
TEST SEMESTRIEL – SEMESTERPRÜFUNG

SOLUTION

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	6	0	
Thinking a bit	3	0	
Functions everywhere	9	0	
Phone numbers ☎	6	0	
Reaching the Apex	7	0	
Washing it with a laundry 🧺	9	0	
Total:	40	0	

This exam has 6 questions, for a total of 40 points.

Rev 1.1 Ω

Question 1 – Short questions (6 points)

Diese Frage ist in unabhängige Aufgaben unterteilt. Die Anzahl Punkte jeder Aufgabe ist am Rand vermerkt. Cette question est séparée en plusieurs exercices indépendants. Le nombre de points pour chaque exercice est indiqué dans la marge.

[1 Pt] (a) Gegeben sei: | Soit le code suivant :

```
1 long fooBar = Bar.foo;
```

Deklariere Sie die Variable `foo` auf zwei Mögliche Arten:

Donnez deux déclarations possibles de `foo`.

Solution:

```
1 static int foo;
2 static long foo;
```

[2 Pt] (b) Was zeigt der folgende Code in der Konsole an? Qu'affiche le code suivant sur la console ?

```
1 char[] foo = {'a', 'c', 'a', 'o', 'r', 'e', 'z', 'c', 't'};
2
3 int i = 1, j = 2;
4
5 do {
6     if (i == 6) {
7         i += j;
8         continue;
9     }
10
11     System.out.print(foo[i]);
12
13     if (i % 3 == 0) {
14         j = 1;
15         i += j;
16         System.out.print(foo[i]);
17     } else
18         i += j;
19
20 } while (i < 9);
```

Solution: correct

[3 Pt] (c) Wahr oder falsch? | Vrai ou faux ?

Die für ein Array festgelegte Anzahl Elemente kann nachträglich nicht verändert werden.

Le nombre d'éléments d'un tableau ne pas être modifié une fois le tableau créé.

True | False

|

Eine Funktion kann nur ein Ergebnis eines einzigen Datentyps zurückgeben.

Une fonction ne peut retourner qu'un seul résultat d'un seul type.

True | False

|

Die Grösse eines Arrays kann mit Hilfe der Methode `length()` bestimmt werden.

Si l'on a un tableau, on peut obtenir sa taille à l'aide de la méthode `length()`.

True | False

|

Man muss eine Instanz erzeugen, um eine statische Methode aufrufen zu können.

On doit créer une instance pour pouvoir appeler une méthode statique.

True | False

|

Klasse un Objekt sind Synonyme.
Classes et objets sont des synonymes.

True | False
 |

Um eine Instanz einer Klasse erzeugen zu können, muss in jedem Fall der Konstruktor der Klasse aufgerufen werden.
Pour créer une instance d'une classe, le constructeur de cette classe va forcément être appelé.

True | False
 |

Question 2 – Thinking a bit (3 points)

Grundsätzlich gilt: Eine Funktion kann nur ein einziges Ergebnis eines einzigen Typs zurückgeben.

Par principe, une fonction peut retourner un seul résultat d'un type donné.

- [1 Pt] (a) Wie können Sie dennoch mehrere Werte unterschiedlicher Datentypen als Rückgabewert einer Funktion zurückgeben?

Comment faire pour retourner plusieurs valeurs de types différents depuis une fonction ?

Solution: We can write a class to store the different values we want to return and return an object of this type from the function.

- [2 Pt] (b) Geben Sie ein Beispiel, welches zeigt, wie ein `int`, ein `double` sowie ein `String` von einer Methode zurückgegeben werden kann, die keine Parameter erhält.

Donnez un exemple complet démontrant comment retourner à la fois un `int`, un `double` ainsi qu'un `String` depuis une fonction qui ne prend aucun argument.

