CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)

U.B.S. - I.U.T. de Vannes

Département Informatique
Contrôle de 2e année
Date : 23/10/2017

M3101 - Principe des systèmes d'exploitation

Devoir surveillé

Responsable(s) et auteur(s) : F. Merciol – M. Le Lain

Documents personnels et calculatrice autorisés
Téléphone interdit — Durée : 1h30
Les barèmes sont donnés à titre indicatif — 8 page(s) de texte

Ce contrôle est a rédiger sur le support d'énoncé lui-même. A la fin de l'examen, placez en sortant chaque partie dans sa corbeille

---

Conseils :

- Indiquez **votre nom** sur chaque feuille à rendre **dès qu'elles vous sont données**.
- Afin d'éviter la copie, toute réponse non justifiée sera considérée comme nulle.
- **Les parties sont indépendantes** et corrigées par des enseignants différents (commencez par la plus simple pour vous).
- **Rendez les 3 parties séparément** (dans les 3 réceptacles à la sortie).
- Il est demandé des réponses à la fois claires et concises.
- **Lisez en entier** le contrôle avant de commencer à répondre.
- **Ne restez pas bloqué**, vous pourrez revenir sur une question difficile par la suite.
- **Ne brûlez pas toute votre énergie**, des calculs longs peuvent vous rapporter moins de points que la réponse à des questions de réflexion.
- **Conservez du temps pour chaque partie**
  (90 min / 20 pts => pas plus de 4 minutes 30 par point)

Le devoir se compose de 3 parties :

1) [4 pts] questions de cours (QCM : Questions à Choix Multiple)
2) [8 pts] résolution de problème sur les tâches (en java)
3) [8 pts] Expressions régulières

Ce corrigé est donné à titre indicatif. D'autres réponses peuvent être considérés comme justes.

Corrigé (Les réponses apparaissent avec ce fond de caractère)

NOM : _____ PRENOM : _____ GROUPE TD : ____ Note :

1.) [4 pts] question de cours (QCM : Questions à Choix Multiple)

✗ Cochez la case réponse

Question	✾	✱	❀	✲
Les objets Java ou C++ créés dynamiquement lors de l'exécution sont placés la mémoire nommée ✾) code ✱) données ❀) pile ✲) tas				○
En Java, un moniteur ✾) prévient de la noyade ✱) neutralise un sémaphore ❀) s'oppose au Mutex ✲) résout une section critique				○
Les premiers systèmes de gestion de fichiers sont apparus dans les années : ✾) 90 ✱) 70 ❀) 80 ✲) 60				○
En informatique, MBR veux dire ✾) Manual Before Repair ✱) Memory Boot Recover ❀) Master Boot Record ✲) Mon Bien Réutilisable			○	
L'électricité est un vecteur de transport d'énergie, mais dans le monde, quelle source d'énergie est la moins utilisée par l'Internet ? ✾) Le renouvelable ✱) Le nucléaire ❀) Le charbon ✲) Les gaz de schiste	○			
On qualifiera une application de "temps réel" si des actions peuvent y être : ✾) en moins d'1 s ✱) en moins d'1 ms ❀) en moins d'1 ns ✲) abandonnées				○
Concernant l'emploi hors État-Unis d'Amérique, l'Internet ✾) repeuple les campagnes ✱) développe le commerce local ❀) créer des libraires ✲) détruit des emplois				○
Quel est le marquage de fin de ligne sous Unix ? ✾) \r\n ✱) \n\r ❀) \r ✲) \n				○
Sous Unix, le PID désigne : ✾) un programme ✱) la Pile Interne des Données ❀) un processeur ✲) un processus				○
Pour ne pas céder d'informations aux réseaux commerciaux de collecte de données personnelles (comme FaceBook), il faut ✾) interdire les cookies ✱) faire un vœu ❀) créer un profil restreint ✲) éviter les pages avec un f				○
Sous Unix, un processus terminé dont le père n'a pas encore lu le testament se nomme : ✾) accidenté ✱) orphelin ❀) en attente ✲) zombie				○
L'algorithme de suppression de page LRU : ✾) tient compte du dernier utilisé ✱) prive de ressources ❀) prend en file ✲) vide le cache	○			
La mémoire flash ✾) permet de modifier un octet ✱) sert à étaler sa vie privée ❀) est une forme d'EEPROM ✲) ne dure pas longtemps			○	
StringTokenizer découpe en ✾) anneaux magiques ✱) lignes ❀) mots ✲) paragraphes			○	
En informatique, l'acronyme RAID désigne ✾) un regroupement de disques ✱) une unité de police ❀) une épreuve sportive ✲) un algorithme de recherche	○			
Le SFINX est ✾) l'interconnexion des réseaux en France ✱) un routeur aléatoire ❀) un animal mythique ✲) le protocole d'échange égyptien	○			
Sous Unix, un processus démon ne peut jamais être : ✾) défunt ✱) suspendu ❀) créé par un moldu ✲) tué	○			
Un programme Java qui ouvre une fenêtre se termine ✾) par un banquet ✱) par appel à System.exit ❀) à la fin du main ✲) dans un temps déterminé		○		
Quel est le cycle énergétique le plus court ? ✾) Le nucléaire ✱) Le pétrole ❀) Le papier ✲) Celui de la comète de Halley			○	
Pour Unix, un numéro majeur désigne ✾) un dispositif matériel ✱) une expression régulière ❀) un développement important ✲) une version stable	○			

