le logiciel que nous allons développer devra être capable de générer un ou plusieurs fonds de planning.

Il devra être capable de lire des fichiers Excel de type Xlsx, mais aussi pouvoir créer des fichiers Excel de même type, avec des formules, de la mise en forme et de la mise en forme conditionnelle.

2 Spécifications Fonctionnelles

Le logiciel que nous allons développer aura comme fonction de générer un ou plusieurs fonds de planning. Pour effectuer cette tâche, il devra en remplir plusieurs autres.

1. Lire un fichier Excel

(a) Rôle : Primaire

(b) Spécification :

- Entrée : Fichier Excel
- Sortie : Données lues
- Pré-condition : Fichier Excel au format valide
- Post-condition :
 - Cette fonction a pour but récupérer les données utiles à la génération d'un fond de planning, contenue dans un fichier Excel.

2. Parser les données récupérés dans les fichiers Excel

(a) Rôle : Primaire

(b) Spécification :

- Entrée : Données brutes
- Sortie : Données structurées
- Pré-condition : Les données brutes doivent être valides
- Post-condition :
 - Cette fonction a pour but structurer les données brutes.

3. Stocker les données

(a) Rôle : Primaire

(b) Spécification :

- Entrée : Données structurées
- Sortie : Objets construis avec les données
- Pré-condition : Les données brutes doivent être valides
- Post-condition :
 - Cette fonction sera utile pour stocker les données structurées dans des objets

4. Générer un objet Planning

(a) Rôle : Primaire

(b) Spécification :

- Entrée : Objets construis avec les données
- Sortie : Objets planning
- Pré-condition : Tous les objets doivent être présents
- Post-condition :
 - Cette fonction a pour un objet fond de planning complet.

5. Créer un fichier Excel contenant le fond de planning

(a) Rôle : Primaire

(b) Spécification :

- Entrée : Objets planning
- Sortie : Fichier Excel
- Pré-condition : L'objet doit être valide
- Post-condition :
 - Cette fonction a pour but transformer un objet planning en un fichier Excel.

3 Modélisation

Nous avons fait le choix d'utiliser l'architecture MVC (Modèle-vue-contrôleur) pour notre projet. La partie contrôleur est utilisée pour lire les données d'entrée, analyser les données et générer le planning. La partie modèle est utilisée pour stocker les données qu'on a lit et qu'on utilisera pour le planning. La partie vue est utilisée pour permettre les utilisateurs saisissent les commandes pour la génération des fond de planning.

