CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL MIDI - PYRENEES

NFA 032– Travaux pratiques

IPST-CNAM Licence

PROJET

Années 2014-2015
TOULOUSE
J. LAFORGUE

1. INTRODUCTION

Ce projet est la suite du projet réalisé cette année dans le cours NFA 031.
Deus solutions :

- vous commencez avec le projet que vous avez réalisé. Dans ce cas, il faut qu'il contienne toutes les fonctionnalités demandées dans le sujet de NFA 031
- vous commencez avec ma correction qui est disponible sur le site du cours NFA 031 année 2014-2015 (http://jacques.laforgue.free.fr/SITE_NFA031/Site/Projets.phtml)

2. ETAPE 1

Il s'agit dans cette étape de changer l'implémentation de l'architecture des classes de définition d'un rendez-vous; le fonctionnement du programme reste le même.

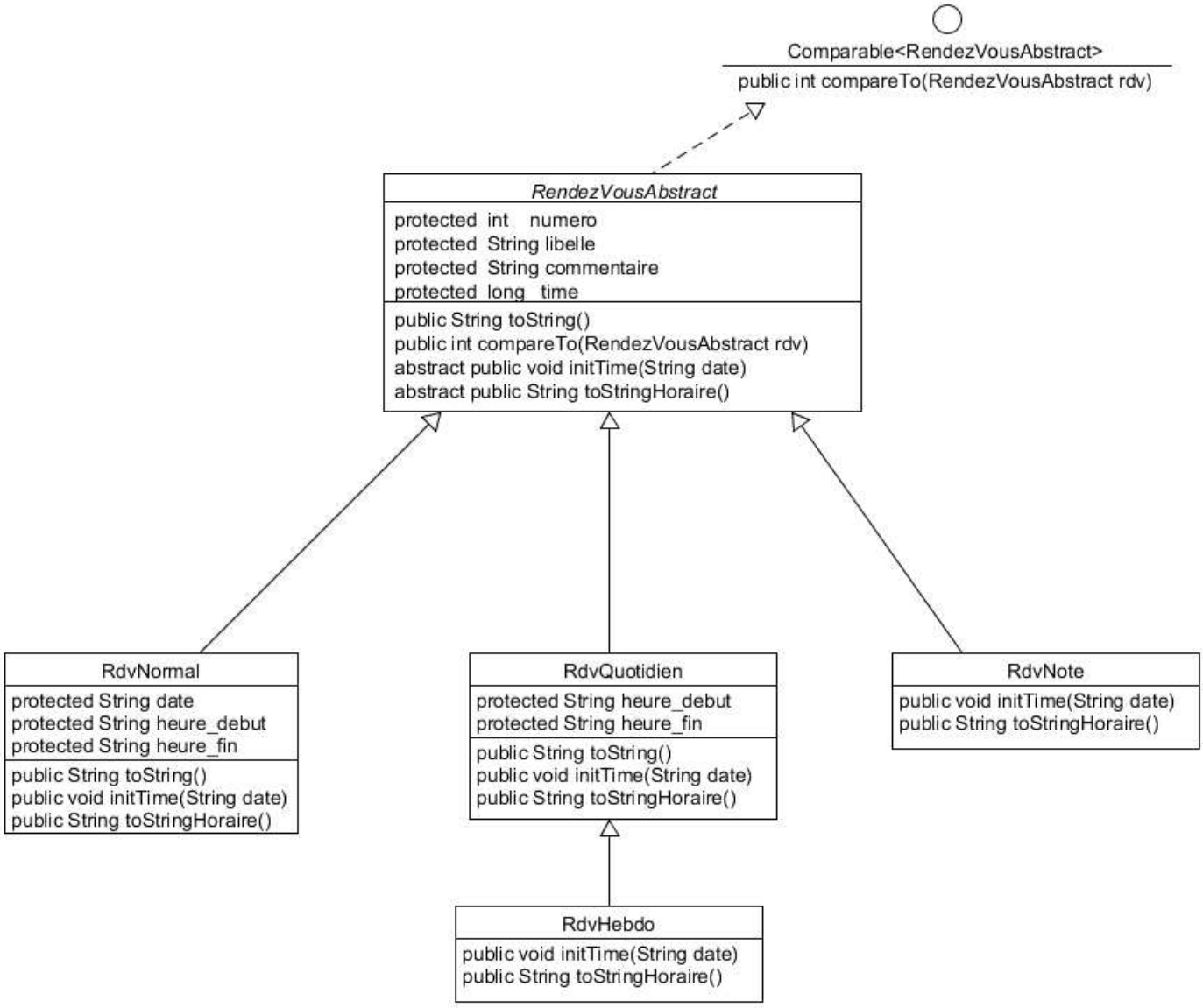
Nous avons vus qu'il existe différents types de rendez-vous, il s'agit de faire autant de classes concrètes que de types de rendez-vous. Toutes ces classes héritent d'une classe abstraite qui contient tous les attributs, méthodes communes et méthodes abstraites.

