

INF1120 Programmation I
Automne 2004

Examen Intra
6 novembre 2004

CONSIGNES

Solutionnaire

IDENTIFICATION

PRÉNOM:

NOM:

CODE PERMANENT:

SIGNATURE:

GROUPE:

PROFESSEUR:

#1 _____ / 10

#2 _____ / 10

#3 _____ / 20

#4 _____ / 10

#5 _____ / 5

#6 _____ / 10

#7 _____ / 15

#8 _____ / 20

TOTAL

_____ / 100

COMMENTAIRES

Question 1 (10 points)

Dans toutes les questions qui suivent, on suppose que les variables ont été initialisées. Répondez par Vrai ou Faux à chacune des affirmations suivantes (entourez V ou F):

- V F a) On peut affecter une valeur de type `int` à une variable de type `short`.
- V F b) Une boucle `do ... while` effectue toujours au moins une itération.
- V F c) L'opérateur `cast` (conversion explicite de type) doit être utilisé lorsqu'on veut affecter une valeur de type `int` à une variable de type `float`.
- V F d) Toute instruction `if` doit comporter la partie `else`.
- V F e) L'expression booléenne `(max >= 9) || (max <= 11)` a toujours la valeur `true`.
- V F f) Il est interdit d'affecter une valeur de type `char` à une variable de type `long`.
- V F g) Si `n` est une variable de type `int` ayant une valeur positive ou nulle alors la boucle
- ```
for (int i = 0; i < n; i = i + 1) {
 System.out.println(i);
}
```
- effectuera toujours au moins `n` itérations.
- V    F    h)    L'instruction `if (n == n + 1) { a = b; } else { a = c; }` a toujours le même effet que l'instruction `a = b;`.
- V    F    i)    On peut changer le type d'une variable, de façon définitive, en utilisant l'opérateur `cast` (conversion explicite de type).
- V    F    j)    Les paramètres d'une méthode doivent toujours être initialisés avant que celle-ci ne soit appelée.

**Question 2 (10 points : 2 points par sous-question)**

À partir de ces déclarations :

```
int i = 3;
int j = 5;
long k = 2;
float x = 4.0F;
double y = 1.0;
char car1 = 'E';
char car3 = 'D';
boolean a = false;
```

donnez le type ainsi que la valeur de chacune des expressions suivantes. Si vous jugez que l'expression ne peut pas être évaluée, écrivez simplement ERREUR.

**a)**  $(5.0 + x / y)$

réponses :

double

## 9.0

**b)** `(j % i + (long) x)`

réponses :

long

6

c) `( (float) ( (int) car1 - (int) car3 ) * y )`

réponses :

double

1.0

**d)**  $(i > k > j)$

réponses :

**ERREUR**

e) `(!( j > k ) && ( a || ( ( x / (-y) ) > i ) ))`

réponses :

**boolean**

**false**

### Question 3 (20 points : 5 points par sous-question)

À la suite de l'exécution de chacune des boucles suivantes, indiquez la valeur affichée (les 4 blocs sont indépendants les uns des autres). S'il s'agit d'une boucle infinie, indiquez-le. Considérez que les déclarations suivantes sont faites.

```
int i;
int j;
int k;
```

|                                                                                                                                                                   |                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a)<br><br><pre>k = 4;<br/><br/>while ( k != 4 ) {<br/>    k = k + 2;<br/>}<br/><br/>System.out.println ( k );</pre>                                               | b)<br><br><pre>i = -3;<br/>j = 1;<br/><br/>for ( j = 2; j &lt; 8 ; j = j + 1 ) {<br/>    i = i + j;<br/>}<br/><br/>System.out.println ( i );</pre>                                                                                             |
| c)<br><br><pre>j = 18;<br/>k = 0;<br/><br/>do {<br/>    j = j - 2;<br/>    k = k + 4;<br/>} while ( j != k    k != 21 );<br/><br/>System.out.println ( j );</pre> | d)<br><br><pre>i = 7;<br/>j = 1;<br/><br/>do {<br/>    for ( int k = 3 ; k &lt;= i ; k = k + 1 ) {<br/>        j = j + 1;<br/>    }<br/>    i = i - 2;<br/>    j = j - 1;<br/>} while ( i % 4 != 1 );<br/><br/>System.out.println ( j );</pre> |