3.1 UML

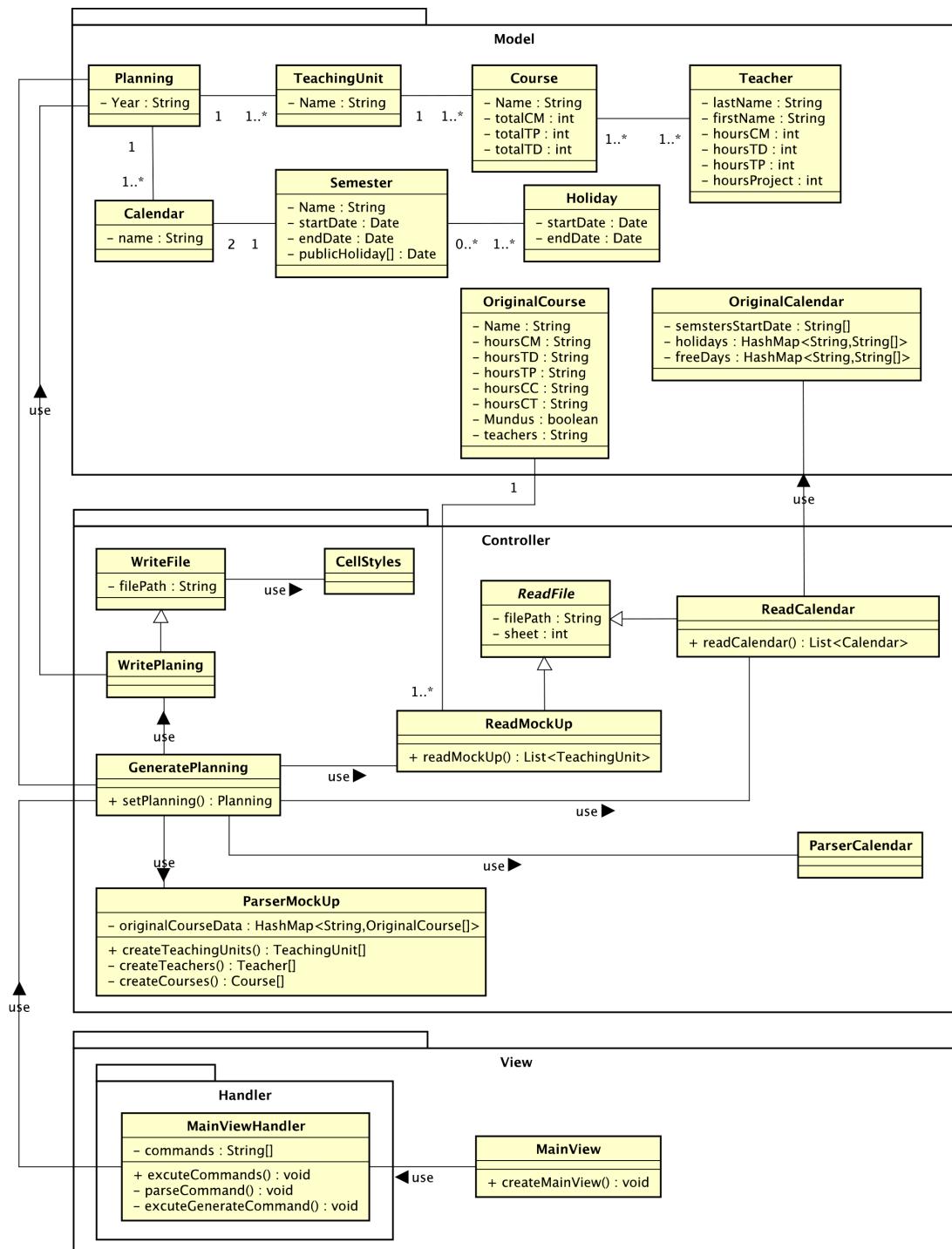


Figure 1 – Diagramme de classes - UML

3.2 Définitions des classes

3.2.1 Modèle

1. class Calendar

Le but de cette classe est de stocker les informations d'un calendrier. Un calendrier représente deux semestres.

Attributs :

- Le nom de calendrier, il stocke le nom de l'année.
- La liste des semestres, une arrayList des semestres de l'année.

2. class OriginalCalendar

Le but de cette classe est de stocker les informations originaux qui sont lues à partir de fichier calendrier pour analyser.

Attributs :

- Un LinkedHashMap des semestres, il stocke les informations originaux des semestres.
- Un LinkedHashMap des vacances, il stocke les informations originaux des vacances d'une année.
- Un LinkedHashMap des jours libres, il stocke les informations originaux des jours libres d'une année.

3. class Teacher

Le but de cette classe est de stocker les informations des professeurs d'un cours.

Attributs :

- Le nom de professeur.
- L'heure totale de CM que ce professeur fera.
- L'heure totale de TD que ce professeur fera.
- L'heure totale de TP que ce professeur fera.
- L'heure totale de Projet que ce professeur fera.
- L'heure totale de TD pour Mundus que ce professeur fera.
- L'heure totale de TP pour Mundus que ce professeur fera.

4. class Course

Le but de cette classe est de stocker les informations de cours. **Attributs :**

- Le nom de cours ;
- Une liste de tous les professeurs qui font ce cours.
- L'heure totale de CM de ce cours.
- L'heure totale de TD de ce cours.
- L'heure totale de TP de ce cours.
- L'heure totale de Projet de ce cours.
- Un booléen pour marquer si ce cours a un cc.
- Un booléen pour marquer si ce cours a un ct.
- Un booléen pour marquer si ce cours est un cours de type Mundus.
- Un string pour noter le type de ce cours est SI ou ASR.

5. class OriginalCourse

Le but de cette classe est de stocker les informations originaux qui sont lues à partir de fichier maquette pour analyser.

Attributs :

- Le nom de cours ;
- Une chaîne de caractère de l'affectation de professeurs.
- L'heure totale de CM de ce cours.
- L'heure totale de TD de ce cours.
- L'heure totale de TP de ce cours.
- L'heure totale de Projet de ce cours.
- Un booléen pour marquer si ce cours a un cc.
- Un booléen pour marquer si ce cours a un ct.
- Un booléen pour marquer si ce cours est un cours de type Mundus.

6. class TeachingUnit

Le but de cette classe est de stocker les informations d'une unité d'enseignement (UE).

Attributs :

- Le nom de UE ;
- Une liste de cours de cette UE.

7. class FreeDay

Le but de cette classe est de stocker les informations d'un jour libre.

