

# Épreuve de Langages Orientés Objets

## Licence d'informatique

Catherine Dezan, Bernard Pottier

Mai 2003

### 1 Modélisation

Une agence propose de réserver des vols organisés par différentes compagnies aériennes. La réservation peut concerner un ou plusieurs vols en fonction de la liaison demandée (cas de vol direct ou non). La recherche d'un vol pour un client peut s'effectuer si l'on connaît la ville d'arrivée, de départ et la date du départ. Le résultat des investigations peut renvoyer une liste de vols en donnant leur durée, leur prix et en détaillant le parcours pour les vols non directs. Pour qu'un vol non direct soit proposé, il faut que la durée pour la connexion, soit la durée d'attente entre l'arrivée du vol précédent et le départ du vol suivant, soit minimum de trois quarts d'heure si l'arrivée et le départ sont effectués dans le même aéroport de la ville concernée par la connexion et de deux heures au moins si le départ est effectué dans la même ville mais dans des aéroports différents (par exemple, l'aéroport de Roissy et l'aéroport d'Orly à Paris). Pour des soucis d'efficacité, la recherche des vols non directs est limitée à deux connexions.

Le coût d'un vol qui est proportionnel à la durée de ce vol, dépend non seulement de la compagnie qui organise le vol mais aussi des places disponibles lors de la réservation (économique et affaire) et du statut du client (enfant, senior, abonné).

1. Définir les classes nécessaires pour modéliser ce système de réservation sous forme de diagramme de classes.
2. Définir une méthode Smalltalk permettant de rechercher les vols pour la liaison demandée par le client. Vous prendrez bien soin de détailler les méthodes annexes ~~que~~ vous avez besoin notamment pour vérifier que les durées des connexions soient bien respectées.
3. Définir une méthode permettant de calculer le coût total d'un vol correspondant à la liaison aérienne demandée par le client.
4. Définir une méthode Smalltalk permettant de trier les vols sélectionnés selon le critère de coût ou selon le critère de durée (durée totale incluant les durées d'attente pour les connexions) en précisant si le vol est direct ou non.
5. Les réservations faites par l'agence sont sauvegardées dans un fichier. Définir la méthode Smalltalk permettant cette sauvegarde.
6. Le client souhaite annuler une réservation. Définir la ou les méthodes permettant d'effectuer cette annulation.

### 2 Mémoire d'objets

- Pour chaque instruction du code suivant, décrivez les objets qui sont créés et détruits automatiquement.
- Donnez le bilan global de la méthode, et expliquez clairement ce qu'il advient des objets référencés au travers des variables `r`, `streamOnTable`.
- Énoncez le principe des ramasse-miettes à comptage de référence, et l'apport de ce mécanisme au niveau de la programmation.

uneMethode

```
| streamOnTable table r |  
streamOnTable := (Array new) writeStream.
```

```

2 streamOnTable nextPut: (r := Rectangle origin: 1@2 corner: 2@4).
3 streamOnTable nextPut: (r scaledBy: 2).
4 streamOnTable nextPut: r .
5 table := streamOnTable contents.
6 ((table at: 2) == table first) ifTrue: [
    table at: 1 put: nil.
    table at: 3 put: nil.].
^ table

```

### 3 Structure des classes

On crée deux classes dans le but d'effectuer des présentations graphiques 2D associant des valeurs en ordonnées à des valeurs en abscisses.

1. NuageDePoints comporte deux variables `lesX` et `lesY`, toutes deux sont des collections séquençables contenant des nombres.
2. FonctionSurUnIntervalle, hérite de NuageDePoints, et comporte une variable d'instance `fonction`, qui retient le *texte d'une méthode* permettant de calculer une valeur de `y` à partir d'une valeur de `x`.

#### 3.1 héritage

On crée les méthodes d'accès aux variables dans ces deux classes.

- Donnez un diagramme présentant les relations existant entre une nouvelle instance de la classe `FonctionSurUnIntervalle`, la classe `NuageDePoints`, le dictionnaire des méthodes de `NuageDePoints`, la méthode compilée associée au sélecteur `lesX`.
- Expliquez ce qui va se passer si un objet de la classe `FonctionSurUnIntervalle` reçoit le message `lesX`.

#### 3.2 code dynamique et compilation

Expliquez comment on peut rendre opérationnel un objet de la classe `FonctionSurUnIntervalle` créé par une méthode `fonction : texteDeLaFonction valeurs : intervalle`.

L'objet doit disposer d'un nuage de points complètement défini (`lesY`), et le texte de la fonction est donné par l'utilisateur.

#### 3.3 structure des programmes

Donnez l'arbre de programme que le compilateur génère pour la méthode qui suit :

```

squaredPlusOne: x
| y |
y := x * x + 1.
^y

```