

Name/Nom :

Klasse/Classe:

TEST SEMESTRIEL – SEMESTERPRÜFUNG

Informatique 1 | Informatik 1

Anweisung / Consigne :

Lesen Sie die Fragen gut durch und beantworten Sie diese **leserlich** auf den Aufgabenblättern. Für diese Prüfung dürfen Sie 2 Blätter (mit Vor- und Rückseite) mitnehmen, jedoch keine elektronischen Hilfen.

Tipp: Verlieren Sie bei einzelnen Fragen nicht zu viel Zeit. Beantworten Sie zuerst die Fragen, die Ihnen keine Probleme stellen, und kommen Sie später auf die für Sie schwierigeren Fragen zurück. Die Skala ist unverbindlich

Lisez attentivement la donnée et répondez de manière **lisible** aux questions. Vous avez droit pour cet examen à un aide-mémoire de 4 pages (2 feuilles recto-verso). Aucun moyen électronique n'est permis.

Un conseil : ne restez pas bloqués sur une question. Répondez tout d'abord aux questions avec lesquelles vous êtes à l'aise et revenez ensuite aux questions posant problème. Le barème indiqué est indicatif.

Question	Points	Bonus	Score
Short questions	10	0	
Working with strings	12	0	
Home Lighting Automation ?	6	0	
Some functions	6	0	
Des petits tableaux	8	0	
Gotta catch 'Em all	8	0	
Total:	50	0	

This exam has 6 questions, for a total of 50 points.

Rev 1.01

Question 1 – Short questions (10 points)

Cette question est séparée en plusieurs exercices indépendants. Le nombre de points pour chaque exercice est indiqué dans la marge.

[2 Pt] (a) Soient les déclarations suivantes :

```
int a = 5;
int b = 6;
double c = 7.0;
double d = 8.0;
String e = "karaoke";
```

Donnez le **type** et la **valeur** des expressions suivantes :

1) a+b

1) _____

2) a/b

2) _____

3) a + b + e

3) _____

4) e.charAt(e.length()-3) + "" + e.charAt(0)

4) _____

[2 Pt] (b) Soit le code suivant:

```
1  class Foo {
2      public static int a;
3      private int b;
4
5      public static void fun1()
6      {
7          // Loc1
8      }
9  }
10
11 class Bar {
12     private static int c;
13     public int d;
14
15     private void fun2()
16     {
17         Foo inst1 = new Foo();
18
19         // Loc 2
20     }
21 }
```

Si les instructions suivantes sont écrites à la position Loc1, sont-elles valides ?

b = 3;

True	False
<input type="checkbox"/>	<input type="checkbox"/>

a = 4;

True	False
<input type="checkbox"/>	<input type="checkbox"/>

Bar.d = 2;

True	False
<input type="checkbox"/>	<input type="checkbox"/>

Même question pour la position Loc2 :

d = 3;	True <input type="checkbox"/>	False <input type="checkbox"/>

c = 4;	True <input type="checkbox"/>	False <input type="checkbox"/>

inst1.b = 5;	True <input type="checkbox"/>	False <input type="checkbox"/>

[2 Pt] (c) Cochez **la (ou les)** affirmations correctes (il peut donc y avoir plus d'une réponse).

- 1) `bar = ' ' + 12.0;` est une instruction possible si
 - `bar` de type `String`
 - `bar` est déclaré comme `static double`
 - `bar` est `final`
- 2) Après l'exécution de `String[] x = new String[10];`
 - `x` contient un mot de 10 lettres
 - `x` peut contenir au plus 10 chaînes de caractères
 - `x` est un tableau vide qui contient `null` dans chaque case
- 3) `Foo.bar` est valide si
 - `bar` est statique
 - `Foo` est une classe privée
 - `bar` est une instance de `Foo`
- 4) Un objet
 - peut contenir d'autres objets
 - est une instance d'une classe
 - peut être un attribut statique d'une classe

[4 Pt] (d) Vrai ou faux ?

