Université de Bourgogne
novembre 2004

Contrôle continu I1a

Durée 2h, tous documents autorisés

Le barème est donné à titre indicatif

Exercice 1 Algorithme de Syracuse (5 pts.) Pour un $n > 0$ donné par l'utilisateur:

- Si le nombre est pair on le divise par 2 et on obtient un nouveau nombre.
- Si le nombre est impair, on le multiplie par 3 et on ajoute 1 et on obtient un nouveau nombre.
- On recommence ce processus avec le nouveau nombre jusqu'à ce que celui ci soit égal à 1.

Ecrire un programme permmettant de décrire l'algorithme de Syracuse. Vous donnerez le nombre d'étapes nécessaires avant d'obtenir 1.

Exercice 2 (5 pts.) Proposer un programme Java qui permette la saisie d'un entier n et l'affichage de toutes les paires d'entiers compris entre 1 et n. Par exemple, avec $n = 4$, le programme devra afficher:

```
1,2
1,3
1,4
2,3
2,4
3,4
```

Exercice 3 (5 pts.) Proposer un programme Java permettant la saisie d'un entier n, le tirage aléatoire de n points (x, y), avec $x, y \in [0, 1[$, et l'affichage de nombre de ces points situés dans un cercle de centre $(0.5, 0.5)$ et de rayon 0.5.

Exercice 4 Chronomètre (5 pts.) Lisez tout l'énoncé avant de commencer.
1. (2 pts) On a une variable «int temps», qui contient un temps en secondes. Écrivez un bout de programme qui affiche ce temps sous forme d'heures, minutes, secondes, avec trois chiffres pour les heures, deux pour les minutes, et deux pour les secondes.

2. (2 pts) Si vous écrivez

```
long heure;
heure = System.currentTimeMillis();
```

alors la variable heure contient l'heure à laquelle cette instruction a été exécutée, comptée en millisecondes depuis une origine arbitraire (le premier janvier 1970, si vous tenez vraiment à le savoir).

Écrivez un chronomètre, c'est à dire un programme qui, indéfiniment, affiche à la seconde près, et de manière lisible, le temps depuis lequel il a été lancé. Ce programme doit en outre sonner une fois toutes les minutes.

Indications:

- Utilisez une variable «long heure_debut» pour noter l'heure à laquelle votre programme a été lancé. De même, utilisez une variable «derniere_sonnerie» pour noter quand il a sonné pour la dernière fois.
- Le caractère spécial $\backslash r$ fait revenir le curseur au début de la ligne. Ainsi, si vous écrivez «System.out.print("$\backslash r$");» (pas println!), ce qui sera écrit par la suite remplacera ce qui était déjà sur la ligne.
- Le caractère spécial $\backslash a$ fait sonner l'ordinateur.
- Vous pouvez vous dispenser de recopier le bout de programme de la question 1 si vous indiquez précisément ce qu'il faut reprendre.
- Ne faites pas attention aux détails entre long et int.

3. (1 points) La fonction «System.wait(temps);» arrête le programme pendant environ le temps indiqué (mesuré en millisecondes). Ajoutez au programme précédent le nécessaire pour qu'il ne consomme pas tout le temps de calcul de l'ordinateur à afficher le même temps des milliers de fois par seconde. Vous pouvez vous contenter d'indiquer précisément le(s) ligne(s) à ajouter à la question 2.

Vous pouvez également faire les trois questions directement dans un seul programme, si vous préférez.
4. (0 points) La prochaine fois que vous faites la cuisine, utilisez ce programme pour chronométrer le temps de cuisson, et pour vous rappeler d'aller remuer de temps en temps.