Solution:

```

1  class Result{
2      int a;
3      double b;
4      String c;
5  }
6
7  Result foo(){
8      Result r;
9      r.a = 3;
10     r.b = 5.0;
11     r.c = "Hello";
12     return r;
13 }
```

Question 3 – Functions everywhere (9 points)

- [2 Pt] (a) Schreiben Sie eine Funktion namens `count`, die zurückgibt, wie oft der Buchstabe 'a' in einer als Argument erhaltenen Zeichenkette vorkommt.

Écrivez une fonction nommée `count` qui retourne combien de fois la lettre 'a' apparaît dans une chaîne de caractères passée en argument.

Solution:

```

1 public static int count(String s) {
2     int res = 0;
3     for(int i = 0; i < s.length(); i++) {
4         if(s.charAt(i) == 'a')
5             res++;
6     }
7
8     return res;
9 }
```

- [3 Pt] (b) Implementieren Sie die Funktion `String unique(String s)`, die einen `String` zurückgibt, der jeden in `s` vorkommenden Buchstaben einmalig in alphabetischer Reihenfolge zurückgibt. Gehen Sie davon aus, dass die Zeichenkette `s` nur aus kleinen Buchstaben besteht; Leerschläge werden für die Ausgabe nicht berücksichtigt.

Écrivez la fonction `String unique(String s)` qui retourne un `String` contenant les lettres contenues dans `s`, dans l'ordre alphabétique et une seule fois. On part du principe que `s` ne contient que des lettres minuscules (on ne tiendra pas compte des espaces).

Beispiel | Exemple :

```

1 unique("bbaaac") // Returns "abc"
2 unique("the quick brown fox jumps over the lazy dog") // Returns "abcdefghijklmnopqrstuvwxyz"
```

Solution:

```

1 public static String unique(String input) {
2     String output = "";
3
4     for (char c = 'a'; c <= 'z'; c++) {
5         // Alternative 1
6         for (int i = 0; i < input.length(); i++) {
7             if (c == input.charAt(i)) {
8                 output += c;
9                 break;
10            }
11        }
12
13        // Alternative 2
14        if(input.contains(c))
15            output += c;
16    }
17
18    return output;
19 }
```

- [4 Pt] (c) Implementieren Sie eine Funktion, die ein zweidimensionales String-Array als Parameter erhält und die den kürzesten String dieses Arrays zurück gibt. Sollte es zwei Strings gleicher Länge geben, gibt die Funktion den ersten gefundenen der beiden zurück, unabhängig von der Richtung (Zeilen, Spalten), in welcher man das Array durchläuft. Gehen Sie davon aus, dass das Array mindestens ein Element enthält und immer zwei Dimensionen hat.

Écrivez une fonction qui reçoit un tableau bidimensionnel de String en argument et qui retourne le String le plus court de ce tableau. Si deux Strings possèdent la même longueur, on retournera le premier String de cette longueur rencontré (peu importe que l'ordre de parcours du tableau soit par ligne ou colonne). On part du principe que le tableau possède au moins un élément et toujours deux dimensions.

Beispiel | Par exemple :

```

1 String[][] check =
2 {
3     { "vote", "foo", "bar" },
4     { "failure", "Donald", "president" }
5 };
6 System.out.println(minString(check));
7
8 → Prints foo on the console

```

Solution:

```

1 public static String minString(String[][] t) {
2     int minLength = t[0][0].length();
3     String minString = t[0][0];
4
5     for (int i = 0; i < t.length; i++) {
6         for (int j = 0; j < t[0].length; j++) {
7             String current = t[i][j];
8             if (current.length() < minLength) {
9                 minLength = current.length();
10                minString = current;
11            }
12        }
13    }
14    return minString;
15 }

```

Question 4 – Phone numbers ☎ (6 points)

[4 Pt] (a) Schreiben Sie eine Klasse *Phone*, welche die Methode *testNumber* enthält, die überprüft, ob eine als Parameter übergebene Zeichenkette mit 10 Zeichen eine Mobiltelefonnummer ("0788011699") darstellen kann.