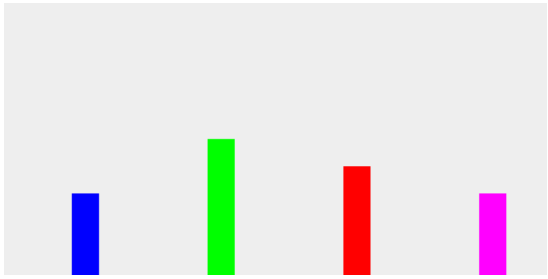
+	-	.	/20

CORRIGÉ (LES RÉPONSES APPARAISSENT AVEC CE FOND DE CARACTÈRE)

NOM : _____ PRENOM : _____ GROUPE TD : ____ Note :

2.) [8 pts] Résolution de problème sur les tâches (en java)

« Regarde pousser les arbres ! »



Rien de plus reposant que de regarder la nature et les arbres pousser. D'accord ce sont plutôt des arbres en béton coloré. Mais au moins on peut accélérer le temps pour les voir changer.

Dans cet exemple, très stylisé, nous avons deux classes d'objets :

- le **Terrain** est matérialisé par un fond gris et réalisé par un *JPanel* et
- les **Arbre**(s) matérialisés par des rectangles de couleur et réalisés par des *JLabel*.

Chaque arbre passe son temps à demander au terrain de le nourrir. Mais il doit attendre son tour. Le terrain va régulièrement faire pousser un arbre au hasard en le réveillant. Un arbre réveillé peut passer son tour. Lorsqu'un arbre à atteint le maxime, il indique que tous les arbres doivent repartir à zéro en les réveillant tous.

Voici la classe Java Arbre qui vous est fournie :

```
public class Arbre extends JLabel {
    Terrain terrain;
    int index, age;

    public Arbre (Terrain terrain, Color couleur, int index) {
 this.terrain = terrain;
 setBackground (couleur);
 setOpaque (true);
 this.index = index;
 setAge (1);
 terrain.add (this);
 startSingle ();
    }

    public void setAge (int age) {
 this.age = age;
 setLocation ((int)((index+.5)*Terrain.AE), (Terrain.MA-age)*Terrain.AS);
 setSize (Terrain.AS, age*Terrain.AS);
    }

    public boolean pousse () {
 setAge (age+1);
 return age > Terrain.MA;
    }

    public void recommence () {
 setAge (1);
    }
// startSingle fonction qui démarre l'activité de l'arbre
}
```

Corrigé (Les réponses apparaissent avec ce fond de caractère)

```
    public class Terrain extends JPanel {

      static public int AS = 20, AE = 100;
      static public int NA = 4, MA = 10;
35    static public int TL = AE*NA, TH = MA*AS;

      static public void main (String[] arg) {
   Terrain terrain = new Terrain (TL, TH);
   testComponent ("Terrain", terrain);
40
   new Arbre (terrain, Color.BLUE, 0);
   new Arbre (terrain, Color.GREEN, 1);
   new Arbre (terrain, Color.RED, 2);
   new Arbre (terrain, Color.MAGENTA, 3);
45
   for (;;) {
       pause (500);
       terrain.reveilUnArbre ();
   }
50    }

      static public void pause (int millis) {
   try {
       Thread.sleep (millis);
55 } catch (InterruptedException e) {
   }
      }

      public Terrain (int width, int height) {
60 super (null);
   Dimension size = new Dimension (width, height);
   setMinimumSize (size);
   setPreferredSize (size);
      }
65
      int aRecommencer = 0;
      public void nourri (Arbre arbre) {
   if (aRecommencer > 0) {             // (A)
       arbre.recommence ();
70     aRecommencer--;
   } else if (((int)(Math.random () * NA)) == 0)    // (B)
       notify ();
   else if (arbre.pousse ()) {         // (C)
       aRecommencer = NA;
75     notifyAll ();
   }
   try {
       wait ();
   } catch (InterruptedException e) {
80 }
      }
  // reveilUnArbre fonction à écrire
  // testComponent fonction crée une JFrame contenant le Terrain
}
```

Corrigé (Les réponses apparaissent avec ce fond de caractère)

Compréhension du code.

