DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS: IFT1010 (Hiver 2002)

ENSEIGNANTS: Michel Reid & Philippe Langlais

TITRE DU COURS: Programmation I

SOLUTIONNAIRE DE L'EXAMEN INTRA

Date : 21 février
Heure : 17:30 - 19:30

DIRECTIVES PÉDAGOGIQUES:

- Aucune documentation n'est permise, pas même vos propres brouillons.
- **Inscrivez tout de suite votre nom et code permanent dans l'encadré.**
- Répondez **sur le questionnaire** en utilisant l'espace libre qui suit chaque question. Si vous manquez de place, écrivez au dos des feuilles, en l'indiquant clairement.

NOTATION:

Exercice 1:	/20	Exercice 5:	/15
Exercice 2:	/15	Exercice 6:	/20
Exercice 3:	/1	Exercice 7:	/20
Exercice 4:	/10		
Total:	/101		

Inscrivez votre nom et votre code permanent ici

CONSEIL:
Les exercices 1, 2 et 3 contrôlent vos connaissances générales de Java. N'y passez pas un temps déraisonnable (vous connaissez la réponse ou pas). Les autres exercices vérifient votre aptitude à la programmation structurée (prenez le temps de réfléchir).

Exercice 1 (20 points):

Dans cet exercice on suppose que la variable `a` est de type `int` et qu'elle peut prendre n'importe quelle valeur positive ou nulle de ce type.

Question A- Simplifiez[1] cette instruction:

```
if ( ((a%2) ==0) || ((a%2) == 1) )     System.out.println("IFT");
else                                   System.out.println("1010");
```

Réponse:

Cette instruction dit: *si a est pair ou si a est impair alors imprimer IFT sinon imprimer 1010.* L' instruction suivante est donc équivalente:

```
System.out.println("IFT");
```

Question B- Simplifiez cette instruction:

```
System.out.println( ((a > 5) && (a > 7))? "A" : "B");
```

Réponse:

```
System.out.println((a > 7)? "A" : "B");
```

Question C- Les deux morceaux de programmes suivants sont-ils identiques ? Dans la négative, indiquez une valeur de `a` pour laquelle ces deux programmes diffèrent.

Programme 1:

```
if (a>5)
  if (a>7) System.out.println("A");
  else System.out.println("B");
else
  System.out.println("C");
```

Programme 2:

```
if (a>7) System.out.println("A");
  else
    if (a>5) System.out.println("B");
    else
      System.out.println("C");
```

Réponse: oui !

[1] changer pour une instruction plus simple qui a exactement le même comportement.

Question D- Soit le bloc d'instructions:

```
switch (a) {
  case 1:
  case 2: System.out.print(a); break;
  case 3: System.out.print(a);
  case 4: System.out.print(a); break;
  case 5: System.out.print(a-1); break;
  default: System.out.print(a); break;
}
```

Quel affichage produit-il si a vaut 1 ? **Réponse: 1**
Quel affichage produit-il si a vaut 2 ? **Réponse: 2**
Quel affichage produit-il si a vaut 3 ? **Réponse: 33**
Quel affichage produit-il si a vaut 4 ? **Réponse: 4**
Quel affichage produit-il si a vaut 5 ? **Réponse: 4**

Question E- Écrivez de la manière la plus simple possible le bloc d'instructions de la question D avec des instructions `if/else`.

Réponse: Finalement le bloc précédant affiche toujours la valeur de a, sauf dans le cas où a vaut 3 (affichage = 33) et dans le cas où a vaut 5 (affichage = 4). Voici deux versions plus simples de ce bloc d' instructions:

Solution 1:

```
if (a == 5) System.out.print(4);
else {
  System.out.print(a);
  if (a == 3)
    System.out.print(3);
}
```

Solution 2:

```
if (a == 5) System.out.print(4);
else
  if (a == 3) System.out.print(33);
  else  System.out.print(a);
```

Exercice 2 (15 points):

Indiquez pour chaque bloc d'instructions, l'affichage produit. Si les instructions ne sont pas compilables (compilateur = javac), signalez-le par **ERREUR DE COMPILATION** puis expliquez le problème. Si l'instruction est compilable, mais peut provoquer une erreur à l'exécution (machine virtuelle = java), signalez-le par **ERREUR D'EXÉCUTION** et expliquez le problème.

```
int i=4;
int j= i++ * 3;
System.out.println("i=" + i + " j=" + j);
```

Réponse: i=5 j=12

```
int i = 4;
int j = (((i%3) + 3) == i)? (i++)/2 : (++i)/3;
System.out.println("i=" + i + " j=" + j);
```

Réponse: i=5 j=2

```
int i = Keyboard.readInt();
boolean b;

if ( (i==0) || (b = (i == 4)) )  b = ! b;
System.out.println("i=" + i + " b=" + ((b)? "VRAI": "FAUX"));
```

Réponse: ERREUR DE COMPILATION car lorsque `i` est nul, alors le test est vrai et l'instruction exécutée demande la valeur de `b` qui n'est pas initialisée.

```
int i = 0;
System.out.println("i=" + ((((i % 2) != 0) && ((4/i) == 3))? i+3 : i-2));
```

Réponse: i=-2

```
int i = 0;
System.out.println("i=" + ((((i % 2) != 0) || ((4/i) == 3))? i+3 : i-2));
```

Réponse: ERREUR D'EXÉCUTION car le premier terme de l'expression est faux, donc le second est évalué et puisque `i` est nul, cela provoque une division par 0.