Attributs :

- Le nom de ce jour libre.
- La date de ce jour.
- Le nombre de créneaux

8. class Planning

Le but de cette classe est de stocker les informations d'un planning pour l'écriture.

Attributs :

- L'année du planning que on veut générer.
- Une liste de l'unité enseignement de cette année dans un planning.
- Un calendrier de cette année.

9. class Holiday

Le but de cette classe est de stocker les informations des vacances.

Attributs :

- Le nom des vacances.
- La date début des vacances.
- La date fin des vacances.

10. class Semester

Le but de cette classe est de stocker les informations d'une semestre.

Attributs :

- Le nom d'une semestre, par exemple " S5 ".
- La date début d'une semestre.
- La date fin d'une semestre.
- Une liste de jour libre.
- Une liste des vacances.

3.2.2 Contrôleur

1. class GeneratePlanning

Le but de cette classe est de générer le planning à partir des informations stockées dans les objets Planning.

2. class ReadFile

Le but de cette classe est de fournir les fonctions basiques pour lire le fichier maquette et le fichier calendrier.

3. class WriteFile

Le but de cette classe est de fournir les fonctions basiques pour écrire le planning dans un fichier Excel.

4. class ParserCalendar

Le but de cette classe est d'analyser les informations de calendrier à partir d'un objet OriginalCalendar.

5. class ReadMockUp

Le but de cette classe est de lire le fichier maquette et de stocker les informations originaux d'un cours dans un objet OriginalCourse. On utilise la méthode getTeachingUnits de cette classe pour retourner un LinkedHashMap des unités d'enseignement. On utilisera ce LinkedHashMap pour analyser les informations d'une UE.

6. class WritePlanning

Le but de cette classe est d'écrire les informations d'un objet planning dans un fichier Excel.

7. class ParserMockUp

Le but de cette classe est d'analyser les informations de maquette à partir du . LinkedHashMap qui est fourni par la classe ReadMockUp.

8. class StylesLib

Le but de cette classe est de fournir des style de cellule.

9. class ReadCalendar

Le but de cette classe est de lire le fichier calendrier et de stocker les informations originaux d'un calendrier dans un objet OriginalCalendar.

10. class ToolBox

Le but de cette classe est de fournir des méthodes comme vérifier le type de cours.

3.2.3 Vue

1. class Mainview

Le but de cette classe est de créer une vue pour l'utilisateur.

2. class MainViewHandler

Le but de cette classe est de traiter les commandes qui sont saisies par l'utilisateur et de faire l'exécution des commandes en appelant aux contrôleurs.

4

Planification

1 Diagramme de Gantt

Pour mener à terme ce projet, il nous a fallu le décomposer en tâches, leurs déterminer une priorité et en estimer leurs durées. Ces informations nous ont permis de réaliser un diagramme de Gantt prévisionnel qui à été mis à jour en cours de projet.