Soit la déclaration suivante <code>public int bar()</code> se trouvant dans la classe <code>Bar</code> . Il est possible d'appeler cette méthode avec l'instruction <code>Bar.bar()</code> .	True <input type="checkbox"/>	False <input type="checkbox"/>

Il est possible de faire une boucle <code>for</code> infinie.	True <input type="checkbox"/>	False <input type="checkbox"/>

Un tableau possède une taille variable.	True <input type="checkbox"/>	False <input type="checkbox"/>

Un constructeur peut retourner une valeur.	True <input type="checkbox"/>	False <input type="checkbox"/>

<code>(new Seat[10])[0] != null</code>	True <input type="checkbox"/>	False <input type="checkbox"/>

Il est possible de déclarer deux méthodes ayant le même nom, le même type et le même nombre de paramètres pour autant que le type de la valeur de retour soit différent.	True <input type="checkbox"/>	False <input type="checkbox"/>

Pour pouvoir utiliser <code>FunGraphics</code> correctement, il est nécessaire de créer une instance de cette classe.	True <input type="checkbox"/>	False <input type="checkbox"/>

Lorsque l'on souhaite connaître la taille d'un tableau, on appelle la méthode <code>length</code> appartenant à la classe <code>tableau</code> .	True <input type="checkbox"/>	False <input type="checkbox"/>

Question 3 – Home Lighting Automation ? (6 points)

Afin de mettre à profit vos nouvelles compétences en informatique, vous souhaitez ajouter un peu de domotique chez vous. Pour ce faire, vous décidez de réaliser un programme permettant d'automatiser l'éclairage dans une pièce.

Écrivez une classe `HomeAutomation` contenant les attributs nécessaire à la gestion de cette automation. Notamment, vous souhaitez avoir les états possibles pour la lumière : `OFF`, `CHILL`, `READ`, `EAT`. Deux interrupteurs poussoir déclenchent les méthodes suivantes :

1. `void button1()`, permet d'éteindre la lumière mais pas de l'allumer.
2. `void button2()`, lorsque le système est allumé ce bouton permet de changer de mode selon la séquence ci-dessous (qui n'est pas un diagramme d'état). Lorsque le système est éteint, ce bouton sert à rallumer le système qui se retrouve alors dans l'état qu'il avait lorsqu'on l'a éteint.

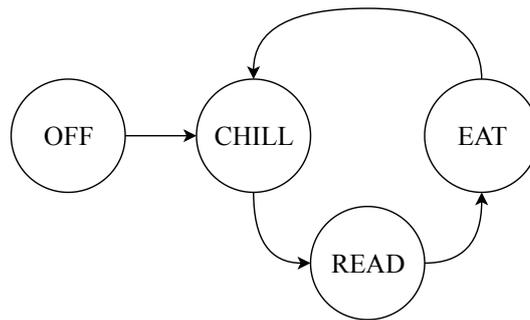


Figure 1 – Diagramme de séquence deuxième bouton

Lors de la création des instances de cette classe, il doit être possible de spécifier l'état dans lequel se trouve. Si aucun état n'est spécifié à la création de l'instance, l'éclairage doit se trouver en mode `CHILL`.

Ajoutez également *tout le code nécessaire* afin que l'exécution du main donne le même résultat que ci-dessous.

```

public static void main(String args[]) {
    HomeAutomation hal = new HomeAutomation();

    System.out.println(hal); → affiche sur la console "Current state : CHILL"
    hal.button1();
    System.out.println(hal); → affiche sur la console "Current state : OFF"
    hal.button1();
    System.out.println(hal); → affiche sur la console "Current state : OFF"
    hal.button2();
    System.out.println(hal); → affiche sur la console "Current state : CHILL"
    hal.button2();
    System.out.println(hal); → affiche sur la console "Current state : READ"
    hal.button1();
    System.out.println(hal); → affiche sur la console "Current state : OFF"
    hal.button2();
    System.out.println(hal); → affiche sur la console "Current state : READ"
    hal.button2();
    System.out.println(hal); → affiche sur la console "Current state : EAT"
    hal.button2();
    System.out.println(hal); → affiche sur la console "Current state : CHILL"
}
  
```


Solution:



Fin|Ende
