Name/Nom : .....

Klasse/Classe: .....

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

TEST ANNUEL – JAHRESPRÜFUNG

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 sind 4 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	5	0	
String understanding	4	0	
True or false	6	0	
Small program with a loop	4	0	
Working with lists	4	0	
Recursive functions	8	0	
Electric streams ↯	10	0	
MMORPG	9	0	
Total:	50	0	

This exam has 8 questions, for a total of 50 points.

Rev 1.0

Question 1 – Short questions (5 points)

Diese Frage ist in verschiedene selbständige Aufgaben unterteilt. Der Wert von jeder Aufgabe wird am Rand angegeben. Cette question est séparée en plusieurs exercices indépendants. Le nombre de points pour chaque exercice est indiqué dans la marge.

[1 Pt] (a) *Was ist eine abstrakte Klasse? Erklären Sie dies zuerst theoretisch und geben sie anschliessend ein Beispiel.*
Qu'est-ce qu'une classe abstraite? Expliquez théoriquement et donnez un exemple.

Solution: An abstract class is a class which can not be instantiated. Its purpose is to provide some functionality (attributes, methods) and force the class that will inherit from it to implement some of its characteristics. For example, an abstract class `Shape` could force all its children to implement the method `getSurface`.

[1 Pt] (b) *Erklären Sie kurz was ein* layout manager *in Swing ist.*
Expliquez brièvement à quoi sert un *layout manager* en Swing.

Solution: A layout manager is a class that creates constraints for the location and size of Swing components. For instance, when a window is resized, a layout manager can force buttons to take all the available space on the window (by resizing or moving them).

[1 Pt] (c) *Was ist der Wert von* `a` *nach dem Ausführen des folgenden Codes:*
Que vaut `a` après l'exécution du code suivant ?

```
1 int a = 12;
2 float b = a % 5;
3 b += (0.4 * b);
4 a = (int)(a / b);
```

(c) **4**

[1 Pt] (d) *Was zeigt der folgende Code an ?*
Qu'affiche le code suivant ?

```
1 for (char a = 'z'; a > 'a'; a -= 7)
2   System.out.print("" + a + 'a');
```

Solution:

```
zasalaea
```

[1 Pt] (e) *Gegeben ist* | Soit

```
1 int method(int n){
2   if (n == 1 || n == 0)
3     return n;
4   else
5     return method(n-1) + method(n-2);
6 }
```

Ist die Komplexität der oben aufgeführten Methode linear? Erklären Sie Ihren Ansatz. Est-ce que la complexité de la méthode ci-dessus est linéaire ? Expliquez votre démarche.

Solution: The complexity of this method is worse than linear. If it were linear, the number of calls would rise linearly with the size of the problem (which is n in our case). In fact, the complexity is $\mathcal{O}(2^n)$ because :

$$1 + 2 + 4 + ... + (n-1) = 1((2^n) - 1)/(2-1)$$
$$= 2^n - 1$$

Question 2 – String understanding (4 points)

Gegeben ist die folgende Funktion: | Soit la fonction suivante:

```
1  public static String mystery(String a){
2    String b = a.trim();
3    int p = b.indexOf(' ');
4
5    do{
6      String sa = b.substring(0, p);
7      System.out.println(sa);
8      b = b.substring(p+1, b.length());
9      p = b.indexOf(' ');
10
11     if(p == -1){
12       System.out.println(b.toLowerCase());
13       break;
14     }
15
16   } while(true);
17
18   return a.trim().toUpperCase();
19 }
```

[1 Pt] (a) *Gesucht ist der Wert für* `b` *(Zeile 2), wenn* `a` *"␣the␣horse␣runs␣fast␣␣␣" entspricht.*
Que vaut `b` à la ligne 2 du code ci-dessus lorsque `a` vaut `"␣the␣horse␣runs␣fast␣␣␣"`.

Solution: `the horse runs fast"`

[1 Pt] (b) *Was gibt diese Methode für* `""Hello World` *?* ***zurück****?*
Que **retourne** cette méthode pour `"Hello world"` ?

Solution: `"HELLO WORLD"`

[2 Pt] (c) *Was zeigt diese Methode an, wenn sie* `"The Mammoth is GrEY"` *als Argument erhält?*
Qu'affiche cette méthode si on lui donne `"The Mammoth is GrEY"` comme argument ?

