heb
esi

Nom : .....
Prénom : .....
Groupe : .....
Identifiant : .....

/ 50

Haute École de Bruxelles
École Supérieure d'Informatique
Bachelor en Informatique

2015 – 2016

Développement – 1ère

Examen

Mahé

Consignes

Pour l'ensemble de l'examen,

- Vous avez 4 heures de temps ;
- Vous gérerez ce temps selon votre meilleure convenance pour réaliser les solutions en algorithmique et en Java.

Mahé est un jeu qui se joue à maximum 6 joueurs. Chaque joueur possède une tortue qui tente de pondre un maximum d'œufs à un endroit déterminé. À chaque fois que votre tortue dépasse le tour de l'ile (21 cases), elle pond une série d'œufs matérialisée sous la forme d'une carte chiffrée se trouvant au dessus du paquet (voir le dessin).

Quand c'est à vous de jouer un coup :

Version du document : 17 janvier 2016

— Vous lancez un dé.
    — Vous avez le choix entre avancer votre tortue de la valeur du dé ou lancer un deuxième dé.

      Si vous choisissez d'avancer votre tortue, votre coup est terminé.
— Dans le cas où vous lancez un deuxième dé :
    — Si le total dépasse 7, vous retournez à la case départ (la case 0, le radeau précédent la case 1 sur le dessin) et votre coup est terminé.
    — Si le total ne dépasse pas 7, vous avez le choix entre avancer de 2 fois le total des dés ou lancer un troisième dé.

      Si vous choisissez d'avancer votre tortue, votre coup est terminé.
— Si vous lancez un troisième dé, c'est pareil :
    — Si vous dépassez le total de 7, vous retournez sur le radeau à la case départ et votre coup est terminé.
    — Si pas, vous avancez de 3 fois le total des dés et votre coup est terminé !

Le tour de l'ile se termine dès qu'une tortue **dépasse** la case 21, elle pondra alors le nombre d'œufs se trouvant sur la carte visible. Cette carte est retirée du tas de carte.

Toutes les tortues retournent sur le radeau, en position de départ. Elles sont prêtes à recommencer un nouveau tour de l'ile.

Le jeu se termine

— Quand toutes les cartes chiffrées représentant les œufs pouvant être pondus sont épuisées.

I Algorithmique

Consignes

Pour la partie algorithmique,

- Vous ne pouvez pas utiliser de notes.
- Vos réponses se feront au bic bleu ou noir sur la feuille de réponses.
- Sauf spécification du contraire, les données lues ou reçues ne comportent pas d'erreurs.
- Les noms en gras (variables et types) doivent être respectés.
- Veillez à travailler de manière modulaire.
- N'hésitez pas à appeler un algorithme déjà écrit dans une question précédente.

1 Afficher les tortues (4 points)

Écrivez un algorithme **afficherTortues** qui reçoit un tableau de **n** entiers contenant des positions des tortues sur le plateau de jeu et qui affiche la position de chaque tortue.

Par exemple : si le tableau reçu contient {6, 10, 6, 0, 23}, votre algorithme affichera :
La tortue 0 est en position 6
La tortue 1 est en position 10
La tortue 2 est en position 6
La tortue 3 est en position 0
La tortue 4 est en position 23

```
module afficherTortues(tortues : tableau de n entiers)
    pour i de 0 à n-1 faire
        afficher "La tortue ", i, " est en position ", tortues[i]
    fin pour
fin module
```

2 Joueurs (4 points)

Écrivez un algorithme **getTortues** qui reçoit **nbTortues**, le nombre de tortues en paramètre, qui crée un tableau de **nbTortues** entiers, qui, sans utiliser la notation raccourcie, initialise toutes les cases du tableau à 0 et qui retourne ce tableau.

```
module getTortues(nbTortues : entier) → tableau de nbTortues entiers
    tortues : tableau de nbTortues entiers
    pour i de 0 à nbTortues-1 faire
        tortues[i] ← 0
    fin pour
    retourner tortues
fin module
```

3 Jouer 1 coup (8 points)

Écrivez un algorithme **jouerCoup** qui reçoit le tableau de **nbTortues** entiers des positions des tortues sur le plateau de jeu et le numéro de la tortue qui joue (le numéro commence à 0) et qui joue le coup.