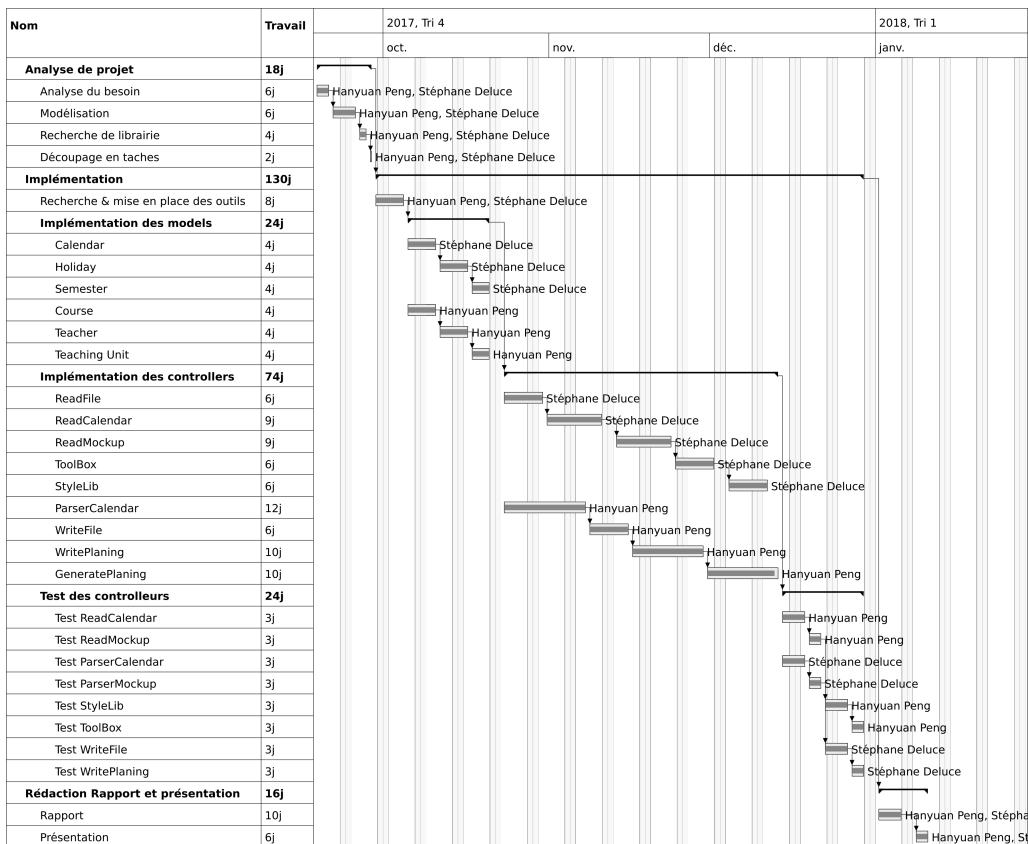


Figure 1 – Diagramme de Gantt

Nous avons paramètré le calendrier avec deux heures de travail par jour, avec des semaines de cinq jours. La répartition de notre travail n'a pas forcément respecté ce calendrier, mais la quantité de travail fournit est équivalente.

2 Répartition des tâches

Nous avons fait l'analyse de projet ensemble, après avoir définie la liste des tâches à effectuer pour le reste du projet, nous nous les sommes réparties de manière équitables.

Durant le projet, nous avons fait des points réguliers pour partager nos remarques et nos questions.

TPÉ	Nom	Démarré	Terminé	Travail	Durée	Latitude	Coût	Assigné à	Etat d'avancement (%)
1	Analyse de projet	sept. 18	sept. 28	18j	9j		0		100
1.1	Analyse du besoin	sept. 18	sept. 20	6j	3j		0	Hanyuan Peng, Stéphane Deluce	100
1.2	Modélisation	sept. 21	sept. 25	6j	3j		0	Hanyuan Peng, Stéphane Deluce	100
1.3	Recherche de librairie	sept. 26	sept. 27	4j	2j		0	Hanyuan Peng, Stéphane Deluce	100
1.4	Découpage en tâches	sept. 28	sept. 28	2j	1j		0	Hanyuan Peng, Stéphane Deluce	100
2	Implémentation	sept. 29	déc. 29	130j	66j		0		0
2.1	Recherche & mise en place des outils	sept. 29	oct. 4	8j	4j		0	Hanyuan Peng, Stéphane Deluce	100
2.2	Implémentation des models	oct. 5	oct. 20	24j	12j		0		0
2.2.1	Calendar	oct. 5	oct. 10	4j	4j		0	Stéphane Deluce	100
2.2.2	Holiday	oct. 11	oct. 16	4j	4j		0	Stéphane Deluce	100
2.2.3	Semester	oct. 17	oct. 20	4j	4j		0	Stéphane Deluce	100
2.2.4	Course	oct. 5	oct. 10	4j	4j		0	Hanyuan Peng	100
2.2.5	Teacher	oct. 11	oct. 16	4j	4j		0	Hanyuan Peng	100
2.2.6	Teaching Unit	oct. 17	oct. 20	4j	4j		0	Hanyuan Peng	100
2.3	Implémentation des controllers	oct. 23	déc. 13	74j	38j		0		0
2.3.1	ReadFile	oct. 23	oct. 30	6j	6j	2j	0	Stéphane Deluce	100
2.3.2	ReadCalendar	oct. 31	nov. 10	9j	9j		0	Stéphane Deluce	100
2.3.3	ReadMockup	nov. 13	nov. 23	9j	9j	1j	0	Stéphane Deluce	100
2.3.4	ToolBox	nov. 24	déc. 1	6j	6j		0	Stéphane Deluce	100
2.3.5	StyleLib	déc. 4	déc. 11	6j	6j	2j	0	Stéphane Deluce	100
2.3.6	ParserCalendar	oct. 23	nov. 7	12j	12j		0	Hanyuan Peng	100
2.3.7	WriteFile	nov. 8	nov. 15	6j	6j		0	Hanyuan Peng	100
2.3.8	WritePlaning	nov. 16	nov. 29	10j	10j		0	Hanyuan Peng	100
2.3.9	GeneratePlaning	nov. 30	déc. 13	10j	10j		0	Hanyuan Peng	95
2.4	Test des contrôleurs	déc. 14	déc. 29	24j	12j		0		0
2.4.1	Test ReadCalendar	déc. 14	déc. 18	3j	3j		0	Hanyuan Peng	100
2.4.2	Test ReadMockup	déc. 19	déc. 21	3j	3j		0	Hanyuan Peng	100
2.4.3	Test ParserCalendar	déc. 14	déc. 18	3j	3j		0	Stéphane Deluce	100
2.4.4	Test ParserMockup	déc. 19	déc. 21	3j	3j		0	Stéphane Deluce	100
2.4.5	Test StyleLib	déc. 22	déc. 26	3j	3j		0	Hanyuan Peng	100
2.4.6	Test ToolBox	déc. 27	déc. 29	3j	3j		0	Hanyuan Peng	100
2.4.7	Test WriteFile	déc. 22	déc. 26	3j	3j		0	Stéphane Deluce	100
2.4.8	Test WritePlaning	déc. 27	déc. 29	3j	3j		0	Stéphane Deluce	100
3	Rédaction Rapport et présentation	janv. 1	janv. 10	16j	8j		0		0
3.1	Rapport	janv. 1	janv. 5	10j	5j		0	Hanyuan Peng, Stéphane Deluce	100
3.2	Présentation	janv. 8	janv. 10	6j	3j		0	Hanyuan Peng, Stéphane Deluce	100

