UNIVERSITE BORDEAUX I — **Maîtrise d'informatique**

Année 1991-1992 — **3 février 1992**

Modèles et techniques de programmation

Durée: 2 heures

Tous documents autorisés

Première partie

1) On considère la définition SCHEME classique de la fonction factorielle:

```
(define (fact n)
  (if (= n 0)
      1
      (* n (fact (- n 1)))))
```

Expliquer comment est évaluée l'expression

$$(\langle LET \rangle \ \texttt{((* +))} \ \texttt{(fact 5))}$$

selon que $\langle LET \rangle$ est une liaison statique ou dynamique.

2) Définir, en utilisant le mécanisme de liaison dynmamique, l'opérateur `trace2` tel que l'expression

$$\texttt{(trace2} \ \langle opbin \rangle \ \langle exp \rangle\texttt{)}$$

affiche lors de son évaluation la trace de chaque appel à l'opérateur binaire $\langle$*op-bin*$\rangle$ dans l'évaluation de $\langle exp \rangle$, et dont la valeur est la même valeur que celle de $\langle exp \rangle$.

Exemples

```
> (trace2 * (fact 3))
 call: (* 1 1)
 call: (* 2 1)
 call: (* 3 2)
6
> (trace2 = (fact 3))
 call: (= 3 0)
 call: (= 2 0)
 call: (= 1 0)
 call: (= 0 0)
6
>
```

```
> (trace2 = (trace2 - (trace2 * (fact 3))))
 call: (= 3 0)
 call: (- 3 1)
 call: (= 2 0)
 call: (- 2 1)
 call: (= 1 0)
 call: (- 1 1)
 call: (= 0 0)
 call: (* 1 1)
 call: (* 2 1)
 call: (* 3 2)
6
>
```

3) On considère la fonction

```
(define (f x K)
  (if (even? x)
      (K x)
      (K (* 2 x))))
```

Expliquer ce que produit l'évaluation des expressions

```
(f 3 (lambda (x) x))

(call/cc (lambda (K) (trace1 K (f 3 K))))
```

L'opérateur `trace1` est l'analogue de l'opérateur `trace2` pour les fonctions d'arité un.

Deuxième partie

On considère une application interactive, développée selon une approche *programmation par objets*, et on veut définir une interface graphique permettant d'effectuer les envois de message de la façon suivante:

- chaque objet instancié possède une représentation dans une fenêtre de l'écran;
- un envoi de message à un objet se fait en *"cliquant"* sur cet objet avec la souris; l'objet affiche alors la liste des messages de sa classe sous la forme d'un menu, permettant de sélectionner le message voulu.

On supppose que le langage de programmation utilisé est un langage à typage statique. Pour présenter les exemples de programmes on pourra utiliser, soit un langage à objets à typage et liaisons statiques, comme C++ ou Eiffel, soit le langage C.

On suppose également qu'il existe une classe `FENETRE`, et que chaque fenêtre sur l'écran correspond à une instance de cette classe. Le fait de sélectionner un point de coordonnées `(x,y)` dans une fenêtre `F` en *"cliquant"* avec la souris se traduit par un envoi de message de la forme

```
send(F, click, x, y)
```

1) Décrire le traitement effectué par la méthode recevant ce message pour activer l'objet *cliqué*.

2) Décrire une classe `MENU` affichant une liste de choix et retournant le numéro du choix sélectionné (on supposera qu'une fenêtre sait afficher et effacer une boîte contenant une chaîne de caractères).

3) Un objet activé doit afficher un menu composé de la liste de ses messages puis appeler la méthode sélectionnée. Décrire l'implémentation d'une classe `MENU_DE_MESSAGES` réalisant ce traitement, et dont héritent toutes les classes d'objets ayant une représentation sur l'écran.

4) La fonction d'instanciation d'une classe dérivée de `MENU_DE_MESSAGES` doit effectuer correctement l'initialisation du menu des messages de l'objet instancié. Décrire une implémentation de ce mécanisme.

5) Décrire un mécanisme permettant à tout moment d'annuler le traitement en cours.

6) Comment peut-on implémenter une instanciation interactive des objets au moyen de la souris. Discuter les problèmes que cela pose selon que les classes sont ou non des constructions de *"première classe"*.