Solution:
```
The
Mammoth
is
grey
```

Question 3 – True or false (6 points)

	True	False
Es ist nicht möglich, eine endlosschleife mit einem for zu schreiben. Il n'est pas possible de faire une boucle infinie avec un *for*.	☐	⊗
Eine Klasse kann zwei Interfaces implementieren. Une classe peut implémenter 2 interfaces.	⊗	☐
Eine abstrakte Klasse kann keine Implementation von einer Methode enthalten. Une classe abstraite ne peut pas contenir l'implémentation d'une méthode.	☐	⊗
Mehrere Klassen können von der gleichen Klasse erben. Plusieurs classes peuvent hériter de la même classe.	⊗	☐
Ein Vektor ist viel effizienter als eine verkettete Liste um Elemente am Anfang einzufügen. Un vecteur est bien plus efficace qu'une liste chaînée pour y insérer des éléments au début.	☐	⊗
Ein Algorithmus der Komplexität $\mathcal{O}(0)$ ist viel effizienter als ein Algorithmus $\mathcal{O}(n^2)$ für eine grosse Anzahl von Elementen. Une algorithme de complexité $\mathcal{O}(0)$ est plus efficace qu'un algorithme en $\mathcal{O}(n^2)$ pour un grand nombre d'éléments.	⊗	☐
In einer try-catch Struktur wird der catch Teil ausgeführt, wenn es ein Problem mit der Kompilation im Try-Teil gibt. Dans une structure *try-catch*, la partie *catch* est exécutée quand il y a eu un problème de compilation dans la partie *try*.	☐	⊗
Eine rekursive Methode ist immer wirksamer als eine iterative Methode (Schleifen). Une méthode récursive est toujours plus efficace qu'une méthode itérative (boucle).	☐	⊗
Die Methode `Math.cos` ist statisch. La méthode `Math.cos` est une méthode statique.	⊗	☐
Eine abstrakte Methode kann nur in einer abstrakten Klasse existieren. Une méthode abstraite ne peut exister que dans une classe abstraite.	⊗	☐
Um von einer abstrakten Klasse zu erben, wird das Schlüsselwort implements benutzt. Pour hériter d'une classe abstraite, on utilise le mot clef *implements*.	☐	⊗
Wenn die Klasse A von der Klasse B erbt und die Klasse B von der Klasse C erbt, so erbt die Klasse A nicht die Attribute welche in der Klasse C definiert sind. Si la classe A hérite de la classe B et la classe B hérite de la classe C, A n'hérite pas des attributs définis dans C.	☐	⊗

Question 4 – Small program with a loop (4 points)

Ein Unternehmen bezahlt seine Verkäufer auf Kommissionsbasis. Jeder Verkäufer erhält einen Grundlohn von 400.-/Woche sowie eine Kommission von 9% seiner Verkäufe während der Woche. Beispiel: Ein Verkäufer, der Waren für insgesamt 5'000.- verkauft hat, erhält einen Lohn von 850.-.

Schreiben Sie ein vollständiges Programm mit einer `while` oder `do/while`-Schleife, mit dem dieser Lohn berechnet werden kann. Die Ausführung Ihres Programms muss dem nachstehenden Beispiel 100% ähnlich sein (die Eingaben des Benutzers in der Konsole sind fett geschrieben):

```
Enter sales in dollars (-1 to end): 5000
Salary is: $850.0
Enter sales in dollars (-1 to end): 6000
Salary is: $940.0
Enter sales in dollars (-1 to end): -1
```

Une entreprise paie ses vendeurs à la commission. Chaque vendeur reçoit un salaire fixe de 400.- par semaine ainsi qu'une commission correspondant à 9% des ventes de la semaine. Par exemple, un vendeur ayant réalisé pour 5000.- de ventes recevra un salaire total de 850.-.