2.A) [½ pt] A quoi sert la méthode « pause » ? Pourquoi est-elle « static » ?

La méthode pause permet d'attendre un nombre de secondes en paramètre. La méthode n'utilise qu'une méthode de classe de Thread et n'est pas liée à un objet particulier.

2.B) [½ pt] Quel effet produit la méthode « pousse » sur un arbre ? Qu'indique-t-elle en retour ?

Incrémente l'age d'un arbre et modifie l'affichage en conséquence. Retourne vrai si l'arbre a atteint son maximum.

2.C) [1,5 pt] Quel est le rôle des 3 tests (A), (B), et (C) de la méthode nourri

(A) réinitialise les arbres si ils en reste à faire.
(B) passe son tour au moment du réveil
(C) un arbre est réveillé alors il pousse. Si il arrive au maximum, il déclenche la réinitialisation de tous.

Utilisation des tâches.

2.D) [½ pt] Il y a une erreur dans la définition de la méthode « nourri ». Ecrivez la correction ci-dessous.

public synchronized void nourri (Arbre arbre) {
Sinon il n'est pas possible de bloquer la pousse des arbres.

2.E) [2 pt] Quelle différence entre *notify* et *notifyAll* dans les lignes 72 et 75 ? Expliquez cette différence dans chaque cas (B) et (C).

(B) notify va réveiller un seul arbre pour qu'il pousse ou passe son tour en (B)
(C) notifyAll va réveiller tous les arbres pour qu'il soir réinitialisés en (A)

Corrigé (Les réponses apparaissent avec ce fond de caractère)

Complément du programme.

2.F) [1 pt] Ecrivez la méthode « reveilUnArbre ».

```
public synchronized void reveilUnArbre () {
 notify ();
}
```

2.G) [2 pt] Ecrivez la méthode startSingle.

```
public Thread thread;
public void startSingle () {
    (new Thread () {
     public void run () {
        for (thread = Thread.currentThread (); thread == Thread.currentThread (); )
      terrain.nourri (Arbre.this);
      }
 }).start ();
}
```

Corrigé (Les réponses apparaissent avec ce fond de caractère)

NOM : _____ PRENOM : _____ GROUPE TD : ____ Note :

3.) [8 pts] Expressions régulières

Nous nous proposons d'analyser les lignes produites en interrogeant les logs utilisateurs d'un PC. Elles sont constituées par les informations suivantes : Date, heure, machine, type et numéro de service, message.

```
Oct 17 10:39:24 PC-de-Users3 MessageActivityService[395] discarding window 294 as it has a parent 305
Oct 17 10:41:20 PC-de-Users3 MessageActivityService[546] discarding window 947 as it has a parent 562
Oct 17 10:43:31 PC-de-Users3 MessageActivityService[486] discarding window 384 as it has a parent 966
Oct 17 10:43:42 PC-de-Users3 MessageActivityService[566] discarding window 854 as it has a parent 240
Oct 17 10:43:54 PC-de-Users3 MessageActivityService[192] discarding window 129 as it has a parent 183
Oct 17 10:44:17 PC-de-Users3 MessageActivityService[394] discarding window 845 as it has a parent 637
Oct 17 10:44:28 PC-de-Users3 MessageActivityService[193] discarding window 963 as it has a parent 689
```

3.A) [4,5 pt] Proposer un algorithme permettant de vérifier en plusieurs expressions régulières la composition des lignes ci-dessus (une expression par élément de ligne).

Corrigé (Les réponses apparaissent avec ce fond de caractère)

3.B) [1 pt] Proposer une expression régulière permettant de vérifier la totalité de la structure des lignes (une expression pour la ligne).

3.C) [2,5 pt] Proposer un algorithme avec une expression régulière permettant de vérifier la véracité d'une adresse web. Ces adresses web sont du format : http://www.nomdudomaine.fr avec ces caractéristiques :

- Adresses déclarées en France (.fr) ou adresses commerciales (.com)
- Protocole : http
- Sous-domaine : www
- Nom de domaine de deuxième niveau (exemple : nomdudomaine) : 12 caractères maximum