Pour jouer un coup,

- Lancez un dé à 6 faces.
  - Laissez le choix en demandant soit d'avancer la tortue soit de lancer un deuxième dé. Vous lirez un booléen pour savoir si il veut continuer ou pas.

    Si la tortue avance, le coup est terminé.
- Dans le cas du lancement d'un deuxième dé :
  - Si le total dépasse 7, retour de la tortue à la case départ (0) et le coup est terminé.
  - Si le total ne dépasse pas 7, laisser le choix entre avancer la tortue de 2 fois le total des dés ou lancer un troisième dé. Vous lirez un booléen pour savoir si il veut continuer ou pas.

    Si la tortue avance, le coup est terminé.
- Dans le cas du lancement d'un troisième dé, c'est pareil :
  - Si le total dépasse 7, retour de la tortue à la case départ (0) et le coup est terminé.
  - Si pas, avancer la tortue de 3 fois le total des dés et le coup est terminé.

Nous ne gérons pas encore ici le fait qu'une tortue puisse avoir fait le tour de l'ile.

Vous ferez appel à l'algorithme **hasard(n : entier)** pour lancer le dé.

```
module jouerCoup(tortues↓↑ : tableau de nbTortues d'entiers, numTortue↓ : entier)
    total, nbLancer : entiers
    continuer : booléen

    total ← hasard(6)
    nbLancer ← 1
    afficher "Vous avez ", total
    afficher "Voulez-vous continuer ?"
    demander continuer
    tant que nbLancer < 3 ET continuer faire
        total ← total + hasard(6)
        nbLancer ← nbLancer + 1
        afficher "Vous avez ", total
        si total < 7 ET nbLancer < 3 alors
            afficher "Voulez-vous continuer ?"
            demander continuer
        sinon
            continuer ← faux
        fin si
    fin tant que
    si total <= 7 alors
        tortues[numTortue] ← tortues[numTortue] + nbLancer*total
    sinon
        tortues[numTortue] ← 0
    fin si
fin module
```

4 Jouer 1 tour complet de l'ile (9 points)

Écrivez un algorithme **jouerTour** qui reçoit un tableau de **nbTortues** entiers des positions des tortues sur le plateau de jeu, un tableau de **nbTortues** entiers du nombre d'œufs déjà pondus par chaque tortue et du nombre d'œufs à pondre.

Cet algorithme permettra de jouer un tour complet, c'est-à-dire jusqu'à ce qu'une tortue dépasse la case 21 et remporte le nombre d'œufs qu'elle rajoute à son nombre d'œufs déjà pondus. Ce n'est pas cet algorithme qui se charge de remettre les tortues sur le radeau à la fin d'un tour.

C'est le joueur numéro 0 qui commence.

Pensez à afficher les positions de chacune des tortues au fur et à mesure.

```
module jouerTour(tortues↓↑ : tableau de nbTortues d'entiers, oeufs↓↑ : tableau de
nbTortues d'entiers, oeufsAGagner↓ : entier)
    joueurCourant : entier
    gagné : booléen
    joueurCourant ← 0
    gagné ← faux
    tant que NON gagné faire
        afficherTortues(tortues)
        afficher "C'est au joueur ", joueurCourant, " de jouer"
        jouerCoup(tortues, joueurCourant)
        gagné ← tortues[joueurCourant] > 21
        si NON gagné alors
            joueurCourant ← (joueurCourant + 1) MOD nbTortues
        fin si
    fin tant que
    oeufs[joueurCourant]← oeufs[joueurCourant]+ oeufsAGagner
fin module
```

