	<b>Programmation Java : les bases</b>  Projet de NFA 031  2014 - 2015	18/12/2015  Par J. LAFORGUE
---	---	--------------------------------------

CONSERVATOIRE  
 NATIONAL  
 DES ARTS  
 ET METIERS  
  
 CENTRE REGIONAL  
 MIDI - PYRENEES

**NFA 031 – Travaux pratiques**

*IPST-  
 CNAM  
 Licence*

**PROJET**

Années 2014-2015  
 TOULOUSE  
 J. LAFORGUE

## 1. LE BESOIN

On se propose de programmer un agenda électronique.  
 Un agenda est une classe Agenda qui permet de gérer des rendez-vous.  
 Un rendez-vous est un objet de la classe RendezVous.

A terme le programme devra être capable de :

- [FUNC 1] lire un fichier texte contenant les rendez-vous et les mémoriser dans le programme
- [FUNC 2] afficher tous les rendez-vous
- [FUNC 3] afficher les rendez-vous d'un jour (date) donnée
- [FUNC 4] naviguer (jour suivant, jour précédent, date courante)
- [FUNC 5] gérer des notes (chose à faire) (une note est un rendez-vous sans date et sans heures, ou sans heure)
- [FUNC 6] ajouter un nouveau rendez-vous
- [FUNC 7] supprimer un rendez-vous

Il n'est bien sur pas envisager de tout développer en une seule fois. Nous allons réaliser ce programme en 3 étapes.



## 2. ETAPE 1

Pour démarrer votre programme vous partez du code fourni (projet.zip) dans lequel vous trouvez :  
une classe **Projet** avec une méthode main qui crée l' **Agenda** et l' **IHMAgenda**.  
L'IHMAgenda prend en entrée l'Agenda.

Dans cette étape, on se propose de réaliser les fonctionnalités [FUNC 1] [FUNC 2] et [FUNC 3]

Le constructeur de la classe Agenda doit lire un fichier texte dont chaque ligne est un rendez-vous (Vous utilisez la méthode `String[] Terminal.lireFichierTexte(String nomFichier);` )

Cette ligne est de la forme :

**date;heure;heure\_fin;libellé;commentaire**

avec date au format JJ/MM/AAAA

avec heure et heure\_fin au format HH:MM

Exemple :

12/12/2014;10:30;11:00;mèdecin;ne pas oublier d'apporter la radio

14/01/2015;14:00;17:00;réunion d'avancement;

Pour chaque ligne lue, un rendez-vous est créé (`new RendezVous(...)`). On passe en paramètre du constructeur de `RendezVous` les champs lus. Puis ce rendez-vous est ajouté dans une collection (`ArrayList`) de l'Agenda.

La classe **IHMAgenda** crée un formulaire dans lequel :

- un bouton permettant d'afficher tous les rendez-vous dans une zone vertical de résultat (chaque rendez-vous est affiché sous la forme d'une chaîne de caractère)
- un texte de saisi et un bouton permettant d'afficher tous les rendez-vous d'une date donnée.

Pour réaliser ces deux actions, écrivez dans la classe Agenda les deux méthodes suivantes :

**public String getTousRendezVous()**

Cette méthode retourne sous la forme d'une chaîne de caractère, tous les rendez-vous (une boucle qui appelle la méthode `toString` de chaque agenda)


**public String getDateRendezVous(String date)**

Cette méthode retourne sous la forme d'une chaîne de caractère, tous les rendez-vous dont la date est égale à celle passée en paramètre.

Ces deux méthodes sont appelées dans la méthode **submit** de la classe IHMAgenda.

Dans cette étape vous gérez la date d'un rendez-vous comme une simple chaîne de caractère (`String`). nous verrons que à partir de l'étape 2 il faudra gérer cette date sous la forme d'un **Calendar**.

Nous supposons que le fichier texte ne contient pas d'erreur de syntaxe: pas de "point-virgule" en moins, la date et l'heure sont correctement écrite.

	<b>Programmation Java : les bases</b>  Projet de NFA 031  2014 - 2015	18/12/2015  Par J. LAFORGUE
--	---	--------------------------------------

### 3. ETAPE 2

Dans cette étape on se propose de réaliser les fonctionnalités [FUNC 4] [FUNC 5].

Au sujet du format du fichier :

**date;heure;heure\_fin;libellé;commentaire**

Si le champ **date** est absent cela signifie que le rendez-vous est tous les jours.

Si le champ **date** est égal à "**hebdo**" alors le rendez-vous est tous les jours sauf le samedi et le dimanche.

Si le champ **date** et **heure début** sont absent alors il s'agit d'une note mémo (quelque chose à faire)

Si le champ **heure** est absent alors il s'agit d'une note mémo pour la date précisée.

L'heure de fin **heure\_fin** peut être vide.

