

IFT 1020

Programmation II

Mélissa Jourdain

June 17, 2004

Travail pratique #3

L'héritage et les collections

1 Explications générales

Équipe: Le travail est à faire en binôme ou en monôme, mais pas plus de deux.

Remise: Deux remises à effectuer : Électronique le **5 juillet à 22h30** au plus tard, sans possibilité de retard et papier au début du cours le mardi le **6 juillet 2004**.

Conseil: N'attendez pas à la dernière minute pour faire votre travail, on ne sait jamais les surprises qui nous attendent quand on programme! Aussi, lisez attentivement l'énoncé et prenez bien conscience du problème avant de vous lancer dans la programmation.

2 But

Ce travail a pour but de vous familiariser avec le principe de l'héritage et aussi avec l'utilisation des collections de Java. Aussi, il vous permettra de vous assurer de vos connaissances en terme d'interface graphique.

3 Énoncé

3.1 Mise en situation

Vous avez un système d'organisation d'horaire à implanter pour une clinique de médecins. L'utilisateur entre, via l'interface graphique, les différentes informations concernant un patient, et votre programme doit lui attribuer un rendez-vous qui répond le plus possible au besoin du patient.

3.2 Description de la clinique

Il y a trois médecins dans la clinique, tous spécialistes dans un domaine différent: un généraliste, un dermatologue et un cardiologue. Ainsi, vous aurez trois grilles horaires à remplir et un patient est placé dans une des grilles horaires dépendamment du type de soins qu'il nécessite.

Pour simplifier le problème, vous pouvez considérer que vous remplissez la grille horaire pour la semaine qui suit. Ainsi, vous prenez pour acquis que le rendez-vous le plus près

possible dans le temps, est le lundi matin. Quand la semaine dont vous construisez l'horaire est remplie, vous pouvez commencer à attribuer des rendez-vous pour la semaine suivante. Pour chacun des patients, un seul rendez-vous par médecin peut être pris à la fois. Par exemple, un patient peut prendre un rendez-vous avec un dermatologue et un autre avec le médecin généraliste, mais il ne peut demander deux rendez-vous avec le dermatologue.

La clinique est ouverte du lundi au vendredi de 8hres à 16hres. Les rendez-vous sont chacun d'une durée d'une demie-heure et les médecins prennent une heure de dîner de midi à 13hres tous les jours. Il y a donc 14 rendez-vous possibles dans une journée.

3.3 Description des patients

Un patient est décrit par certaines informations personnelles:

1. Son nom
2. Son numéro d'assurance maladie
3. L'urgence de son cas (qui indique la priorité qu'il a par rapport aux autres patients)
4. Le(s) jour(s) préféré(s) pour se faire soigner(facultatif). Dans le cas où un patient n'a pas de préférence pour une journée, on le place selon l'urgence de son cas dans la grille.
5. Le type de soins qu'il doit recevoir, ce qui détermine à quel spécialiste de la clinique on le réfère.

À partir de ces informations, vous pouvez placer les patients en ordre de priorité, ce qui vous permet de construire la grille horaire. La représentation interne de ces champs est faite selon votre inspiration. Il faut cependant que vous expliquiez vos choix dans le rapport.

3.4 Contraintes de priorité des patients

Tel que décrit précédemment, certaines informations sur le patient font que l'on peut construire un ordre de priorité entre les patients. En effet, un patient dont la maladie est plus urgente à traiter qu'un autre aura un rendez-vous plus rapidement. Il est à noter que même si ce n'est pas vraiment réaliste, on peut déplacer le rendez-vous des patients, pour pouvoir traiter un cas plus urgent plus rapidement. De cette façon, l'ordre des patients dans la grille horaire respecte toujours l'ordre de priorité.

Pour déterminer la journée à laquelle un patient a son rendez-vous, on peut se baser sur sa(ses) préférence(s), et on essaie autant que possible de le placer durant la journée qu'il préfère. Par exemple, un patient qui veut avoir un rendez-vous le mercredi, a priorité sur les patients qui sont le mercredi et qui n'ont pas demandé cette journée.

Ces indications sur les priorités vous donnent une idée de la façon dont le problème doit être traité. Mais vous aurez sûrement à faire certains choix d'implantation concernant la priorité lors de la programmation. Il est très important de les commenter et d'expliquer vos choix dans votre rapport.

