



# Examen réparti d'ILP

Christian Queinnec

5 janvier 2012

## Conditions générales

Cet examen est formé d'un unique problème en plusieurs questions auxquelles vous pouvez répondre dans l'ordre qui vous plaît.

Le barème est fixé à 20 ; la durée de l'épreuve est de 3 heures. Tous les documents sont autorisés et notamment ceux du cours.

Votre copie sera formée de fichiers textuels que vous laisserez aux endroits spécifiés dans votre espace de travail pour Eclipse. L'espace de travail pour Eclipse sera obligatoirement nommé `workspace` et devra être un sous-répertoire direct de votre répertoire personnel.

À l'exception des clés USB en lecture seule, tous les appareils électroniques communicants sont prohibés (et donc notamment les téléphones portables). Vos oreilles ne doivent pas être reliées à ces appareils.

L'examen sera corrigé à la main, il est donc absolument inutile de s'acharner sur un problème de compilation en Java ou sur des méthodes à contenu informatif faible. En revanche quelques lignes de code sont souvent plus claires que plusieurs paragraphes filandreux. Il est beaucoup plus important de rendre aisé, voire plaisant, le travail du correcteur et de lui indiquer, par tout moyen à votre convenance, de manière claire, compréhensible et terminologiquement précise, comment vous surmontez cette épreuve. À ce sujet, vos fichiers n'auront que des lignes de moins de 80 caractères, n'utiliseront que le codage ASCII ou UTF-8 enfin, s'abstiendront de tout caractère de tabulation.

Les langages à étendre sont ILP3 puis ILP4. Le paquetage Java correspondant sera nommé, comme usuellement, `fr.upmc.ilp.ilp3retval`. Sera ramassé, à partir de votre `workspace` (situé sous ce nom directement dans votre HOME), le seul répertoire `ILP/Java/src/fr/upmc/ilp/ilp3retval/` et tout ce qu'il contient. Le mémento pour étendre les diverses versions d'ILP peut être de quelque utilité.

Pour vous éviter de la taper à nouveau, voici l'URL du site du master (mais il faut, pour y accéder, ne plus passer par le proxy) et celle de l'ARI où se trouvent de nombreuses documentations dont celle de Java :

```
http://www-master.ufr-info-p6.jussieu.fr/site-annuel-courant/  
http://www-ari.ufr-info-p6.jussieu.fr/
```

## Introduction

En Pascal, la valeur que renvoie une fonction  $f$  est spécifiée par une sorte d'affectation où  $f$  est le nom de la pseudo-variable affectée. Ainsi, peut-on écrire, en pseudo-Pascal :

```
function moinsun (x) {  
  if ( x > 0 ) {  
    moinsun := x-1  
  }  
}  
  
function fact (n) {  
  if n < 2  
  then  
    fact := one()  
    print("OK")  
  else  
    fact := n * fact(moinsun(n))  
    fact(one())  
  end  
}  
  
// Ainsi fact(3) imprime coucou et OK trois fois de suite et renvoie 6  
  
function quoi (x, y) {  
  x + y  
}  
  
function one () {  
  one := 1  
  print("coucou")  
}
```

On souhaite ajouter cette caractéristique à ILP3.

### Question 1 – ILP3 (9 points)

Décrire comment définir, interpréter et compiler cette nouvelle caractéristique dans le cadre d'ILP3.

#### Livraison

– le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp3retval/strategie.txt`

### Question 2 – Sous-spécification (3 points)

La fonction `quoi` dans l'exemple précédent ne spécifiait pas la valeur qu'elle pouvait retourner. On considère maintenant que cette fonction est erronée, modifier (si ce n'est déjà fait) les réponses précédentes afin que ce problème soit signalé.

#### Livraison

– le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp3retval/strategie.txt`

### Question 3 – Sur-spécification (3 points)

On souhaite maintenant qu'il soit aussi erroné de redéfinir la valeur de retour d'une fonction. Ainsi

```
function eerie (b) {
  if (b) {
    eerie := "encore OK"
    if (b) {
      eerie := "mais plus du tout OK"
    }
  } else {
    eerie := "OK"
  }
}
print(eerie(false)) // imprime OK
try {                // imprime KO
  eerie(true)
} catch (e) {
  print "KO"
}
```

Modifier (si ce n'est déjà fait) les réponses précédentes afin de signaler ce problème.

#### Livraison

– le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp3retval/strategie.txt`

### Question 4 – ILP4 (5 points)

Décrire comment définir, interpréter et compiler cette nouvelle caractéristique dans le cadre d'ILP4.

#### Livraison

– le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp3retval/strategie.txt`