1. die Zeichenkette muss genau 10 Zeichen umfassen
2. sie muss mit einer 0 beginnen
3. sie darf nur aus Ziffern bestehen

Falls dies der Fall ist, soll die Nummer in folgender Form zurückgegeben werden: "+4178/801.16.99". Für ungültige Nummern gibt die Methode *null* zurück.

Hint: Sie dürfen die Methode der Klasse *String* verwenden, deren Prototyp *String substring(int beginIndex, int endIndex)* lautet und die den Substring der ursprünglichen Zeichenkette von Position *beginIndex* (eingeschlossen) bis *endIndex* (nicht eingeschlossen) umfasst. Beispiel für die *substring* Methode:

```
1 "hamburger".substring(4, 8) // Returns "urge"
2 "smiles".substring(1, 5) // Returns "mile"
```

Écrivez une classe *Phone* contenant une méthode *testNumber* qui transforme une chaîne de caractères contenant 10 chiffres (par exemple "0788011699") dans un autre format.

1. La chaîne doit contenir 10 chiffres exactement
2. Elle doit impérativement commencer par 0
3. Elle ne doit contenir que des chiffres

Si la chaîne est correcte, vous devez la transformer dans le format "+4178/801.16.99". Si la chaîne est incorrecte, la fonction doit retourner *null*.

Conseil: Vous pouvez utiliser la méthode existant dans la classe *String* et dont le prototype est *String substring(int beginIndex, int endIndex)* qui retourne la sous-chaîne commençant à la position *beginIndex* (compris) à *endIndex* (non compris) de la chaîne originale. Exemple méthode *substring* :

```
1 "hamburger".substring(4, 8) // Returns "urge"
2 "smiles".substring(1, 5) // Returns "mile"
```

Solution:

```
1  class Phone {
2      public static String testNumber(String number) {
3          String res = "";
4          int length;
5          char c;
6
7          length = number.length();
8
9          if (length != 10 || number.charAt(0) != '0')
10             return null;
11
12         for (int i = 0; i < length; i++) {
13             c = number.charAt(i);
14
15             if (c < '0' || c > '9')
16                 return null;
17         }
18
19         res += "+41";
20         res += number.substring(1, 3);
21         res += "/";
22         res += number.substring(3, 6);
23         res += ".";
24         res += number.substring(6, 8);
25         res += (".");
26         res += number.substring(8, 10);
27         return res;
28     }
29 }
```

Solution:

- [2 Pt] (b) Implementieren Sie ein `main`, welches die Verwendung der Klasse aufzeigt. Der Anwender gibt eine Zeichenkette mit der Nummer in der Konsole ein. Verwenden Sie die erstellte Methode und geben Sie die Nummer aus, wenn sie gültig ist, im anderen Fall den Text "Invalid number".

Faites un exemple complet (avec un `main`) de code utilisant cette classe de manière à pouvoir lire interactivement depuis la console un numéro de téléphone et l'afficher dans le format donné s'il est valide. Si le numéro n'est pas valide, votre code doit afficher "Invalid number" sur la console.

Solution:

```
1 public class PhoneNumbers {
2     public static void main(String[] args) {
3         System.out.println("Please enter phone number : ");
4         String in = Input.readString();
5         String trans = Phone.testNumber(in);
6
7         if(trans != null)
8             System.out.println(trans);
9         else
10            System.out.println("Invalid number");
11    }
12 }
```




Question 5 – Reaching the Apex (7 points)

Sie implementieren eine Anwendung, die es erlaubt, die Personen für ein Videospiel zu erzeugen, indem die folgende Klassen verwendet wird:

Vous réalisez une application permettant de générer des personnages pour un jeu vidéo qui utilisent les classes suivantes :

```

1  class Character{
2      String firstName;
3      String lastName;
4  }
5
6  class Utils{
7  }