Réponses :

- |    |                       |
|----|-----------------------|
| a) | <b>4</b>              |
| b) | <b>24</b>             |
| c) | <b>boucle infinie</b> |
| d) | <b>5</b>              |

#### Question 4 (10 points : (a) 4; (b) 4; (c) 2)

Considérez les déclarations suivantes :

```
public final static int d = 13;

public static int bidon (int n) {
 int t = n;
 while (t >= d) {
 t = t - d;
 }
 return t;
} // bidon
```

(a) Si la valeur 44 est passée comme paramètre effectif lors de l'invocation, quelle valeur retourne `bidon` ?

**5**

(b) Si la valeur 65 est passée comme paramètre effectif lors de l'invocation, quelle valeur retourne `bidon` ?

**0**

(c) En général, que retourne la méthode `bidon` si la valeur passée comme paramètre effectif est positive ?

**Retourne le reste de la division entière du paramètre effectif par 13.**

## Question 5 (5 points)

Écrivez une méthode qui prend en paramètre un caractère et qui retourne :

- 1 si le caractère est une lettre minuscule,
- 2 s'il s'agit de l'un des deux caractères de ponctuation parmi les suivants : '?' ou '!' et
- 3 s'il s'agit de tout autre caractère.

```
public static int question5 (char c) {
 int reponse;
 if ((c >= 'a') && (c <= 'z')) {
 reponse = 1;
 } else if ((c == '!') || (c == '?')) {
 reponse = 2;
 } else {
 reponse = 3;
 }
 return reponse;
} // question5
```

## Question 6 (10 points)

La classe suivante comporte 5 erreurs de compilation. Identifiez-les et corrigez-les.

```
public class Question { // 1

 public static void main (String[] args) { // 2

 final double TAUX_TPS = 0.07 // 3
 final double TAUX_TVQ = 0.075; // 4

 double achat; // montant des achats effectués // 5
 double montantTps; // montant calculé de TPS // 6
 double montantTvq; // montant calculé de TVQ // 7
 double total; // total des achats et des taxes // 8

 System.out.println ("VENTE - Calculer les taxes de vente et le grand total"; // 9
 System.out.println (); // 10

 System.out.print ("Entrer le montant des achats : "); // 11
 Achat = Clavier.lireDouble(); // 12

 if (! ((achat = 0.0) || (achat < 0.0))) { // 13
 // bloc des calculs // 14
 montantTps = achat * TAUX_TVQ; // 15
 montantTvq = (achat + montantTps) * TAUX_TVQ; // 16
 total = achat + montantTps + montantTvq; // 17

 System.out.println (); // 18
 System.out.println ("Le montant des achats : " + achat); // 19
 System.out.println ("Le montant de la TPS : " + montantTps); // 20
 System.out.println ("Le montant de la TVQ : " + montantTvq); // 21
 System.out.println ("Le montant total : " + total); // 22

 else { // 23
 System.out.println ("Le montant des achats est nul ou négatif."); // 24
 } // 25
 } // 26
} // 27
```

(1) ligne 3 : il manque le point-virgule

```
final double TAUX_TPS = 0.07;
```

(2) ligne 9 : il manque la parenthèse fermante

```
System.out.println ("VENTE - Calculer les taxes de vente et le grand total");
```

(3) ligne 12 : la variable `Achat` devrait commencer par une lettre minuscule, car elle est déclarée ainsi.

```
achat = Clavier.lireDouble();
```

(4) ligne 13 : utilisation de l'opérateur d'affectation `=` au lieu de l'opérateur de comparaison `==`

```
if (! ((achat == 0.0) || (achat < 0.0))) {
```