Écrivez un programme **complet** utilisant une boucle `while` ou `do while` permettant de calculer ce salaire. L'exécution de votre programme doit être **exactement** pareil que l'exemple ci-dessous (ce que l'utilisateur tape dans la console est en gras):

```
Enter sales in dollars (-1 to end): 5000
Salary is: $850.0
Enter sales in dollars (-1 to end): 6000
Salary is: $940.0
Enter sales in dollars (-1 to end): -1
```

Solution:

```java
import java.util.Scanner;

public class Salary {

  public static void main(String[] args) {

    Scanner input = new Scanner(System.in);
    int val = 0;

    while(true){
      System.out.print("Enter sales in dollars (-1 to end): ");
      val = input.nextInt();
      if(val == -1) break;

      double com = val * 9.0 / 100 + 400;

      System.out.print("Salary is: ");
      System.out.println(com);
    }

    input.close();
  }
}
```

Question 5 – Working with lists (4 points)

Die nachstehenden Klassen sind die Implementierung einer verketteten Liste.
Les classes suivantes sont l'implémentation d'une liste chaînée.

```
1  public class LinkedList {
2    public Node head;
3
4    public void addToStart(String item) {
5      head = new Node(item, head);
6    }
7  }
8
9  class Node {
10   String item;
11   Node next;
12
13   public Node(String item, Node nextElement) {
14     this.item = item;
15     this.next = nextElement;
16   }
17 }
```

Implementieren Sie eine Methode namens `reverseAndRemoveZ` (in Klasse `LinkedList`, welche keine Parameter besitzt und welche eine neue verkettete Liste zurückgibt. Diese neue Liste enthält alle Elemente in umgekehrter Reihenfolge ausser den Elementen welche den Buchstaben 'z' (Gross und Kleinbuchstabe) enthalten. Zum Beispiel: wenn die Funktion auf folgender Liste aufgerufen wird:

Ajoutez à la classe `LinkedList` une méthode nommée `reverseAndRemoveZ` qui ne prend pas de paramètre et qui retourne une nouvelle liste chaînée, qui contient les éléments de la liste dans l'ordre inverse, sauf tous les éléments qui comportent la lettre 'z' (majuscule ou minuscule). Appelée sur cette liste :

```
zut -> Guys -> Zorro -> Hi -> null
```

Gibt die Funktion folgende Liste zurück. | elle retourne la liste suivante:

```
Hi -> Guys -> null
```

Solution:

```
1  public LinkedList reverseAndRemoveZ() {
2    LinkedList result = new LinkedList();
3    for (Node a = head; a != null; a = a.next) {
4      if (a.item.indexOf("z") != -1 || a.item.indexOf("Z") != -1)
5        continue;
6      result.addToStart(a.item);
7    }
8    return result;
9  }
```

Question 6 – Recursive functions (8 points)

Um die Lesbarketi von grossen Zahlen zu verbessern, ist es üblich, Trenn Apostrophe bei jedem Vielfachen von 1000 anzufügen. Somit werden die Zahlen 500, 1234 *und* 99999999 *wie folgt geschrieben* 500, 1'234, *und* 99'999'999.

Lorsque l'on écrit des grands nombres, il est d'usage de rajouter des apostrophes séparatrices à chaque multiple de mille afin d'améliorer la lisibilité. Ainsi, les chiffres 500, 1234 et 99999999 peuvent s'écrire comme suit : 500, 1'234 et 99'999'999.

[3 Pt] (a) *Erklären Sie **ausführlich** das Prinzip (Berechnungen, verwendete Methoden, etc.) um dies rekursiv zu realisieren. Im Moment müssen Sie diese Methode nicht implementieren.*

Expliquez de manière **détaillée** le principe (calculs, méthodes utilisées…) à utiliser pour réaliser cela de manière récursive. Vous ne devez pas implémenter la méthode pour l'instant.