```

Sie möchten sich das wiederholte Lösen von Aufgaben vereinfachen. In der Tat liefert Ihnen das "storyline" Department ihrer Unternehmung eine Liste mit zu erzeugenden Personen, wobei der bereitgestellte String alle Personen mit Vornamen und Namen getrennt durch ein Semikolon (;) beinhaltet. Zum Beispiel:

Vous souhaitez vous simplifier la vie pour les tâches répétitives. En effet, le département *storyline* de votre entreprise vous fournit une liste de personnages à créer à partir d'un texte qui contient le prénom et le nom de chaque personnage, séparés par un point-virgule ;. Par exemple :

```

1  String text = "Anita_Bangalore;_Frank_BloodHound;Alex_Caustic;_Renee_Wraith;_Makoa_Gibraltar;Ajay
    _Lifeline;_Elliott_Mirage;_Marv_Pathfinder";

```

Diese Zeichenkette beinhaltet den Vornamen gefolgt vom Nachnamen für jede Person. Gelegentlich (aber nicht immer) steht vor dem Vornamen ein Leerzeichen.

Cette chaîne de caractères contient le prénom suivi du nom de famille de chaque personnage. Il y a parfois (mais pas toujours) un espace avant le prénom.

[7 Pt] (a) Schreiben Sie den benötigten Code (indem Sie die Klasse *Character* und die Klasse *Utils* vervollständigen), so dass der nachfolgende Code funktioniert und das erwartete Ergebnis anzeigt:

Écrivez le code nécessaire (en complétant la classe *Character* et la classe *Utils*) pour que le code suivant fonctionne et affiche le résultat attendu comme ci-dessous :

```

1  String text = "Anita Bangalore; Frank BloodHound;Alex Caustic; Renee Wraith; Makoa Gibraltar;
    Ajay Lifeline; Elliott Mirage; Marv Pathfinder";
2
3  Character[] apex = Utils.buildChars(text);
4
5  for(int i = 0; i < apex.length; i++) {
6      System.out.println(apex[i]);
7  }

```

Anzeigt | affiche

```

Anita, Bangalore
Frank, BloodHound
Alex, Caustic
Renee, Wraith
Makoa, Gibraltar
Ajay, Lifeline
Elliott, Mirage
Marv, Pathfinder

```

Hinweis: um den Code schreiben zu können, verwenden Sie die Methode *String.split()*, ggf. auch mehrmals. Diese Methode mit dem Prototyp *String[] split(char split)* teilt den *String* am durch den Parameter angegebenen Zeichen in auf und stellt jeden so gefundenen Teilstring als Eintrag in dem von der Methode zurückgegebenen Array zur Verfügung. Beispiel : "hello-world-this is nice".split('-') gibt {"hello", "world", "this is nice"} zurück.

Conseil : pour réaliser le code ci-dessus, nous vous demandons d'utiliser la méthode `String.split` (potentiellement plusieurs fois). Cette méthode a pour prototype `String[] split(char split)` et sépare le `String` sur lequel la méthode a été appelée en `n` chaînes, stockées dans un tableau, qui correspondent à la séparation par le caractère `split`. Par exemple `"hello-world-this is nice".split('-')` retournera le tableau `{"hello", "world", "this is nice"}`.

Solution:

```
1  class Utils {
2      static public Character[] buildChars(String s) {
3          // Splits all names which are separated with the ; symbol
4          String[] allNames = s.split(";");
5
6          int nChars = allNames.length;
7
8          // The end result which will contain the same amount
9          // of elements that are contained in the original String split with ;
10         Character[] chars = new Character[allNames.length];
11
12         // Parse each character first and last names one by one, removing the leading
13         // whitespace when required.
14         for (int i = 0; i < nChars; i++) {
15             String fullName = allNames[i];
16
17             // Remove start whitespace if present
18             if (fullName.charAt(0) == ' ') {
19                 fullName = fullName.substring(1);
20             }
21
22             String[] parsed = fullName.split(" ");
23
24             // Check that names are well formed
25             if (parsed.length != 2) {
26                 System.err.println("Invalid name, parsing stopped!");
27                 return null;
28             }
29
30             // Create Character from parsed values
31             chars[i] = new Character(parsed[0], parsed[1]);
32         }
33
34         return chars;
35     }
36 }
```