(5) ligne 23 : il manque l'accolade fermante qui va avec l'ouvrante de la ligne 13

```
} else {
```

## 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 class Question7 {
 public static void main (String [] args) {
 int i = 5; int j = 1; int k = 8; int x = 4; int y = 3;

 if (i + j + x == k) {
 y = y * y;
 } else {
 y = y * 2;
 }

 System.out.println (y); // réponse #1 6

 switch (k - 2) {
 case 0 :
 j = i + k;
 break;
 case 2 :
 j = i + k;
 break;
 case 3 :
 j = x - j;
 break;
 case 6 :
 j = k - 2;
 break;
 case 5 :
 j = x + 3;
 break;
 case 8 :
 j = x + 3;
 break;
 default :
 j = 0;
 break;
 }

 System.out.println (j); // réponse #2 6

 if ((k >= 7) && (x <= 3)) {
 i = i / 2;
 k = 4;
 } else {
 i = i - 2;
 k = x - 4;
 }

 System.out.println (i); // réponse #3 3

 System.out.println (k); // réponse #4 0

 if ((i - j == 3) || (k != 1)) {
 y = (j - 1) * 2;
 } else {
 y = (x + 2) * 3;
 }

 System.out.println (y); // réponse #5 10
 } // main
} // Question7
```



## Question 8 (20 points)

Écrivez un petit programme Java (méthode `main`) qui devra lire (saisir) au clavier des valeurs entières comprises entre -50 et 50 inclusivement. La valeur spéciale 999 sera entrée par l'utilisateur du programme pour indiquer la fin de l'entrée des données. Les valeurs entrées (qu'on supposera entières) devront être validées : si l'utilisateur entre une donnée non valide, le programme doit afficher un message d'erreur. Puis le programme affichera la moyenne des valeurs entrées en excluant le 999 (moyenne = somme des valeurs divisée par le nombre de valeurs). Cette moyenne doit être réelle. Ensuite le programme doit indiquer si au moins une valeur strictement inférieure à -40 a été entrée (parmi les valeurs valides). Si aucune donnée valide n'est saisie, sauf 999, la moyenne est 0.

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

Voici un exemple d'exécution :

```
Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : -60
*** Le nombre doit être entre -50 et 50 inclusivement.
Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : -40
Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : 28
Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : -45
Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : 999
```

La moyenne est : -19.0

Il y a au moins une valeur strictement inférieure à -40.

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

```
public class Question8 {

 public static void main (String [] args) {
 String messageInvite =
 "Entrez un nombre entier entre -50 et 50 inc. (999 pour terminer) : ";
 String messageErreur =
 "*** Le nombre doit être entre -50 et 50 inclusivement.";

 final int sentinelle = 999;

 int somme = 0;
 int nbValeurs = 0;
 int valeurLue;
 double moyenne;
 boolean fini = false;
 boolean ilExisteValeurInferieureMoins40 = false;

 while (!(fini)) {
 System.out.print (messageInvite);
 valeurLue = Clavier.lireInt();
 if (valeurLue == sentinelle) {
 fini = true;
 } else if ((valeurLue < -50) || (valeurLue > 50)) {
 System.out.println (messageErreur);
 } else { // valeur valide
 ilExisteValeurInferieureMoins40 = ilExisteValeurInferieureMoins40 ||
 (valeurLue < -40);

 nbValeurs = nbValeurs + 1;
 somme = somme + valeurLue;
 }
 } // boucle de lecture des valeurs
 }
}
```

```

// calcul de la moyenne

if (nbValeurs == 0) {
 moyenne = 0;
} else {
 moyenne = (double) somme / nbValeurs;
}

System.out.println ();
System.out.println ("La moyenne est : " + moyenne);
if (ilExisteValeurInferieureMoins40) {
 System.out.println ("Il y a au moins une valeur strictement inférieure" +
 " à -40.");
}

} // main

} // Question8

```