

INF1120 Programmation I

Hiver 2009

Examen Intra

14 mars 2009

CONSIGNES

SOLUTIONS

#1 _____ / 8

#2 _____ / 12

#3 _____ / 16

#4 _____ / 12

#5 _____ / 10

#6 _____ / 10

#7 _____ / 15

#8 _____ / 17

TOTAL

_____ / 100

IDENTIFICATION

PRÉNOM:

NOM:

CODE PERMANENT: _ _ _ _ _

SIGNATURE:

GROUPE:

PROFESSEUR:

COMMENTAIRES

Question 1 (8 points)

Répondez par Vrai ou Faux à chacune des affirmations suivantes (entourez V ou F).

- V** a) Il est permis d'insérer une instruction `switch` dans une instruction `if`.
- F** b) Un programme qui plante à l'exécution est un programme qui ne compile pas.
- F** c) Toute instruction `if` doit comporter la partie `else`.
- F** d) L'instruction `System.out.print(2 + 'A')` affichera la chaîne `2A`.
- V** e) Le compilateur Java peut compiler une classe qui ne contient aucune variable.
- V** f) Une constante déclarée dans une méthode ne peut être utilisée à l'extérieur de celle-ci.
- F** h) Les deux boucles suivantes sont équivalentes (ont le même effet sur la valeur de la variable x):

```
for ( int i = 5; i < 10; i++ ) {x = x + 1;}  
for ( int i = 10; i > 5; i++ ) {x = x + 1;}
```
- F** i) Les instructions à l'intérieur d'une boucle `for` s'exécutent toujours au moins une fois.

Question 2 (10 points : 1 point réponse)

Considérez les déclarations suivantes :

```
int      i = 6;  
double   d = 6.0;  
char     c = 'A';  
long     q = 3;  
float    t = 1.5f;  
boolean  b = false;
```

Indiquez le type et la valeur de chacune des expressions suivantes :

	Type	Valeur
a) <code>3.0 + 4 * 2</code>	double	11.0
b) <code>q * 3 + i / 2</code>	long	12
c) <code>t + d / i - q</code>	double	-0.5
d) <code>(q < 3) ((b == true) && (c != 'a'))</code>	boolean	false
e) <code>(byte)q + i * (short)(t - 1.5) + (byte)t</code>	int	4

Question 3 (15 points : 3 points par sous-question)

À la suite de l'exécution de chacune des boucles suivantes, indiquez la valeur affichée (les 5 blocs sont indépendants les uns des autres). S'il s'agit d'une boucle infinie, indiquez-le.

<p>a)</p> <pre>int n = -2; for (int r = 7; r > 3 ; r--) { n = n + (7 - r); } System.out.println(n);</pre>	<p>b)</p> <pre>int x = 3; int p = 9; for (int i = 9; i >= x; i--) { do { p = p - 1; x = x + 1; } while (p != x && i >= x - 1); p = p + i; } System.out.println(p);</pre>
<p>c)</p> <pre>int s = -11; int k = 13; while (k >= -s) { k = k - 2; s = s + 3; } System.out.println (k - s);</pre>	<p>d)</p> <pre>int x = 9; do { x = x + 3; } while (x >= 12); System.out.println(x);</pre>
<p>e)</p> <pre>int x = 5; int y = 0; do { while (x < 10) { x = x + 1; } y = y + 5; } while (y < 19); System.out.println (x + y);</pre>	<p>Réponses :</p> <p>a) 4</p> <p>b) 19</p> <p>c) boucle infinie</p> <p>d) boucle infinie</p> <p>e) 30</p>

Question 4 (12 points : 3,5; 2,5; 6)

Considérer l'algorithme suivant :

```
debut
  Afficher "Entrez un nombre entier : "
  Saisir n (un entier)
  si n est supérieur ou égal à 10000 ou négatif alors
    Afficher "Nombre non valide"
  sinon
    si n est strictement supérieur à 999 alors
      machin <- 2
    sinon
      machin <- 1
    fin si
    si n est strictement supérieur à 99 alors
      machin <- machin + 1
    fin si
    si n est strictement supérieur à 9 alors
      machin <- machin + 1
    fin si
    afficher machin
  fin si
fin
```

(a) Donnez ce qui sera affiché pour chacune des valeurs de n saisies suivantes :

Valeur saisie n	Affiché à la console
-5	Nombre non valide
0	1
7	1
654	3
78	2
8645	4
1000	4

(b) Dites en une phrase ce que l'algorithme fait en général (ne pas expliquer le code).

Affiche le nombre de chiffres que contient n si n est entre 0 et 9999 inclusivement.