Figure 2 – Listing des tâches

5

Implémentation

1 Choix de bibliothèques

Pour réaliser la lecture et l'écriture des fichiers Excel, nous avons fait le choix d'utiliser une bibliothèque. Après avoir effectué des recherches, nous avons trouvé deux bibliothèques qui pourraient répondre à nos besoins.

La première, [Java Excel API](#), qui est un projet Open Source mis à jour pour la dernière fois en 2009.

La seconde, [POI Apache](#), gratuit, mais sous licence [Apache](#). Elle a été mise à jour pour la dernière fois le 15 Septembre 2017. Ce n'est pas forcément gage de bonne qualité, mais ça signifie que le projet évolue et qu'il n'est pas à l'abandon. Cette bibliothèque est très bien documenté et utilisé par une large communauté.

Après quelques essais avec les deux bibliothèques, nous avons fais le choix d'utiliser [POI Apache](#) pour réaliser ce projet.

2 Organisation

Tel que vu précédemment (1), nous avons choisi [POI Apache](#) comme bibliothèque pour la lecture / écriture de fichiers Excel. Pour l'ajouter à notre projet, il faut ajouter ces lignes à notre fichier pom.xml. Maven récupérera automatiquement les bibliothèques renseignées dans ce fichier.

```
1 <dependency>
2   <groupId>org.apache.poi</groupId>
3   <artifactId>poi-ooxml</artifactId>
4   <version>3.17</version>
5 </dependency>
```

2.1 Organisation des fichiers

Les groupes des classes modèles, vues et contrôleurs, chacunes ont été placés dans un package, ce nous donne l'arborescence suivante :

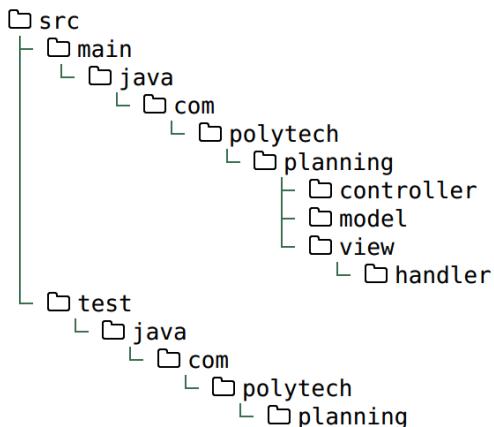


Figure 1 – Organisation des dossiers

3 Fichiers d'entrées

Les fichiers d'entrées sont des fichiers Excel de type .xlsx.

3.1 Maquette

Le fichier maquette présente les unités d'enseignements (UE) avec les cours pour chaque UE par semestres. Ils sont définies par un nom et un volume horaire par type de cours. Des enseignants sont attribués à des cours, il peut y en avoir plusieurs. Ils peuvent être attribués à certains groupes et certain type de cours (CM, TD, TP).

3.2 Calendrier

Le calendrier permet de définir les dates de début et de fin de chaque semestres. Il contient aussi les dates de vacances pour l'année complète ainsi que les jours libres qui sont définies par un nom une date et un créneau (correspond à 2 heures).

