

SERIE 4– SOLUTION

Informatique 1 | Informatik 1

Part 1 - Function prototypes

Question | Frage 1

Geben Sie den Prototypen der folgenden Funktionen an :

Donner le **prototype** des fonctions suivantes :

- (a) Erheben eine reelle Zahl des Typs `float` in eine Dreierpotenz.¹
Fonction de mise au cube d'un nombre réel de type `float`².

Solution:

```
[public] [static] float cube(float x)
```

- (b) x^y mit x und y als Ganzzahlen; der Rückgabewert ist vom Typ `double`.
Fonction calculant x^y avec x et y entiers, retournant un `double`.

Solution:

```
[public] [static] double power(int x, int y)
```

- (c) Statische Funktion, genannt `f1`, die nichts zurückgibt und ohne Argumente.
Une fonction nommée `f1` ne retournant rien et ne prenant aucun argument.

Solution:

```
[public] static void f1()
```

- (d) Eine Funktion, welche die Anzahl Buchstaben eines Worts zurückgibt, das als Argument übergeben wurde.
Une fonction retournant le nombre de lettres d'un mot passé en argument.

Solution:

```
[public] [static] int wordLength(String s)
```

- (e) Eine Funktion, genannt `f2`, die zwei Ganzzahlen und ein `String` als Eingabewert nimmt und einen Booleschen Wert zurückgibt.
Fonction nommée `f2` prenant deux entiers et un `String` en argument et retournant un `booléen`.

Solution:

```
[public] [static] boolean f2(int arg1, int arg2, String arg3)
```

Question | Frage 2

Was ist der Typ der Argumente der Funktion `f`, wenn keine automatische Typumwandlung benützt wird ?

Quel est le type des arguments de la fonction `f` dans les cas suivants si le mécanisme de la promotion de type automatique n'est pas utilisé ?

- (a) `f1("Hello")`

(a) _____ **String** _____

- (b) `f2(2.5, 3)`

(b) _____ **double, int** _____

¹In die Lösungen, was zwischen `[]` liegt ist fakultativ.

²Notez que dans les solutions imprimées, le code imprimé entre crochets `[]` est optionnel.

(c) f3((int) 2.5, (int) 3.2f)

(c) _____ **int, int** _____

(d) f4(true, 3f)

(d) _____ **boolean, float** _____

(e) f5("2.5f", 3)

(e) _____ **String, int** _____

(f) f6('a', 1e3f)

(f) _____ **char, float** _____

Part 2 - Function definitions

Question | Frage 3

Geben Sie **den Code** für die folgenden Funktionen:

Donnez le code **complet** des fonctions suivantes :

- (a) Eine Funktion, die eine Ganzzahl (*param*) als Argument nimmt. Wenn *param* gleich 1, 2 oder 3 ist, soll die Funktion die englische Zahl zurückgeben (z.B. '1' ergibt one, '2' ergibt two). Wenn *param* ungleich 1, 2 oder 3 ist, soll die Funktion den Text "*X is invalid*" zurückgeben, wobei *X* durch den Wert von *param* ersetzt werden soll. Z.B. gibt der Parameter 6 das Resultat "*6 is invalid*" zurück.

Fonction prenant un nombre entier, nommé *param* en argument. Si *param* est égal à 1, 2 ou 3, la fonction retourne le texte anglais qui correspond au chiffre (par ex. '1' donne *one*, '2' donne *two*). Si *param* est différent de 1, 2 ou 3, la fonction retourne le texte "*X is invalid*", avec *X* remplacé par *param*. Par exemple, le paramètre 6 retourne le résultat "*6 is invalid*".

Solution:

```
1 public static String askAndTransformToText(int param){
2     String textValue = "";
3
4     switch (param) {
5     case 1:
6         textValue = "one";
7         break;
8     case 2:
9         textValue = "two";
10        break;
11    case 3:
12        textValue = "three";
13        break;
14    default:
15        textValue = param + " is invalid";
16        break;
17    }
18
19    return textValue;
20 }
```

- (b) Summe von aller geraden positiven Zahlen, die kleiner als die Zahl *maxValue* sind, welche als Argument übergeben wird.

