

SERIE 6– SOLUTION

Informatique 1 | Informatik 1

Question | Frage 1

bar ist eine Variable `int` von Klasse *Foo*. Wie deklarieren Sie *bar* (z.B. `private static int toto;`), so dass *bar* :
bar est une variable `int` appartenant à la classe *Foo*. Comment déclarez-vous *bar* (par exemple `private static int toto;`), afin que *bar* :

- (a) *ein int ist, konstant, und dessen Wert ist zwischen alle Instanzen von Foo geteilt wird. Es darf auch nicht von eine andere Klasse zugreifbar sein*
 soit un `int` dont la valeur constante qui est partagé par toutes les instances de *Foo* et ne soit pas accessible depuis une autre classe

(a) `private static final int bar;`

- (b) *ein float ist, die von überall zugreifen kann und denen Wert unabhängig von die Instanzen von Foo ist.*
 est un `float` accessible de partout et dont la valeur est indépendante des instances de *Foo*.

(b) `public static float bar;`

- (c) *ein char ist, die nur in der Klasse Foo zugreifen werden kann und denen Wert in die Instanzen von Foo gespeichert ist.*
 est un caractère accessible uniquement depuis la classe *Foo* et dont la valeur dépend des instances de *Foo*.

(c) `private char bar;`

- (d) *ein double mathematische Konstante die überall zugegriffen werden kann ist.*
 est un `double` correspondant à une constante mathématique et utilisable partout.

(d) `public static final double BAR;`

- (e) *ein int ist, die als Zähler von Instanzen verwendet wird. Diese Variable ist nur in der Klasse selbst sichtbar.*
 est un entier utilisé comme un compteur du nombre d'instances d'une classe, utilisable uniquement depuis l'intérieur de la classe.

(e) `private static int bar;`

Question | Frage 2

Printer ist eine Klasse mit zwei Variablen wie folgt:

Printer est une classe avec deux variables telle que :

```
public class Printer{
    private static int totalNumberOfPrinters=0;
    private String printerName;
}
```

Schreiben Sie den Prototyp von: Ecrivez le prototype de :

- (a) *ein Konstruktor, der den Wert von dem Attribut printerName von einem Parameter String kopiert.*
 Un constructeur qui copie le `String` donné en paramètre dans l'attribut `printerName`.

(a) `Printer(String name)`

- (b) *Eine Methode setPrinterName, die ein als Parameter gegeben String in das Attribut printerName kopiert und true zurückgibt, wenn dieser Parameter verschieden von dem alten Wert von printerName ist, false ansonsten. Diese Methode muss von aussen der Klasse verwenden kann.*

Une méthode `setPrinterName` copiant le `String` donné en paramètre dans l'attribut `printerName` et qui retourne `true` si le paramètre est différent de la valeur précédente de `printerName`, et `false` autrement. Cette méthode doit être accessible depuis l'extérieur de la classe.

(b) `public boolean setPrinterName(String name)`

- (c) Eine Methode `getTotalNumberOfPrinters`, ohne Eingabewert, die die Anzahl von Printer Instanzen zurückgibt. Diese Methode soll von aussen der Methode verwenden kann.
 Une méthode `getTotalNumberOfPrinters`, sans paramètres et qui retourne le nombre total d'imprimantes existant. Cette méthode doit être accessible en dehors de la classe.

(c) public static int getTotalNumberOfPrinters()

- (d) Eine Methode `toLowerCase`, ohne Parameter und ohne Rückgabewert. Diese Methode ändert die Name von die Instanz so dass die Name klein geschrieben ist. Diese Methode ist von aussen der Klasse nicht sichtbar.
 Une méthode `toLowerCase` sans paramètre qui ne retourne rien. Cette méthode change la valeur de `printerName` en la mettant en minuscules. Cette méthode n'est pas accessible en dehors de la classe.

(d) private void toLowerCase()

Question | Frage 3

- (a) Was ist den Unterschied zwischen einer Statische- und einer Instanz- Variabel ?
 Quelle est la différence entre une variable statique et une variable d'instance ?

Solution: Each object of a class has its own copy of each instance variable, but a single copy of each static variable is shared by all objects. A static variable is present only once in memory.

- (b) Dürfen Sie einer Instanzvariable in einer statische Methode benutzen ? Und in einer nicht-statische Methode ?
 Pouvez-vous utiliser une variable d'instance dans une méthode statique ? Et dans une méthode non statique ?

Solution: No, you cannot use an instance variable in the definition of a static method of the same class (because in a static method you wouldn't know to which instance the variable relates to).
 2nd question: yes, you can use an instance variable in the definition of a nonstatic method of the same class.

- (c) Warum ist eine Methode `private` deklariert ? Pourquoi déclare-t-on une méthode `private` ?

Solution: The standard use of private method is when a method is being used solely as a helping method in the definition of other methods.