(c) Traduire en Java le pseudo-code entre `debut` et `fin`

(Utilisez la page suivante)

```
System.out.print ( "Entrez un nombre entier :" );
n = Clavier.lireInt();
if ( n < 0 || n > 9999 ) {
    System.out.println ( "Nombre non valide" );
} else {
    if ( n > 999 ) {
        machin = 2;
    } else {
        machin = 1;
    }
    if ( n > 99 ) {
        ++machin;
    }
    if ( n > 9 ) {
        ++machin;
    }
    System.out.println ( machin );
}
```

Question 5 (10 points)

Vous devez écrire une méthode qui **affiche** le numéro du jour de la semaine déterminé à partir d'un jour de référence et d'un décalage dans le temps. Les jours de la semaine sont numérotés à partir de 1 et le numéro de dimanche est 1. Exemples d'utilisation :

q5(1,4) **affiche** : Le numéro du jour de la semaine est : 5
 indice : 4 jours après le premier jour de la semaine, donc le jour numéro 5

q5(7,9) **affiche** : Le numéro du jour de la semaine est : 2
 indice : 9 jours après le septième jour de la semaine, donc le jour numéro 2

q5(3,0) **affiche** : Le numéro du jour de la semaine est : 3
 indice : 0 jours après le troisième jour de la semaine, donc le jour numéro 3

q5(3,-5) **affiche** : Le numéro du jour de la semaine est : 5
 indice : 5 jours avant le troisième jour de la semaine, donc le jour numéro 5

q5(10,2) **affiche** : Le numéro du jour de référence n'est pas valide

```
/**
 * paramètre numeroJourDeLaSemaine le numéro du jour de la semaine (1 = dimanche, 7 = samedi)
 * paramètre decalage               le nombre de jours de décalage à partir du jour de la semaine.
 * Affiche le numéro correspondant au jour de la semaine en fonction de numeroJourDeLaSemaine et
 * de decalage.
 */
```

public static void à compléter

```

public static void q5(int numeroJourDeLaSemaine, int decalage) {
    int jour = numeroJourDeLaSemaine;
    if ( numeroJourDeLaSemaine < 1 || numeroJourDeLaSemaine > 7 ) {
        System.out.println( "Le numero de jour de la semaine n'est pas valide" ) ;
    } else {
        if (decalage >= 0) {
            for (int i = 1; i <= decalage; i++) {
                if (jour < 7) {
                    jour++;
                } else {
                    jour = 1;
                }
            }
            // OU: jour = ((numeroJourDeLaSemaine - 1 + decalage) % 7) + 1;
        } else {
            for (int i = -1; i >= decalage; i--) {
                if (jour > 1) {
                    jour--;
                } else {
                    jour = 7;
                }
            }
        }
        System.out.println(« Le jour de la semaine est : » + jour) ;
    }
}

```

Question 6 (10 points)

Trouvez les 5 erreurs de **compilation** qui se cachent dans le code source Java suivant. Pour chacune, indiquez le numéro de ligne où se situe l'erreur et la correction à apporter.

```
1.    public class ErreurCompilation {
2.
3.        public static void main (String[] args) {
4.
5.            short court = 3;
6.            double i = 4.0;
7.            FINAL float d = 1.0F;
8.            byte octet = 9.0;
9.
10.           do {
11.               for (int k = 9; k < k; k--)
12.                   i = d + (long)k;
13.               System.out.println(k + 'b');
14.           }
15.           octet = octet + "2";
16.           while (court < octet)
17.
18.       } /** main */
19.
20. } /* ErreurConpilation */
```

Réponses :

	No ligne	Correction
1)	5	mettre le FINAL en minuscule ou l'enlever
2)	6	mettre 9 au lieu de 9.0 ou double au lieu de byte
3)	12	Mettre 2 au lieu de « 2 »
4)	13	Il manque l'accolade fermante avant le while
5)	13	Il manque un point-virgule après la parenthèse

Question 7 (15 points)

À la suite de l'exécution des sélections suivantes, indiquez la valeur affichée par chacune des instructions

System.out.println.

```
public static void main (String[] params) {
    int x = 5;
    int i = 10;
    int k = 3;
    long a = 10;
    double d = 9.2;
    boolean valide = false;
    float m = 7.1f;
    String message = "rien";

    if ( x + d <= 14 ) {
        if ( a % x == 0 ) {
            a = 0;
        }
    }
    else
        a = a * 2;
}
a = a + 2;
System.out.println ( a );                                // reponse : 12

if ( !( x != 3 && !valide ) != true ) {
    message = "ok";
} else {
    message = "ko";
}
System.out.println ( message );                          // reponse : ok

switch ((int) m) {
    case 3 : m = x % i;
    case 1 : m = x + 1;
    case 2 : m = k + 5;
    case 4 : i = i / 2;
    case 5 : m = m + 9;
    case 6 : m = x + i;
    case 8 : m = m + 1;
    case 7 : m = m + (int) (m + 10) / 10;
    default : m = m + 1;
}
System.out.println(m);                                  // reponse : 9.1

if ( x == 6 ) {
} else {
    switch (x + 2) {
        case 7 : break;
        case 6 : x = x + 2;
        case 5 : x = x + 1;
    }
}

System.out.println ( x );                                // reponse : 5

if ( d - m != -m + d ) {
    d = 100; }
    m = 17.1f;
System.out.println ( (int) ( d + m ) );                  // reponse : 26
} // main
```

Question 8 (20 points)

Écrire un petit programme Java (méthode `main`) qui lit 3 nombres entiers: `depart`, `pas` et `limite`. Les valeurs de `pas` et de `limite` doivent être validées :

- La valeur de `pas` doit être non nulle.
- Si `pas` est positif, la valeur de `limite` doit être strictement supérieure à `depart`.
- Si `pas` est négatif, la valeur de `limite` doit être strictement inférieure à `depart`.