Le plus important est que la façon dont les patients sont disposés dans la grille horaire fasse en sorte que l'ordre de priorité des patients soit respectée.

3.5 Services offerts par l'interface graphique

Votre interface graphique doit assurer certains services.

1. **L'ajout d'un nouveau patient** Quand un patient est ajouté, un rendez-vous lui est attribué automatiquement dans la bonne grille horaire. On doit pouvoir entrer les informations sur le patient via l'interface graphique soit son nom, son numéro d'assurance sociale...
2. **L'annulation d'un rendez-vous** Un patient peut annuler un rendez-vous. Attention, ici vous devez déplacer les autres patients de façon à bien remplir de nouveau la grille horaire de façon à combler le mieux possible la plage laissée vide par cette annulation et aussi de façon à conserver l'ordre de priorité des patients.
3. **La visualisation de la grille horaire** On peut voir la grille horaire d'un des médecin à partir de l'interface graphique. [indice: Jetez un coup d'oeil à la classe JTable dans l'API]

3.6 Contrainte d'implantation

Les structures de données dont vous aurez besoin ne seront pas dans l'API directement. Pour vous aidez à représenter la grille horaire, vous devrez implanter une **file de priorité**. Cette structure de données ressemble étrangement à une liste à l'exception du fait que le premier élément de la liste, est toujours le plus prioritaire, et le dernier le moins prioritaire. D'autre moyen d'implanter une file de priorité existe, mais vous pouvez vous en tenir à l'implantation la plus simple soit sous forme de liste.

Heureusement, vous n'aurez pas à réimplanter complètement une structure de données. Vous devez avoir recours à l'héritage, en faisant hériter votre structure de données par une autre déjà implantée dans l'API, ce qui va alléger de beaucoup l'implantation.

Vous devez commenter vos choix par rapport aux structures de données choisies dans votre rapport.

Suggestion: plusieurs types de patients existent (au moins trois types, un associé à chacun des médecins). Vous pouvez avoir recours à l'héritage ici aussi.

Une attention particulière aux model-view-controller sera donnée lors de la correction!!

3.7 Mot de la fin

Il est très important de se prendre en avance pour ce devoir. De plus, ne vous lancez pas dans la programmation tout de suite après avoir lu votre TP. Prenez le temps de vous faire un plan! C'est la suggestion du jour, et elle est primordiale à suivre!

4 Remise

Il est important de noter que votre TP sera exécuté par les démonstrateurs sur une machine Linux avec le compilateur jdk (1.4.2.01). Si par choix vous décidez d'utiliser un autre compilateur, vérifiez que le code que vous avez produit (qui normalement fonctionne correctement chez vous) fonctionne bien sur les ordinateurs du DIRO. Avant de faire cela, assurez vous d'abord que vous avez la bonne version de jdk, en utilisant pour cela la commande : "java -version" devra donner le numéro de version "1.4.2_01". Pour éviter d'oublier un des fichiers lors de la remise, nous vous conseillons de les regrouper dans un seul fichier compressé tp3.zip et de ne remettre que ce dernier. La remise comprend 3 (TROIS) choses :

1. Envoyez votre fichier "tp3.zip" par la procédure de remise électronique habituelle (Pour obtenir de l'aide sur cette commande, tapez dans un xterm : man remise). Respectez les noms des fichiers. Cette remise est due pour le **5 juillet avant 22h30**

remise dift1020 tp3 tp3.zip

2. Remettez une copie papier d'un rapport qui devrait décrire votre programme (pour le contenu d'un rapport voir la FAQ sur la page web du cours). Cette remise doit être faite le **6 juillet 2004**, au début du cours.
3. N'oubliez pas de remettre avec votre rapport une copie papier de votre programme Java.

5 Barème

Ce TP3 est noté sur 14 points.

1. La non remise électronique (volontaire ou par erreur) est sanctionnée par la note 0.
2. La non remise papier vous pénalise de 1 point.
3. Les programmes ne contenant pas d'en-tête, -1 point.
4. Un programme qui ne compile pas : 0/14
5. Un programme qui compile mais ne fait pas les choses prévues dans la spécification : 0/14
6. Le code doit être lisible, bien indenté et commenté.

6 Des questions à propos de ce TP?

L'adresse email: dift1020@iro.umontreal.ca

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre email, au moins la chaîne:

“[IFT1020]”

7 Mise à jour

16-06-2004 diffusion