4 Analyse des données

Les données lues doivent être analysées pour ensuite être stockées dans des objets. La partie la plus complexe est l'affectation de professeur, il faut analyser une chaîne de caractère pour chaque cours. Nous avons utilisé les expressions régulières et des méthodes basiques de class String pour analyser la chaîne de caractère.

Par exemple, pour une chaîne de caractère qui contient plusieurs professeurs, nous utilisons ; pour les séparer et pour les types de cours (CM, TD, TP), nous utilisons , pour les séparer. Donc nous utilisons la méthode `split()` pour trouver les professeurs ou les types de cours différents.

Pour trouver l'heure et les groupes de chaque type de cours qu'un professeur fait, nous avons utilisé les expressions régulières. Afin d'analyser la chaîne de caractère correctement, il y a des règles à respecter :

1. Il n'existe pas d'espace entre le nombre d'heure et 'h'. Par exemple, il faut utiliser **8h** mais pas **8 h**.
2. Si un type de cours (TD, TP) est suivi par les groupes de DI3, on peut mettre un **x*gr** (* représente *nombre de groupe*) après TD/TP, ou on peut mettre rien.
3. Si un type de cours (TD, TP) est suivi par le groupe Mundus, il faut mettre **Mundus** ou **+Mundus** après TD/TP.

Les règles 2 et 3 sont uniquement pour année 3. Il y a d'autres règles qui peuvent être trouvées dans le manuel.

5 Fichier de sortie

Le fichier de sortie contient le planning d'une année qui est générée à partir des fichiers d'entrées. Il présente les unités d'enseignements (UE) avec les cours pour chaque UE par semestres. Des enseignants sont attribués à des cours. Le volume d'heure que l'enseignant fait pour chaque type de cours peut être placé par semaine dans un semestre. Le volume de travail par étudiant peut être calculé par des formules.

6 Documentation

Afin de rendre notre code évolutif, nous avons généré une documentation. Nous avons utilisé JavaDoc. Cette documentation a pour but d'aider un éventuel développeur à reprendre et comprendre le projet.

Voici un exemple de documentation sur la méthode *searchContent*, présent dans la class *ReadFile*.

```
/**
 * Get the coordonates of cell who contain the search content
 *
 * @param sheetNb
 *         Number of the sheet to be readed
 * @param content
 *         the search String
 * @return A table with two values, the first the row number and the second the
 *         column number of the first content find
 */
public int[] searchContent(int sheetNb, String content, boolean startWith) throws NullPointerException { }
```

Figure 2 – Exemple de documentation

7 Tests et validation

Les tests unitaires ont été réalisés après l'écriture des méthodes de chaque classe. Ils ont été faits de manières croisées, les méthodes écrites par l'un de nous deux ont été testées par l'autre. Nous avons utilisé la bibliothèque **JUnit** pour réaliser nos tests unitaires.

Certains tests visuels, pour le formatage des feuilles Excel sont faites visuellement. Nous n'avons pas trouvé d'autres méthodes pour vérifier ceci.

7.1 Résultat d'exécution

Pour exécuter le programme, il faut utiliser cette commande :

```
1  java -jar planning-generator.jar <startYear/endYear> <-all|-di3|-di4|-di5> ↵  
    <MaquetteFile.xlsx> <CalendarFile.xlsx>
```

```
→ ~ java -jar planning-generator.jar 2017/2018 -all Maquette.xlsx Calendar.xlsx  
***** Planning Generator *****  
Excuting command...  
Reading Calendar...  
Reading MockUp...  
Parsing data...  
Writing data into file...  
Creating <Planning Année 3 DI 2017 - 2018.xlsx> finished  
Reading Calendar...  
Reading MockUp...  
Writing data into file...  
Creating <Planning Année 4 DI 2017 - 2018.xlsx> finished  
Reading Calendar...  
Reading MockUp...  
Writing data into file...  
Creating <Planning Année 5 DI 2017 - 2018.xlsx> finished
```

Figure 3 – Résultat d'exécution

6

Conclusion

Durant le projet, nous avons rencontré quelques de problèmes. Comme par exemple le temps de lecture des fichiers d'entrés était anormalement long. Après des recherches auprès de la documentation et de la communauté de la bibliothèque POI Apache, nous avons trouvé la solution.

Il est possible d'utiliser un fichier Excel en lecture et écriture ou seulement en lecture. Le temps de lecture est divisé par 10 si on l'utilise uniquement en lecture.