II
Java et laboratoire

Consignes

Pour la partie java,

- *Vous réaliserez votre travail sur* `linux1` *et le déposerez dans le casier linux de votre professeur par la commande* `casier`.
- *Vous disposez de toutes vos notes ainsi que de l'aide en ligne.*
- *Il ne suffit pas que votre code compile. Testez-le pour identifier d'éventuelles erreurs à l'exécution.*
- *La cotation tiendra compte aussi du style de programmation que vous avez acquis.*
- *Respectez bien les noms de package, classe, méthodes demandés dans l'énoncé.*
- *Vous remplacerez bien sûr g12345 par votre numéro d'étudiant.*

5 Sine qua non (0 point)

Créez un répertoire `evaluations/janvier`. Changez les droits sur votre répertoire `janvier` pour donner les permissions de lecture et d'exécution aux professeurs mais aucun droit aux autres étudiants. Appelez votre professeur pour lui montrer que vos permissions ont bien été changées.

Vous ne continuerez pas l'examen tant que cette question n'a pas été validée par votre professeur.

6 Travailler dans un package (3 points)

Dans la suite de l'interro, votre classe fera partie du package `evaluations.janvier`.

Votre programme s'appellera *Mahé.java* et sera situé dans
`/home/g12345/evaluations/janvier/sources/` [1]
et *Mahé.class* dans
`/home/g12345/evaluations/janvier/classes/evaluations/janvier`.

Écrivez ici :

— l'instruction que doit contenir votre classe pour faire partie du package demandé ;
package evaluations.janvier ;

— la commande (complète et précise) que vous allez utiliser pour **compiler** votre classe ;
javac -d ~/evaluations/janvier/classes Mahé.java

1. Vous remplacerez évidemment *g12345* par **votre** identifiant

— la commande (complète et précise) que vous allez utiliser pour **exécuter** votre classe ;
java evaluations.janvier.Mahé
— le contenu minimal de votre variable d'environnement *CLASSPATH*.
~/evaluations/janvier/classes

7 Mahé (6 points)

Dans votre classe *Mahé*, écrivez les méthodes :

— public static void afficherTortues(int[] tortues) ;
— public static int[] getTortues(int nbTortues) qui lancera une IllegalArgumentException si **nbTortues** n'est pas compris entre 2 et 6 ;
— public static void jouerCoup(int[] tortues, int numTortue) ;
— public static void jouerTour(int[] tortues, int[] oeufs, int oeufsAGagner).

8 UtilMahé (6 points)

Nous avons écrit pour vous une classe *UtilMahé* qui contient une méthode

public static void jouerPartie(int nbTortues)

permettant de jouer une partie complète à partir des méthodes que vous avez écrites.

Nous savons qu'elle se trouve sur la machine linux1 dans un des sous-répertoires de `/eCours`.

La javadoc de la classe est à votre disposition dans un répertoire `doc` situé à côté de la classe.

— Quelle commande allez-vous utiliser pour retrouver le fichier UtilMahé.class ?
— Que devrez-vous ajouter au contenu de la variable d'environnement *CLASSPATH* pour pouvoir exécuter la classe UtilMahé ?
— Écrivez votre méthode *main* dans laquelle vous ferez simplement appel à la méthode jouerPartie(int nbTortues) de UtilMahé pour 2 joueurs.

9 Javadoc (4 points)

Écrivez la javadoc pour la méthode public static int[] getTortues(int nbTortues).

10 JUnit (6 points)

Écrivez une série de tests **JUnit** pour la méthode public static int[] getTortues(int nbTortues).

