

TP n°4

Bytecode Java

On s'intéresse ici à la compilation d'instructions Java en bytecode pour la JVM. Pour écrire, vérifier et exécuter directement ce bytecode, on se servira de `Jasmin`¹, un assembleur Java.

Exercice 1 (Compilation). Pour chacune des portions de code Java suivantes, déterminez une traduction en bytecode, compilez-la à l'aide de `jasmin`, et comparez vos propositions avec la sortie de `javap -c -verbose`.

- `int x = 1-2+5;`
- `System.out.println(42);`
- `int[] t = new int[3];`
`t[1] = 2;`
- `int[][] m = new int[2][2]; m[1][0] = 3;`
- `int res=1;`
`for(int i=0;i<=10;i++) res=res*i;`
- `static int fact(int n) {`
`if (n<=1) return 1;`
`else return (n*(fact (n-1)));`
`}`
- `class Point {`
`int x;`
`int y;`
`Point(int x1,int y1){x=x1;y=y1;}`
`void addVect(int u, int v){x+=u;y+=v;}`
`}`
- `class List {`
`int hd;`
`List tl;`
`int last () {`
`if (tl == null) return hd;`
`else return tl.last();`
`}`
`}`

Exercice 2. En écrivant du bytecode directement dans l'exercice précédent, peut-être avez-vous rencontrés des protestations du vérificateur de bytecode de Java. Il est possible de passer outre cette étape de vérification à l'aide de `java -noverify`. Que se passe-t-il dans ce cas:

- lors d'une pile non vidée en fin d'exécution d'une fonction?
- lors d'un usage de pile croissant au delà des limites prévues?
- lors d'un passage d'argument incorrect, par exemple si une méthode telle que `Object.clone` est appliquée à un entier.

¹Sa documentation complète, sa syntaxe ainsi que des exemples sont disponibles à l'adresse <http://jasmin.sourceforge.net/>. Un index détaillé des instructions de la JVM est consultable à l'adresse <http://homepages.inf.ed.ac.uk/kwxm/JVM/>

Exercice 3 (Examen 2007). Étant donné la portion de bytecode Java suivante:

```
.method public static f([I)I
  .limit locals 3
  .limit stack 3
  iconst_0
  istore_1
  iconst_0
  istore_2
L1:
  iload_2
  aload_0
  arraylength
  if_icmpge L2
  iload_1
  aload_0
  iload_2
  iaload
  iadd
  istore_1
  iinc 2 1
  goto L1
L2:
  iload_1
  ireturn
.end method
```

1. Proposer une portion de code Java dont la compilation produit ce bytecode. Associer à chaque élément de code Java la partie bytecode qui correspond.
2. Proposer une variante du bytecode ci-dessus permettant de minimiser l'usage de l'instruction `arraylength`