Le code qui posait problème

```
1 this.wb = WorkbookFactory.create(file);
```

Le code corrigé

```
1 InputStream st = new FileInputStream(this.file);
2 this.wb = new XSSFWorkbook(st);
```

Nous avons aussi uniformisé le formatage des données pour l'attribution des groupes d'étudiants aux enseignants, afin de simplifier l'analyse des données présentes dans la maquette des unités d'enseignements. (4 (Chapitre 5))

Ce projet nous à permis de prendre en main de nouveaux outils, sur lesquels nous nous sommes auto-formés comme par exemple Maven ou la bibliothèque POI Apache.

Nous avons apprécié travailler ensemble, la communication a été une force au sein de notre équipe. Nous avons essayé d'appliquer le plus possibles les bonnes pratiques de Génie Logiciel (Git, Trello, Documentation, ...).

1 Améliorations & Évolutions

Le projet en l'état, peut évoluer, pour ajouter de nouvelles fonctionnalités.

La gestion des options pour les cinquièmes année.

La mise en forme conditionnelle afin de mettre en avant visuellement les erreurs de remplissage du fichier. La librairie **POI Apache** permet cette fonctionnalité.

7

Annexes

1 Manuel d'utilisation

Comme nous n'avions rien d'imposé pour la création du manuel utilisateur, nous avons fait le choix d'utiliser AsciiDoc.

Format des fichiers

Le générateur de fond de planning prend en entrée deux fichiers. Le premier fichier contient la **maquette des enseignements** de l'année. Le second contient le **calendrier de l'année** avec les dates des semestres, les dates des vacances et les jours libres. (Jours fériés et créneaux libres)



Les fichiers d'entrées doivent être des fichiers excel de type → **.xlsx**

Maquette

Structure du fichier maquette

La structure du fichier maquette n'est pas strict, mais il y a plusieurs choses à respecter.



La position des onglets est importante, elle est utilisée pour indiquer les semestres. L'ordre est **[La première feuille - S5 - S6 - S7 - S8 - S9 - S10]**

- Il doit y avoir une feuille par semestre, soit deux feuilles par années.
- Le titre des feuilles n'est pas utilisé par le générateur de fond de planning.
- Les colonnes doivent être nommées comme sur l'image ci-dessous.



Le générateur de fond de planning n'est pas sensible à la case, ni aux accents.

Tous les titres de colonnes ne sont pas importants, voici la liste de celles utilisées par le générateur de fond de planning :

- Unité d'enseignement** (*Doit être décalé d'une colonne à droite par rapport au nom d'une Unité d'enseignement et dans cette colonne se trouvera le nom des matières*)
- Cours**
- TD**
- TP**
- Projet**
- CC**
- CT**
- Affectations des enseignements**

Entête du fichier maquette

Diplôme d'ingénieur spécialité Informatique : année 3 - S5									
2017-2018	UNITE D'ENSEIGNEMENT	Volume horaire				Contrôle des connaissances		Poids UE	ECTS
		Cours	TD	TP	Projet	CC	CT		
SOUTIEN									

Le seul point important est la disposition des titres des unités d'enseignement, ils doivent être décalés d'une colonne à gauche par rapport à la colonne portant le nom "Unité d'enseignement".

Format de l'affectations des enseignements

Nous allons voir dans cette partie comment formater le contenu de la "Affectations des enseignements". Les lignes qui sont au même niveau que le nom de l'Unité d'enseignement n'est pas importante car les données présentes dans ces cases ne seront pas lu par le générateur de fond de planning. Les lignes en face des nom de matières, quand à elles ont un format particulier.

Structure du contenu

```
<nomEnseignant>, <x>hCM, <y>hTD x<nbGroupes>gr (+ Mundus), <z>hTP x<nbGroupes> (+ Mundus)
```

Dans certain cas, il y a plusieurs enseignants pour une seule matière. Pour renseigner plusieurs enseignants, il faut les séparer par un ;. Comme sur l'exemple ci-dessous. Le programme est cases sensible pour l'affectations des enseignements.

Exemple de données pour la colonne "Affectations des enseignements"

Affectation enseignement et responsabilité UE 2017-2018

C.Tacquard

C.Tacquard,16hCM, 20hTD x3gr + Mundus, 16hTP x3gr + Mundus ; M.

Martineau,12hTP x3gr



Les **virgules** sont très importantes, elles sont utilisées lors de la lecture du fichier par le générateur de fond de planning comme caractère de séparation.



