

# Examen ILP 1ère session (Revision: 1.5 )

Christian Queinnec

21 décembre 2007

## 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 2.5 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. Cet espace de travail pour Eclipse sera obligatoirement nommé `workspace` et devra être un sous-répertoire direct de votre répertoire personnel.

L'examen sera corrigé à la main, il est donc absolument inutile de s'acharner sur un problème de compilation. 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.

Le langage sur lequel travailler est ILP4. Le paquetage Java correspondant à cet examen sera donc `fr.upmc.ilp.ilp4java`. Seul le répertoire associé à ce paquetage sera ramassé !

## 1 Compilation

On souhaite compiler un programme ILP4 non plus vers C mais vers Java. Les schémas de compilation qui vous seront demandés (dans le style de ceux montrés en cours) seront dessinés ainsi (ce qui suit est un exemple de schéma de compilation de l'opération binaire vers C) :

```

-----> d
operand1 operateur operand2

{ ILP_Object tmp1, tmp2;

-----> tmp1=
operand1;

-----> tmp2=
operand2;

d ( tmp1 operateur tmp2 );
}
```

Ce qui se paraphrase ainsi : la compilation d'une opération binaire de direction *d* engendre un bloc local introduisant deux variables locales temporaires (on dit « fraîches » pour indiquer que leur nom ne peut créer de conflit. Par convention, les variables temporaires ont des noms débutant par `tmp`). Le premier opérande est calculé et stocké dans la première variable temporaire ; le second opérande est calculé puis stocké dans la seconde variable temporaire. Enfin, l'opération est effectuée et retournée à la destination *d*.

Nota : À la différence du cours, dans les destinations affectant le résultat à une variable, j'ai suffixé le nom de la variable avec un signe d'affectation afin que le remplacement de *d* par sa valeur produise un code syntaxiquement correct. Vous pouvez mentionner ce signe `=` ou non à votre guise.

## Question 1 – Schémas ILP1 (6 points)

Donnez les schémas de compilation des constantes, de l'opération unaire, de l'alternative, de l'affectation, du bloc local unaire, de la séquence et de la boucle **while**. Vous ajouterez un commentaire à ces schémas s'ils ne s'autosuffisent pas.

**Livraison**

- un fichier `q1.txt` placé dans le répertoire `Java/src/fr/upmc/ilp/ilp4java/`.

**Question 2 – Schémas ILP4 (7 points)**

Donnez les schémas de compilation des variables globales, des définitions de fonctions, de l'invocation des fonctions globales (primitives ou non) ainsi que le schéma global de compilation d'un programme complet (ainsi que l'éventuel patron où il s'insère). Vous justifierez vos choix d'implantations (et notamment le protocole d'invocation des fonctions et primitives) notamment ceux dépendant des caractéristiques d'ILP4 ou de Java.

**Livraison**

- un fichier `q2.txt` placé dans le répertoire `Java/src/fr/upmc/ilp/ilp4java/`.

**Question 3 – Bibliothèque d'exécution (3 points)**

Décrivez puis implantez la bibliothèque d'exécution des programmes ILP4 compilés en Java.

**Livraison**

- le fichier `JavaRuntime.java` placé dans le paquetage `fr.upmc.ilp.ilp4java.runtime`.

**Question 4 – Implantation (4 points)**

Comment implanteriez-vous le compilateur ILP4 vers Java ? Pour étayer votre explication (qui vise à convaincre le correcteur que vous sauriez le faire), vous donnerez les classes correspondant à l'alternative accompagnées de toute autre classe nécessaire à la compréhension du compilateur.

**Livraison**

- un fichier `q4.txt` placé dans le répertoire `Java/src/fr/upmc/ilp/ilp4java/`.
- les fichiers `JCEASTParser.java`, `JCEASTalternative.java` (ainsi que les autres) placés dans le paquetage `fr.upmc.ilp.ilp4java.ast`.