

Introduction à la programmation : contrôle  
continu (45 min.)  
Version 1

Octobre 2014

NOM :  
PRÉNOM :

Répondre aux questions sur cette feuille.

## 1 Évaluer une expression (4 points).

Quelle est la valeur des variables à la fin des instructions suivantes ?

```
int x=0;  
int y=x+1;  
if (y>x && x>0) {  
    x=x+1;  
} else {  
    y=x+1;  
};
```

x  y

```
int n=10;  
int acc=0;  
for (int i=0; i<n; i++) {  
    acc=acc+i;  
};
```

n  acc

```
int x = 10;  
if (x>5) {  
    b=(x>2);  
} else {  
    b=(x<10);  
};
```

x  b

```
int x = 2;  
if (x>5) {  
    b=(x>2);  
} else {  
    b=(x<10);  
};
```

x  b

## 2 Connaître les types d'une variable et typer une fonction (5 points).

Dans les extraits de code suivants, remplissez les trous par des types vus en cours. *Attention, il faut parfois lire l'extrait de code jusqu'au bout pour trouver le bon type.*

- (a) `public static  square( x){  
 y=x*x;  
return y;  
}`
- (b) `public static  printWithNewLine( x){  
printString(x+"\n");  
}`
- (c) `public static  concatNewLine( x){  
 s=(x+"\n");  
return s;  
}`
- (d) `public static  greater( x, y){  
return (x>=y);  
}`
- (e) `public static  mystery( x){  
 b = ((x%2)==0);  
if (b) {  
return (x/2);  
} else {  
return (3*x+1);  
}  
}`

### 3 Compléter un programme à trous (6 points).

- Compléter la procédure `line` ci dessous, afin qu'à son exécution elle affiche  $n$  caractères "\*" à l'écran puis saute une ligne. *Rappel : la procédure `printString(s)` affiche la chaîne de caractères  $s$  à l'écran. La procédure `printNewLine()` fait sauter une ligne.*

```
public static void line(int n){

    for(int i=0; i<...;i++){
        .....
        .....
        .....
    };
    .....
}
```

- Complétez ci-dessous la procédure `rectangle` prenant 2 arguments  $k$  et  $n$  et qui affiche  $k$  lignes de "\*" de longueur  $n$ .

```

public static void rectangle(int k,int n){

    for(int i=0; i<...;i++){
        .....
        .....
        .....
    };
}

```

- Dessinez ce que le code suivant va produire à l'écran lors de son exécution :

```

int j=0;
for(int i=0; i<5; i++){
    j=j+1;
    rectangle(i,j);
    printNewLine();
};

```

- Dessinez ce que le code suivant va produire à l'écran lors de son exécution :

```

for(int i=0; i<5; i++){
    for(int j=0; j<5; j++){
        rectangle(i,j);
        printNewLine();
    };
}

```

#### 4 Comprendre un énoncé (5 points + 2 bonus).

- Écrire une fonction `cube` qui prend un entier  $x$  en paramètre et renvoie  $x^3$ .

- Écrire une fonction `isCube` qui prend des entiers  $x$  et  $y$  en paramètres et renvoie le booléen `true` si  $x^3 = y$  et `false` sinon. *Rappel : la comparaison entre deux entiers en Java se fait avec le test `==`.*

- **(Question Bonus)** Écrire une fonction `HasCubicRoot` qui prend un entier  $x$  en paramètre et qui renvoie `true` si la racine cubique de  $x$  est un entier entre 0 et  $x$ .



*Rappel : un entier  $n \leq x$  est la racine cubique de  $x$  si  $n^3 = x$ .*

*Aide : utilisez une boucle `for` pour tester tous les entiers entre 0 et  $x$  en utilisant la fonction `isCube`. On utilisera une variable booléenne initialisée à `false` et qui sera mise à `true` si le compteur de boucle est la racine cubique de  $x$ .*