Pour un professeur, s'il fait tous les CM, TD et TP pour tous les groupes (Mundus inclu), on peut mettre que le nom de professeur.

Calendrier

Structure du fichier calendrier

La structure du fichier calendrier est simple. Il y trois feuilles dans le fichier calendrier. Ils sont **[Semestres - Vacances - Jours libres]**.



Le titre des feuilles n'est pas utilisé par le générateur de fond de planning. Mais **l'ordre est important**.

Semestres

Le générateur de fond de planning utilise le nom des semestres (S5, S6, S7, S8, S9, S10), ils doivent être suivis de la date de début du semestre puis par la date de fin. Le titre des colonnes n'est pas utilisé.



Le format des dates doit être **jj/mm/yyyy**.

Présentation des semestres

	Nom	Date de début	Date de fin
DI3	S5	09/11/2017	01/22/2018
	S6	01/22/2018	06/04/2018
DI4	S7	09/11/2017	01/22/2018
	S8	01/22/2018	06/04/2018
DI5	S9	09/11/2017	01/22/2018
	S10	01/22/2018	04/08/2018

Vacances

Pour les vacances, l'élément utilisé par le générateur de fond de planning est le titre de la colonne du nom des vacances qui doit être "**Nom**". Le nom de la période de vacances doit être suivi par la date de début puis par la date de fin.



Le format des dates doit être **jj/mm/yyyy**.

Présentation des vacances

Nom	Date de début	Date de reprise
Vacances de la Toussaint	10/30/2017	11/06/2017
Vacances de Noël	12/25/2017	01/08/2018
Vacances d'hiver	02/26/2018	03/05/2018
Vacances de printemps	04/23/2018	05/07/2018

Jours libres

Comme pour les vacances, le titre de la colonne des noms des jours libres, doit être "**Nom**". Le nom du jour libre doit être directement suivi par la date, puis par le nombre de créneaux (un créneau équivaut à 2 heures).



Le format des dates doit être **jj/mm/yyyy**.

Présentation des jours libres

Nom	Date	Créneaux libres
WE Accueil	09/20/2017	2
Forum entreprise SIP pas cours ce jour	11/16/2017	2
Nuit de l'info	12/08/2017	2

Utilisation

Pour utiliser le générateur de fond de planning, il faut que [Java JRE 8](#) soit installé.

La commande pour utiliser le générateur de fond de planning est :

```
java -jar planning-generator.jar < annéeEtude > < -di3 | -di4 | -di5 | -all > < fichierMaquette > < fichierCalendrier >
```



Tous les paramètres sont **obligatoires**.

1. < **annéeEtude** > ⇒ Ce paramètre est la année d'étude "**annéeDébut/annéeFin**". (Doit être séparé par un /)
2. < **-di3 | -di4 | -di5 | -all** > ⇒ Le second paramètre permet de choisir pour quelle niveau va être généré le fond de planning.
3. < **fichierMaquette** > ⇒ Ici doit être renseigné le chemin d'accès complet au **fichier maquette**.
4. < **fichierCalendrier** > ⇒ Le dernier paramètre permet de renseigner le chemin complet du **fichier calendrier**.

Nous avons ajouté une commande permettant d'avoir de l'aide.

Commande d'aide

```
java -jar planning-generator.jar -h
```

Le fichier généré portera le nom suivant :

```
Planning Année <niveau> DI <annéeDébut> - <annéeFin>.xlsx
```



Les fichiers générés se trouveront au même endroit que le **jar**.

Création de fond de planning

Résumé

Le programme devra créer un fond de planning en Excel afin de faciliter la création des emplois du temps. Grossièrement, le fond de de planning se présente comme un quadrillage avec en ligne les cours (CM, TD, TP) et en colonne les semaines. Des formules de calculs sont ajoutées à certaines cases afin de comptabiliser les heures placées. Le programme prendra en entrée un fichier excel (ou au pire csv) contenant une maquette et les enseignants affectés. Il fournira en sortie un fichier excel.

Mots-clés

Java, Excel, Apache POI

Abstract

The program will need to create an Excel planning fund to facilitate the creation of timetables. Roughly speaking, the planning background is presented as a grid with on-line courses (CM, TD, TP) and in columns during the weeks. Calculation formulas are added to certain boxes in order to count the hours placed. The program will take an excel (or at worst csv) file containing a mock-up and the affected teachers as input. It will output an excel file.

Keywords

Java, Excel, Apache POI

Tuteur académique
Christophe LENTÉ

Étudiants
Hanyuan PENG (DI4)
Stéphane DELUCE (DI4)