Retournant la somme des nombres pairs positifs strictement plus petits qu'un nombre *maxValue* passé en argument.

Solution:

```

1 public static int sumOdd(int maxValue) {
2     int sum = 0; // To hold the sum
3     int currentValue = 0; // Holds the value we are considering
4
5     /*
6      * While the number we want to add is still
7      * smaller than maxValue
8      */
9     while (currentValue < maxValue) {
10
11         /*
12          * If we have an odd number, add it to the
13          * result we will return
14          */
15         if (currentValue % 2 == 0)
16             sum += currentValue;
17
18         currentValue++;
19     }
20
21     // Returns the calculated sum
22     return sum;
23 }

```

(c) Somme des n premiers termes de la série harmonique, d.h.

Fonction calculant la somme des n premiers termes de la série harmonique, c'est-à-dire

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{i=1}^n \frac{1}{i} \quad (1)$$

Solution:

```

1 public static double harmonicSerie(int n) {
2     double sum = 0;
3
4     for (int i = 1; i <= n; i++) {
5         sum += 1.0 / i;
6     }
7
8     return sum;
9 }

```

(d) Écrivez une fonction, qui retourne le nombre d'années bissextiles ayant eu lieu entre l'an 5 AD et l'année qui a été passée en paramètre. Les années bissextiles sont les années:

- un multiple de 400
- un multiple de 4, mais pas de 100 (sauf si multiple de 400).

Exemple: 2000 est un multiple de 400, 1900 n'est pas un multiple de 400, mais un multiple de 100, mais pas de 400. Utilisez plusieurs fonctions si nécessaire.

Écrivez une fonction `nLeapYears` qui retourne le nombre d'années bissextiles ayant eu lieu entre l'an 5 après J.-C. et l'année qui a été passée en paramètre. Les années bissextiles sont les années:

- Multiples de 400
- Les années multiples de 4, mais pas multiples de 100 (sauf si multiples de 400).

Exemple: 2000 est une année bissextile (multiple de 400), alors que 1900 ne l'était pas (multiple de 100, mais pas de 400). *Note:* Vous pouvez écrire plus qu'une fonction si nécessaire.

Solution:

```
1  (static public) boolean isLeap(int year){
2      if(year % 400 == 0 || ((year % 4 == 0) && (year % 100 != 0)))
3          return true;
4      else
5          return false;
6  }
7
8  (static public) int nLeapYears(int until){
9      int total = 0;
10
11     for(int i = 5; i < until; i++)
12     {
13         if(isLeap(i))
14             total++;
15     }
16
17     return total;
18 }
```

Question | Frage 4

Warum kompiliert der folgende Code nicht ?

Pourquoi les codes suivants ne compilent-ils pas ?

(a) Fonction 1

```
1  int foo(int a){
2    if(a > 5)
3      return true;
4    else
5      return false;
6  }
```

Solution: Der erwartete Rückgabewert ist von Typ *int* und im Funktionskörper sind *boolean* zurückgegeben.
La valeur de retour attendue est de type *int* alors que dans le corps de la fonction sont retournés des valeurs de type *boolean*.

(b) Fonction 2

```
1  int bar(boolean isValid){
2    if(!isValid){
3      return 1;
4    }
5  }
```

Solution: Wenn *isValid* ist wahr, keine Rückgabewert ist zurückgegeben.
Lorsque *isValid* est vrai, aucune valeur n'est retournée.

Question | Frage 5

Was macht der folgende Code ?

Expliquer ce que réalise le code suivant en quelques lignes :

```
1  public static boolean foobar(int a, int b, boolean c){
2    boolean result;
3
4    if(c){
5      result = a > b ? true:false;
6    }
7    else{
8      result = a > b ? false:true;
9    }
10
11   return result;
12 }
```

Solution: Die Variable *c* wählt ob der Rückgabewert ist $a > b$ oder $a < b$. *c* funktioniert wie einen Multiplexer.

La variable *c* détermine si la valeur de retour est $a > b$ ou $a < b$ comme le ferait un multiplexeur dont l'entrée de commande serait *c*.