La classe abstraite s'appelle : **RendezVousAbstract**

Les classes concrètes s'appellent :
RdvNormal pour définir un rendez-vous qui a une date et une heure de début
RdvQuotidien pour définir un rendez-vous qui est tous les jours : sans date et avec une heure début
RdvHebdo pour définir un rendez-vous qui est tous les jours sauf le week-end : date="hebdo" et avec une heure début
RdvNote pour définir une note : un rendez-vous qui n'a pas de date et pas d'heure de début

Tous les attributs de ces classes sont protected.

On veut l'arborescence d'héritage suivante (en partant de ma correction) :



La classe Agenda :

Afin de vous guider, les modifications qu'il faut réalisées dans cette classe sont :

- Dans la méthode **ajouterRdv**, il faut créer chaque rendez-vous avec la bonne classe en fonction des champs lus depuis le fichier.
- Dans la méthode **getDateRendezVous(String dateRdv)**, il faut :
  - initialiser l'attribut ***time*** en fonction de dateRdv en utilisant la méthode initTime(String date) qui doit être implémentée dans chaque classe de rendez-vous concrètes.
  - ajouter le rendez-vous dans le tableau temporaire si time != 0
  - trier le tableau temporaire sur le critère de time Collection.sort(List)
  - utiliser lé méthode abstraite toStringHoraire pour construire le résultat
- supprimer la méthode toStringHoraire de la classe Agenda

3. ETAPE 2

1/ La lecture du fichier texte devient une commande d'import exécutée quand on clique sur un bouton.
S'il y a une erreur dans le décodage des lignes alors l'erreur est remonte dans l'IHM.
On saisit le nom du fichier d'import dans un texte de saisi, par défaut "data/Agenda.txt".

2/ Dans le constructeur, on remplace la lecture du fichier texte par la lecture d'un fichier binaire. Le fichier binaire contient :

- un objet String qui est la date courante
- le nombre de rendez-vous (Integer)
- les rendez-vous

Tous ces objets sont écrits en utilisant ObjectOutputStream et lus en utilisant ObjectInputStream.
On ajoute un bouton permettant de sauver l'agenda dans ce fichier.
Le fichier s'appelle data/AGENDA.bin.
Si le fichier n'existe pas alors pas d'erreur. Il sera créé lors de la sauvegarde.

4. ETAPE 3

On crée un nouveau **programme java** (main) (Client) qui est une IHM qui contient :

- un texte de saisi pour saisir une date et un bouton Valider qui demande les rendez-vous de la date saisie
- 2 boutons PREC SUIV qui permet de demander les rendez-vous de la date précédente ou suivante de la date courante
- une zone de texte pour afficher le résultat.

L'IHM gère en local une date courante.
Cette date est initialisée au démarrage de l'IHM en demandant la date courante de l'agenda.

L'IHM réalise une communication socket avec l'agenda..
Les échanges sur le socket se font DataOuputSream/DatainputStream (readUTF writeUTF, readInt, writeInt, ...).

On définit deux requêtes :

- la requête qui demande la date courante :
  - envoi: "GET_DATE_COUR"
  - reçu: *date*
- la requête qui demande les rendez-vous d'une date :
  - envoi: "GET_RDVS_COUR" *date*
  - reçu *resultat*

"chaine" est de type UTF
date est de type UTF
resultat est de type UTF

Il est important de respecter la syntaxe et le type des informations sur le socket afin que l'on puisse utiliser les agendas entre les groupes de TP de la salle de TP.

Le programme client prend en entrée le nom d'host de la machine qui a un agenda et le port utilisé.

Le programme agenda prend en entrée le port utilisé.

En salle de TP, le port doit être entre 9100 et 9110. Vous prenez le port **9100**.

5. 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
  - AGENDA.bin
- **src**
  - **fr**
    - **cnam**
      - **projet**
        - Projet.java (*main*)
        - Agenda.java
        - IHMAgenda.java
        - IHMAjouterRendezVous.java
        - ~~RendezVous.java~~
        - RendezVousAbstract.java
        - RdvNormal.java
        - RdvQuotidien.java
        - RdvHebdo.java
        - RdvNote.java
        - ServeurSocketAgenda.java
        - IHMClient.java (*main*)
      - **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.
Les classes en bleu sont les classes du projet de NFA031.

6. 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.

7. EXTRA

Une fois le projet terminé est mis de côté afin que vous me donniez ce qui a été demandé dans le sujet, vous pouvez modifier l'IHM cliente afin qu'elle devienne l'IHM complète de l'agenda.
Le programme agenda n'a alors plus d'ihm et devient un serveur d'agenda.