Question 6 – Washing it with a laundry (9 points)

- [8 Pt] (a) Erstellen Sie eine Klasse `Laundry`, welches eine Waschmaschine simuliert. Jeder Knopf der Maschine entspricht einer Methode. Neben der Wahl des Programms (welches über die Angabe der Temperatur erfolgt), können zusätzliche Programme gewählt werden, welche einzelne Parameter des gewählten Programms verändern. Das Programm wird für einen Waschgang einmalig ausgewählt, zusätzliche Programme (=Methoden) können für einen Waschgang mehrfach aufgerufen werden.
1. Attribute der Klasse sind die gewählte Temperatur, das Gewicht der Wäsche (in kg), die Dauer des Programms und das maximale Gewicht für das gewählte Programm. Die Attribute sollen vor Zugriff von ausserhalb geschützt werden.
 2. Erstellen Sie einen Default-Konstruktor für 40°, maximal 25 kg und 75 min.
 3. Erstellen Sie einen weiteren Konstruktor, welcher die Wahl des Programms erlaubt. Für 30° ist die Dauer 60 min, für 90° 80 min. Maximal dürfen 25 kg gewaschen werden. Bei allen anderen Eingaben für die Temperatur rufen Sie den Default-Konstruktor auf. Vergessen Sie nicht, dass, wenn Sie in einem Konstruktor einen anderen Konstruktor der Klasse aufrufen, dieser Aufruf die erste Zeile im Konstruktor sein muss!
 4. Implementieren Sie eine Methode `reduceTemp()`, welche die gewählte Temperatur um 10° reduziert. Die minimale Temperatur ist 30°.
 5. Erstellen Sie die Methode `delicate()`, welche für maximal 15 kg die gewählte Zeit um 10 min inkrementiert. Ausserdem muss die Temperatur um 10° reduziert werden. Verwenden Sie hierzu die entsprechende Methode.
 6. Erstellen Sie eine Methode `load()`, welche als Parameter die Menge der zu reinigenden Wäsche umfasst. Die Methode erhöht das Gewicht der aktuellen Ladung und gibt `true` zurück, wenn das neue Gesamtgewicht unter dem maximalen Gewicht liegt, `false` im anderen Fall.
 7. Ergänzen Sie eine Methode `unload()`, mit welcher die Menge Kleidung in der Maschine reduziert werden kann. Parameter der Methode ist das Gewicht, welches entnommen wird. Bei einem Wert ≤ 0 wird die Maschine komplett entleert (in der Regel nach erfolgtem Waschen). Im anderen Fall wird die angegebene Menge entnommen (in der Regel bei überfüllter Maschine). Die Methode gibt `true` zurück, wenn das nach der Entnahme noch verbleibende Gewicht geringer als das zulässige ist, im anderen Fall `false`.
 8. Eine Methode `start()`, mit welcher die Maschine mit dem Waschen beginnt. Geben Sie den Text `Start cleaning` aus sowie die aktuellen Werte für Temperatur, Dauer des Waschvorgangs und aktuelle Beladung.

Dans cet exercice vous devez écrire un programme simulant une machine à laver à l'aide de la classe `Laundry`. Chaque bouton de la machine correspond à une méthode. Le choix du programme se fait à l'aide de la température donnée lors de la création de la classe et qui va déterminer les paramètres de base du lavage qui sont la température, le poids maximal du linge et la durée. En plus de cela, il est possible de donner des modifications qui vont changer le programme de lavage. Ces modifications sont faites par des boutons (= méthodes) qui peuvent être pressés plusieurs fois.