Écrivez ici :

— la commande permettant d'afficher le résultat des tests JUnit ;
java org.jUnit.runner.JUnitCore evaluations.janvier.MahéTest

— le contenu de votre variable d'environnement *CLASSPATH* pour pouvoir exécuter ces tests.
/usr/share/java/junit4.jar

```
1  package evaluations.janvier;
2
3  import java.util.Scanner;
4  import util.mahé.UtilMahé;
5
6  public class Mahé{
7
8      public static void main(String[] args){
9          UtilMahé.jouerPartie(2);
10     }
11     /** affiche les positions des tortues
12      * @param tortues les positions à afficher
13      */
14     public static void afficherTortues(int [] tortues){
15         for(int i=0; i< tortues.length; i++){
16             System.out.println("La tortue " + (i) +
17                     " est en position " + tortues[i]);
18         }
19     }
20
21     private static void afficherOeufs(int [] oeufs){
22         for(int i=0; i< oeufs.length; i++){
23             System.out.println("La tortue " + (i) +
24                     " a pondu " + oeufs[i]);
25         }
26     }
27
28     /** crée un tableau de nbTortues intitalisé à 0
29      * @param nbTortues taille du tableau à créer
30      * @return le tableau des nbTortues positions initialisées à 0
31      * @throws IllegalArgumentException si le nbTortues n'est pas entre 2 et 6.
32      */
33     public static int[] getTortues(int nbTortues){
34         if (nbTortues < 2 || nbTortues > 6)
35             throw new IllegalArgumentException ("paramètre invalide");
36
37         return new int[nbTortues];
38     }
39
40     private static int hasard(int nb){
41         return (int)(Math.random() * 6 + 1);
42     }
43
44     /** joue un coup
45      * @param tortues le tableau des positions des tortues
46      qui sera modifié avec la position après le coup
47      * @param numTortue le numéro de la tortue àbouger
```

```
48          entre 0 et tortues.length - 1
49          * @throws IllegalArgumentException si numTortue
50          n'est pas dans les bornes.
51          */
52      public static void jouerCoup(int[] tortues, int numTortue){
53          if (numTortue<0 || numTortue >= tortues.length)
54              throw new IllegalArgumentException ("paramètre invalide");
55
56          Scanner clavier = new Scanner(System.in);
57          int total;
58          int nbLancer;
59          boolean continuer;
60
61          total = hasard(6);
62          nbLancer = 1;
63          System.out.println("Vous avez " + total +
64                  "\nVoulez-vous continuer (true/false) ?");
65          continuer = clavier.nextBoolean();
66
67          while(nbLancer<3 && continuer){
68              total = total + hasard(6);
69              nbLancer = nbLancer + 1;
70              System.out.println("Vous avez " + total);
71              if(total<7 && nbLancer<3){
72                  System.out.println("Voulez-vous continuer (true/false) ?");
73                  continuer = clavier.nextBoolean();
74              } else {
75                  continuer = false;
76              }
77          }
78
79          if(total <= 7){
80              tortues[numTortue] = tortues[numTortue] +
81                  nbLancer*total;
82          } else{
83              tortues[numTortue] = 0;
84          }
85      }
86
87      /** joue un tour complet
88       * @param tortues les positions des tortues
89       * @param oeufs le nombre d'oeufs/depoints déjà acquis
90       * @param oeufsAGagner le nombre d'oeufs àgagner
91       */
92      public static void jouerTour(int[] tortues, int [] oeufs, int oeufsAGagner){
93          int joueurCourant;
94          boolean gagné;
95
96          joueurCourant = 0;
97          gagné = false;
98
99          System.out.println("Vous jouez pour "  + oeufsAGagner + " oeufs");
100
101         while(!gagné){
102             afficherTortues(tortues);
103             System.out.println("C'est au joueur " + joueurCourant +
104                 " de jouer");
105             jouerCoup(tortues, joueurCourant);
```

```
106                 gagné = tortues[joueurCourant] > 21;
107
108                 if (!gagné){
109                     joueurCourant = (joueurCourant + 1)%tortues.length;
110                 }
111             }
112             oeufs[joueurCourant] = oeufs[joueurCourant] + oeufsAGagner;
113
114             afficherOeufs(oeufs);
115         }
116 }
```

11 Javadoc (x points)

Générez la javadoc dans le répertoire `doc`.

javadoc -charset utf-8 -d doc Mahé.java