Le programme doit faire certains calculs à partir de ces valeurs. Voici les étapes :

Étape 1- saisies et validations :

Votre programme devra **saisir** la valeur de `depart` (qui n'a pas à être validée). Ensuite, le programme saisit et valide la valeur de `pas`. Lorsque `pas` est valide, le programme doit saisir et valider la valeur de `limite`.

Étape 2- calculs :

Une fois que les valeurs sont validées, le programme devra ajouter à `depart` la valeur de `pas` le plus de fois possible. La somme cumulée ne doit pas être supérieure à `limite` lorsque `pas` est positif ou être inférieure à `limite` dans le cas où `pas` est négatif.

Étape 3 - affichage des résultats :

Enfin, le programme devra afficher la valeur de la somme ainsi que le nombre de pas ajoutés.

Exemples :

depart	pas	limite	somme	nombre de pas	détail du calcul
10	3	23	22	4	10 + 3 + 3 + 3 + 3
45	-4	19	21	6	45 + -4 + -4 + -4 + -4 + -4 + -4
100	26	110	100	0	100
-16	2	99	98	57	16 + 2 + 2 + 2 + ... + 2

Prenez pour acquis que l'utilisateur entrera uniquement des valeurs entières (pas de nombres réels ou de caractères non numériques).

Premier exemple d'exécution (les valeurs saisies sont en caractère gras)

```
Veillez entrer la valeur de depart : 156
Veillez entrer la valeur de pas : 0
*** Le pas doit être non nul.
Veillez entrer la valeur de pas : 2
Veillez entrer la valeur de limite: 140
*** La valeur limite doit être supérieure à la valeur de départ.
Veillez entrer la valeur de limite : 156
*** La valeur limite doit être supérieure à la valeur de départ.
Veillez entrer la valeur de limite : 171
La somme est 170 et le nombre de pas est 7
```

Deuxième exemple d'exécution (les valeurs saisies sont en caractère gras)

```
Veillez entrer la valeur de depart : 678
Veillez entrer la valeur de pas : -13
Veillez entrer la valeur de limite : 699
*** La valeur limite doit être inférieure à la valeur de départ.
Veillez entrer la valeur de limite : 500
La somme est 509 et le nombre de pas est 13
```

***** N'utilisez ni les exceptions, ni les méthodes (sauf `main`) pour cette question.**

```

public class Question8 {

    public static void main ( String [] args ) {
        final String messInviteDepart =
            "Veuillez entrer la valeur de depart :";
        final String messInvitePas =
            "Veuillez entrer la valeur de pas :";
        final String messInviteLimite =
            "Veuillez entrer la valeur de limite :";
        final String messErreurPas =
            "*** Le pas doit être non nul.";
        final String messErreurLimiteSup =
            "*** La valeur limite doit être supérieure à la valeur de départ.";
        final String messErreurLimiteInf =
            "*** La valeur limite doit être inférieure à la valeur de départ.";

        // à compléter (veuillez utiliser les six constantes String précédentes)

        int depart;
        int pas;
        int limite;
        int somme;
        int nbPas;
        boolean pasEstPositif;

        // Saisies et validations

        System.out.print ( messInviteDepart );
        depart = Clavier.lireInt();

        System.out.print ( messInvitePas );
        pas = Clavier.lireInt();
        while ( pas == 0 ) {
            System.out.println ( messErreurPas );
            System.out.print ( messInvitePas );
            pas = Clavier.lireInt();
        }

        pasEstPositif = pas > 0;

        System.out.print ( messInviteLimite );
        limite = Clavier.lireInt();

        while ( ( pasEstPositif && limite <= depart ) ||
            ( !pasEstPositif && limite >= depart ) ) {
            if ( pasEstPositif ) {
                System.out.println ( messErreurLimiteSup );
            } else {
                System.out.println ( messErreurLimiteInf );
            }
            System.out.print ( messInviteLimite );
            limite = Clavier.lireInt();
        }
    }
}

```

```

// Calculs

somme = depart;
nbPas = -1;
while ( pasEstPositif && somme <= limite ||
        !pasEstPositif && somme >= limite ) {
    somme = somme + pas;
    nbPas++;
}
somme = somme - pas;

// Affichage des resultats

System.out.println ( "La somme est " + somme
                    + " et le nombre de pas est " + nbPas );

} // main

} // Question8

```