1. Les attributs de la classe sont la température, le poids du linge (en kg), la durée et le poids maximal pour le programme. L'accès à ces attributs doit être protégé de l'extérieur de la classe.
2. Ajoutez un constructeur par défaut réglant une température de 40°, un poids max de 25 kg et une durée de 75 minutes.
3. Ajoutez un constructeur supplémentaire permettant le choix du programme de base à l'aide de la température. Pour 30°, la durée est de 60 minutes et pour 90° celle-ci passe à 80 minutes. Dans les deux cas, le poids maximal est de 25 kg. Pour toutes les autres températures, vous devez appeler le constructeur de base. N'oubliez pas que si vous faites appel à un autre constructeur dans un constructeur, cet appel doit être la première chose que vous faites dans le constructeur !
4. Implémentez une méthode `reduceTemp()` qui réduit la température par pas de 10°. La température minimale est de 30°.
5. Ajoutez une méthode `delicate()` qui réduit le poids max à 15 kg et qui ajoute 10 minutes de lavage. La température doit également être baissée de 10° à l'aide de la méthode correspondante.
6. Ajoutez une méthode `load` recevant en argument une quantité de linge qui est ajouté dans la machine. Cette méthode va retourner `false` si le poids du linge se trouvant dans la machine est trop élevé pour le programme actuel, `true` autrement.
7. Ajoutez une méthode `unload` qui va réduire la quantité de linge dans la machine. Le paramètre de cette méthode est le poids à enlever. Si cette valeur est ≤ 0 , la machine sera complètement vidée (normalement lorsque la machine a terminé le lavage). Si la valeur est positive, la quantité de linge correspondante sera enlevée du tambour. Cette méthode retourne `true` si la machine n'est pas surchargée, `false` autrement.

8. Ajoutez une méthode `start` qui simule le lavage. Cette méthode doit afficher `Start cleaning` ainsi que la quantité de linge chargée, la température choisie et le temps total de lavage.

```
1  public class Laundry {
2  private int temp, load, minutes, actualLoad;
3
4  public Laundry() {
5      temp = 40;
6      load = 25;
7      minutes = 75;
8  }
9
10 public Laundry(int temp) {
11     this();
12
13     if (temp == 30) {
14         minutes = 60;
15     }
16
17     if (temp == 90) {
18         minutes = 80;
19     }
20
21     this.temp = temp;
22 }
23
24 public void reduceTemp() {
25     if(temp >= 40)
26         temp -= 10;
27 }
28
29 public void delicate() {
30     load = 15;
31     minutes += 5;
32     reduceTemp();
33 }
34
35 public boolean load(int weight) {
36     actualload += weight;
37
38     if (actualload + weight < load) {
39         return true;
40     }
41     return false;
42 }
43
44 public boolean unload(int weight) {
45     if (weight <= 0) {
46         actualload = 0;
47         return true;
48     } else {
49         actualload -= weight;
50         if (actualload <= load)
51             return true;
52         else
53             return false;
54     }
55 }
56
57 public void start() {
58     System.out.println("Start cleaning...");
59     System.out.println("Temperatur: " + temp + " Load: " + actualload + " Minutes: " +
        minutes);
```

Listing continues on next page →

... continuing from previous page

60 }
61 }

△ Turn page →

- [1 Pt] (b) *Sie möchten ausserdem wissen, wie viele Waschmaschinen-Instanzen mit Hilfe ihres Codes erzeugt wurden. Erklären Sie im Detail, ohne ihren Code zu modifizieren, was Sie hinzufügen und modifizieren müssen.*
On souhaite connaître combien de machines ont été créées à l'aide du code que vous avez écrit. Expliquez en détails - sans modifier votre code – ce que vous devez ajouter et modifier ?

Solution: The solution is to add a static variable to store the number of created instances and increment it in the default constructor.

Fin|Ende