Exercice 3 (1 point):

Quel est le nom précis et complet que vous donneriez au fichier qui contient ce programme, si vous souhaitez le compiler avec le compilateur `javac` ?

```
public class IFT1010 {
  public static void main(String [] args) {
    System.out.println("Une seule réponse possible");
  }
}
```

Réponse: IFT1010.java

Exercice 4 (10 points):

Soit le programme:

```
int methode(int valeur) {
  int res = 0;

  while (valeur != 0) {
    if (((valeur % 10) % 2) == 1) ++res;
    valeur /= 10;
  }
  return res;
}
```

Quelle est la valeur de retour de l'appel `methode(0);`? **Réponse: 0**
Quelle est la valeur de retour de l'appel `methode(10);`'? **Réponse: 1**
Quelle est la valeur de retour de l'appel `methode(11267);`? **Réponse: 3**

Énoncer en une phrase simple et générale ce que fait cette méthode (paraphraser n'est pas une réponse *(ex: cette méthode réalise une boucle à l'intérieur de laquelle, si valeur vaut...)*):

Réponse: La méthode retourne le nombre de chiffres impairs du nombre entré en argument.

Exercice 5 (15 points):

Écrivez une méthode qui reçoit un entier en paramètre. Si cet entier est pair ou inférieur à 3, la méthode n'affiche rien. Sinon, la méthode affiche un signe plus à l'écran avec des caractères $\star$. L'entier passé en argument fixe la hauteur et la largeur du signe plus. Voici des exemples d'affichages que votre méthode doit produire avec différentes valeurs d'argument.

```
 *      *        *           *            *
***     *        *           *            *
 *    *****      *           *            *
        *     *******        *            *
        *        *       *********        *
                 *           *       ***********       ...
                 *           *            *
                             *            *
                             *            *
                                          *
                                          *

 3      5       7            9            11           ...
```

Réponse de l'exercice 5:

Il s'agit d'une solution possible, il en existe bien sur d'autres.

```
void question5(int n) {
  if ((n % 2 == 1) && (n >2)) {
     int mil = n/2;

     for (int i=0; i<n; i++) {
       for (int j=0; j< n; j++) System.out.print(((j == mil) || (i == mil))? "*":" ");
       System.out.println();
     }
  } // si n repond aux exigences de l'énoncé
} // question5
```

Exercice 6 (20 points):

Écrivez une méthode qui prend deux nombres entiers positifs en paramètres et retourne `true` si l'un est la représentation de l'autre de droite à gauche et `false` sinon. **Remarque:** les zéros à la fin du nombre sont ignorés (voir les exemples).

Exemples:

- Pour les nombres `123` et `321`, la méthode retourne true.
- Pour les nombres `123` et `32100` aussi.
- Pour les nombres `4567` et `7645`, la méthode retourne false.

Une solution possible:

```
boolean question6(int a, int b) {

  // inversons l'une des valeurs (ici b) -> resultat dans val
  // ex: b = 12300 val = 123
  int val = 0;

  while (b != 0) {
    val = val * 10 + (b % 10);
    b /= 10;
  }

  // egalite directe ou facteur 10,100, etc
  int max = (val < a)? a;
  int min = (val < a)? val;

  return ((max % min) % 10) == 0; // inclus l'egalite
}
```

Une autre solution possible:

```
  public static boolean inverse (int a, int b){
    int c = b, d=0;

    // on elimine les 0 eventuels a la fin de a
    while(a>0 && a%10 == 0) a/=10;

    while(c>0){
       d = d*10;
       d += c%10;
       c /= 10;
    }

    return (d==a);
  }
```

Exercice 7 (20 points):

Écrivez une classe `Livre` qui contiendra les informations suivantes:

- le nom de l'auteur
- titre du livre
- le nombre de pages
- la page courante

Votre classe doit contenir les méthodes suivantes:

- un ou plusieurs constructeurs
- une méthode d'accès (assesseur) qui retourne la valeur de la page courante
- une méthode `pageSuivante` qui fait passer la page courante à la page suivante.
- la méthode `toString` qui permettra le bon affichage d'un objet de la classe.

Écrivez ensuite une méthode `main` avec les instructions nécessaires à la création de deux objets de la classe `Livre` ainsi qu'à l'affichage de ces deux objets.

Exemple de méthode `main` demandée:

```
public static void main(String [] args) {
  Livre livre1 = new Livre("Victor Hugo","Les Miserables", 418);
  Livre livre2 = new Livre("Jules Verne", "L'ile mysterieuse", 331);
  System.out.println(livre1 +"\n"+ livre2);
}
```

Le code de la classe (une solution possible):

```
class Livre {

    private String auteur, titre;
    private int nbPages, courante;

    public Livre(String auteur, String titre, int nbPages) {
        this.auteur = auteur;
        this.titre = titre;
        this.nbPages = nbPages;
        courante = 1;
    }

    public int getPageCourante() {return courante;}

    public void pageSuivante() {
        if (courante < nbPages) courante++;
        else courante = 1;
    }

    public String toString() {
        return titre + " " + auteur + " " + nbPages + " " + courante;
    }
}
```