L'IHM contient 3 boutons : **Suivant**, **Courant**, **Precedent** :

- Le bouton **Courant** affiche tous les rendez-vous de la **date courante**. Cette date est initialisée dans le constructeur de Agenda. La première ligne du fichier texte est au format JJ/MM/AAAA et correspond à la valeur de cette date.
- Le bouton **Suivant** change la **date courante** en la passant à la date suivante et affiche les rendez-vous de cette date.
- Le bouton **Precedent** change la **date courante** en la passant à la date précédente et affiche les rendez-vous de cette date.

La classe Agenda gère donc une information : **dateCourante**.

Il faut que quand on affiche les rendez-vous ils s'affichent dans **l'ordre chronologique**.

~~Pour gérer facilement la date d'un rendez-vous, il faut que la classe RendezVous ne contienne plus la date sous la forme d'une chaîne de caractère mais sous la forme d'un **Calendar**.~~

Pour cela, une explication en séance sera faite pour l'utilisation de cette classe Calendar.

Voir l'exemple **Exemple40\_Calendar** sur le site.



## 4. ETAPE 3

Dans cette étape on se propose de réaliser les fonctionnalités [FUNC 6] et [FUNC 7].

1/

On se propose d'ajouter un nouveau rendez-vous dans l'agenda : saisie de tous les champs du rendez-vous.

Pour cela vous devez créer une nouvelle classe **IHMajouterRendezVous** dont le constructeur prend en entrée l'agenda et qui crée une nouvelle IHM permettant de saisir tous les champs d'un rendez-vous. Un bouton Valider permet de valider sa saisie : création d'un rendez-vous et ajoute ce rendez-vous dans l'agenda. Puis la fenêtre se ferme.

Par défaut le champ de saisie de la date du rendez-vous est initialisé avec la date courante.

Vous devez ajouter dans l'IHM de l'agenda un bouton **Ajouter** qui crée l'ihm **IHMajouterRendezVous**.

2/

On se propose de supprimer un rendez-vous dans l'agenda. Pour cela, il faut que tous les rendez-vous soient numérotés de manière unique (compteur).

Dans l'IHM de l'agenda on ajoute un texte de saisie du numéro et un bouton **Supprimer**.

Si le numéro n'est pas valide alors on affiche un texte d'erreur comme résultat dans l'IHM.


## 5. QUELQUES PRECISIONS

Les règles de codage que vous devez suivre sont :

- sauf pour les constantes, les attributs de toutes les classes doivent être privés
- toutes les classes commencent par une majuscule
- tous les attributs, paramètres et variables commencent par une minuscule
- les noms de tous les répertoires de package sont en minuscule
- les packages créés commencent tous par "fr.cnam"
- une indentation stricte est respectée dans tout le code
- un minimum de commentaire est nécessaire afin de comprendre ce que fait le code. Surtout quand ce que fait le code n'est pas décrit précisément dans le sujet.

Conseils :

- Ecrivez peu de code à chaque fois afin d'avoir toujours un code qui se compile correctement
- Après chaque étape ou sous-étape, le programme s'exécutant correctement, faites une sauvegarde des sources. Cela permet de pouvoir revenir en arrière si besoin et permet aussi de pouvoir donner à l'enseignant une version du programme qui s'exécute correctement et qui remplit correctement une partie des fonctionnalités demandées dans le sujet.
- Dans un premier temps ne faites pas plus que le sujet ne le demande. Une fois que le programme respecte le sujet, vous faites une sauvegarde et vous pouvez apporter des améliorations qui sont toujours appréciées par le correcteur, à condition qu'elles soient bien commentées.

	<b>Programmation Java : les bases</b>  Projet de NFA 031  2014 - 2015	18/12/2015  Par J. LAFORGUE
--	---	--------------------------------------

## 6. ARCHITECTURE DES REPERTOIRES

L'architecture suivante est une proposition de création des répertoires et des fichiers pour réaliser votre projet. Cette architecture doit être respectée.

projetNFA031

```

data
  Agenda.txt
src
  fr
    cnam
      projet
        Projet.java (main)
        Agenda.java
        IHMAgenda.java
        RendezVous.java
        IHMAjouterRendezVous.java
      ihm
        Formulaire.java
        FormulaireException.java
        FormulaireInt.java
        ....
      util
        Terminal.java
        TerminalException.java
        ...
bin
  <les .class>
  
```

Les classes en rouge sont des classes qui vous sont données.

Le répertoire **data** contient le fichier contenant une liste de rendez-vous initiaux (cela permet de ne pas saisir les rendez-vous à chaque exécution). Le nombre de rendez-vous ne doit pas être trop grand mais doit être représentatif des traitements demandés dans le projet.

## 7. PLANNING

**Vous avez 3 séances de TP pour réaliser ce projet**

Le projet sera ramassé 1/4 heure avant la fin de la dernière séance de TP (le temps de les ramasser tous).

L'enseignant récupère le projet sur une clef USB en salle de TP.

Il est donc **impératif** que, au moins un membre du groupe de TP, soit présent lors de la dernière séance afin de donner son projet.