

**FSAC1450 – Informatique T4 – Examen Final**  
**Prof. Peter Van Roy**  
**1 décembre 2004**

|                  |  |
|------------------|--|
| Nom              |  |
| Prénom           |  |
| Matricule (NOMA) |  |

**Question 1 (5 pts)**

Pour cette question, écrivez un programme déclaratif qui prend deux listes d'entiers,  $x_s = [x_1 \ x_2 \ \dots \ x_n]$  et  $y_s = [y_1 \ y_2 \ \dots \ y_n]$ , et qui calcule leur convolution  $z_s = [x_1 y_n \ x_2 y_{n-1} \ \dots \ x_n y_1]$ . Dans ce programme, vous devez définir chaque fonction que vous utilisez et chaque fonction doit être une boucle. Attention aux détails de syntaxe !

**FSAC1450 – Informatique T4 – Examen Final**  
**Prof. Peter Van Roy**  
**1 décembre 2004**

|                  |  |
|------------------|--|
| Nom              |  |
| Prénom           |  |
| Matricule (NOMA) |  |

**Question 2 (5 pts)**

Donnez des définitions concises et précises pour les concepts suivants, comme ils ont été utilisés dans le cours. Donnez un petit fragment de programme pour illustrer chaque concept et expliquez comment ce fragment montre le concept.

- Identificateur
- Variable
- Polymorphisme
- Concurrence
- Nondéterminisme

FSAC1450 – Informatique T4 – Examen Final  
Prof. Peter Van Roy  
1 décembre 2004

|                  |  |
|------------------|--|
| Nom              |  |
| Prénom           |  |
| Matricule (NOMA) |  |

Question 3 (5 pts)

Voici un petit programme :

```
declare
fun {NewCounter}
  A1={NewCell 0}
  proc {Inc} A1 := @A1+1 end
  proc {Get X} X=@A1 end
in
  proc {$ M}
    case M of inc then {Inc}
    [] get(X) then {Get X}
    end
  end
end
C={NewCounter}
{C inc}
local X in {C get(X)} {Browse X} end
```

Pour cette question, vous allez exécuter le programme avec la machine abstraite qu'on a vu dans le cours. Faites alors les deux choses suivantes :

1. Montrez l'état à la fin de l'exécution de ce programme.
2. Montrez quelques pas d'exécution représentatifs.

Vous pouvez supposer que le programme est exécuté avec un environnement initial qui contient l'identificateur `Browse`. N'oubliez pas de commencer votre réponse par une traduction partielle vers le langage noyau ! Ne sombrez pas dans les détails, mais choisissez bien les quelques pas que vous montrez. Expliquez en quelques mots ce que vous faites.

# FSAC1450 – Informatique T4 – Examen Final

Prof. Peter Van Roy

1 décembre 2004

|                  |  |
|------------------|--|
| Nom              |  |
| Prénom           |  |
| Matricule (NOMA) |  |

## Question 4 (5 pts)

Pour cette question vous devez écrire les définitions d'un même concept, les nombres complexes et leurs opérations, en type abstrait et en objet.

- *Type abstrait.* Définissez un type abstrait pour manipuler les nombres complexes. Un nombre complexe  $a + bi$  contient les deux nombres réels  $a$  et  $b$ . Vous pouvez modéliser les nombres réels avec des nombres en virgule flottante. Vous devez utiliser les opérations `Wrap` et `Unwrap` pour sécuriser le type abstrait. Vous pouvez supposer que la procédure `NewWrapper` existe et que l'appel `{NewWrapper Wrap Unwrap}` crée les deux fonctions `Wrap` et `Unwrap`. Voici les fonctions à définir :
  - La fonction `C={ToComplex A B}` prend deux nombres en virgule flottante `A` et `B`, qui représentent les réels  $a$  et  $b$ , et renvoie un nombre complexe `C` qui représente  $a + bi$ .
  - La fonction `C3={Add C1 C2}` prend deux nombres complexes `C1` et `C2`, qui représentent  $a + bi$  et  $c + di$ , et renvoie un nombre complexe `C3` qui représente leur somme  $(a + c) + (b + d)i$ .
  - La fonction `C3={Mul C1 C2}` prend deux nombres complexes `C1` et `C2`, qui représentent  $a + bi$  et  $c + di$ , et renvoie un nombre complexe `C3` qui représente leur produit  $(ac - bd) + (ad + bc)i$ .
- *Objet.* Pour la deuxième partie, vous allez essayer de définir les nombres complexes comme des objets. Vous devez définir une classe `Complex` avec trois méthodes `toComplex`, `add`, et `mul`. Votre classe ne doit contenir que ces trois méthodes. Voici ce que font ces méthodes :
  - La méthode `{C toComplex(A B)}` initialise le nombre complexe `C` avec deux nombres en virgule flottante `A` et `B`.
  - La méthode `{C1 add(C2)}` fait l'addition de deux nombres complexes `C1` et `C2` et modifie `C1` pour contenir le résultat.
  - La méthode `{C1 mul(C2)}` fait la multiplication de deux nombres complexes `C1` et `C2` et modifie `C1` pour contenir le résultat.

Essayez de définir les méthodes. Est-ce que c'est possible de les définir? Si c'est possible, donnez les définitions. Si ce n'est pas possible, expliquez pourquoi et expliquez ce qu'on peut faire pour le rendre possible. Qu'est-ce que fait Java dans ce cas?