Université Bordeaux 1, Sciences et Technologies
UFR Informatique et Mathématiques

Maîtrise d'Informatique
Session de janvier 2000

Techniques et Fondement de la Programmation
(partie "techniques")

Durée : trois heures (les deux parties)

Tous documents autorisés sauf photocopies illégales de livres. Votre voisin n'est pas un document.

Rappel : Soyez bref et clair ! Personne n'est impressionné par la quantité !

Bon courage !

Exercice 1. (7p)

Identifier les problèmes du programme suivant, et écrire une version améliorée :

```
; This program returns a list of the elements of l for which
; the predicate f returns nil, and the number of elements of l
; for which f returns true
(defun filter (l f)
       (setq retval nil)
       (do ((rest l (cdr rest))
            (i 0))
           ((null rest))
         (if (funcall f (car rest))
             (setf i (+ i 1))
             (setq retval (append retval (list (car rest))))))
       (values retval i))
```

Exercice 2. (7p)

Un éditeur graphique (de type xfig) maintient une suite d'objets graphiques. Dans notre cas, l'éditeur contient des opérations d'édition qui modifient toujours le premier objet de la suite.

Afin de modifier un objet autre que le premier, on doit d'abord l'échanger avec le premier. Pour cela, on dispose d'une variable globale `*top-object*` contenant le premier objet de la suite (les opérations d'édition utilisent la valeur de cette variable), et d'une opération `(swap-objects obj1 obj2)` pour échanger deux objets de la suite.

On voit alors souvent du code de ce type

```
(swap-objects *top-object* obj)
(edit-operation-1 ...)
(edit-operation-2 ...)
...
(swap-objects *top-object* obj)
```

Écrire une macro Common Lisp `with-object` qui capture cet idiome.

Exercice 3. (6p)

Dans un système de gestion de diplômes, on doit gérer des étudiants et des diplômes de types divers. Afin de factoriser au maximum le code, on souhaite utiliser des classes, et des fonctions génériques CLOS.

Écrire les définitions (sans les créneaux) des classes `etudiant`, `etudiant-erasmus`, ainsi que les classes pour différencier les étudiants masculins des féminins.

Écrire les définitions (sans les créneaux) des classes `diplome`, `diplome-bac+4`, `diplome-bac+5`, `diplome-professionnel`, et `diplome-DESS` (Le DESS est un diplôme bac+5 professionnel).

Écrire la déclaration d'une fonction générique `calculer-note-finale`, pour calculer la note finale d'un étudiant inscrit dans un diplôme.

Écrire l'en-tête (nom, paramètres) des méthodes pour `calculer-note-finale` permettant le calcul pour un étudiant non Erasmus, et un étudiant Erasmus, inscrits dans un diplôme professionnel, et un diplôme DESS.