Solution: The idea is to call the function recursively until we have a number under 1000. When this is done, we return the String corresponding to that number (because it will not contain any apostrophe). The recursive call is done with n / 1000 and we add to the result (a String) the three last characters corresponding to the current value. For instance, for 12384 we have :

String number = + 12384; addSep(12384 / 1000) + "'"+ number.substring(2);

[5 Pt] (b) *Implementieren Sie nun eine Methode namens `addSep`. Diese Methode erhält eine Ganzzahl als Argument und liefert als Rückgabewert einen `String` welcher die Zahl mit Apostrophen enthält. Ihre Methode **muss** rekursiv sein.* Bemerkung: *Wenn Sie nicht wissen, wie diese Methode rekursiv implementiert werden kann, schreiben Sie diese Methode nicht rekursiv. ⚠Zur Erinnerung: der Ausdruck $+3$ entspricht dem `String "3"`. Ihr dürft die Methoden `substring` und `length` der Klasse `String` benutzen.*

Implémentez la méthode que vous nommerez `addSep`. Celle-ci doit prendre un nombre entier en argument et retourne la chaîne de caractères avec apostrophes correspondantes. Votre méthode **doit** être récursive. *Note :* si vous n'avez pas trouvé comment faire la version récursive, implémentez la version non-récursive. ⚠Pour rappel, l'expression $+3$ retourne le `String "3"`. Vous pouvez utiliser les méthodes `substring` et `length` de la classe `String`.

Solution:

```
1  public static String addSep(int n) {
2       // Base case
3       if (n < 1000)
4            return "" + n;
5
6       // Recursive case
7       String number = "" + n;
8       return addSep(n/1000) + "'" + number.substring(number.length()-3);
9  }
```

Question 7 – Electric streams ⚡ (10 points)

Ein Stromzähler ist mit einem Computer verbunden. Er misst in regelmässigen Abständen den Stromverbrauch von verschiedenen Geräten sowie auch verschiedene Informationen wie der Strom und die Netzfrenquenz. Um ein Computerprogramm zu schreiben welcher mit diesen Messungen arbeitet, schlagen wir folgende Klasse vor:

Un compteur électrique est connecté à un ordinateur. Il mesure à intervalle régulier la consommation de divers appareils ainsi que différentes informations comme le courant ainsi que la fréquence de la ligne. Pour réaliser un programme informatique utilisant de telles mesures, nous proposons la classe suivante:

```
1  public class Measure {
2    double power;
3    int frequency;
4
5    public Measure(double p, int freq){
6      power = p;
7      frequency = freq;
8    }
9  }
```

[2 Pt] (a) *Wie möchten folgenden Code schreiben können:* | Nous aimerions pouvoir écrire le code suivant:

```
1  Measure m = new Measure(3);
2  System.out.println(m);
```

Fügen Sie nun den minimalen Code hinzu, damit Ihr Programm korrekt funktioniert und folgendes Resultat auf der Konsole anzeigt (unter der Annahme dass die Standard-Frequenz 50 ist):

Rajouter le moins de code possible pour que ce code soit valide et produise le résultat suivant sur la console (en partant du principe que par défaut la fréquence vaut 50) :

```
Measure : 3.0 [W] @ 50 [Hz]
```

Solution:

```
1  public Measure(double w){
2      this(w, 50);
3  }
4
5  public String toString(){
6    return "Measure : " + watts + " [W] " + "@ " + frequency + " [Hz]";
7  }
```

Die Messungen sind in einer Datei nach unten aufgeführtem Format abgespeichert: Les valeurs sont stockées dans un fichier selon le format défini ci-dessous :

```
1  #,3,500
2  #,10,510
3  #,?,40,?,?,500
4  #,?,?,?,12,500
```

Die erste Zeile entspricht einer Messung von 3 Watts bei 50 Hz, die zweite Linie entspricht einer Messung von 10 Watts bei 51 Hz (beachte dass die Werte der Frequenz in der Datei mit 10 mutlipliziert werden). Jede Datei enthält eine variable Anzahl von Messungen und bei jedem Messintervall wird eine neue Linie in der Datei hinzugefügt. Jede Linie muss mit dem Charakter `#` beginnen und die verschiedenen Werte werden mit einem Komma getrennt.

Wie Sie in Zeile 3 und 4 sehen können, ist es möglich, dass die Datei das Zeichen `?` enthalten kann. Dieses Zeichen entspricht nicht verfügbaren Leistungswerten weleche ignoriert wereden sollen. Auch in Gegenwart von unbekannten Messungen kommt die Leistung immer von der Frequenz und in allen Fällen kommt die Frequenz immer zuletzt.

Zur Information, die Methode `split` von `String` teilt eine Zeichenkette an der Stelle mit einem gegebenen Charackter. Das Resultat ein Array von `String`. Beispiel: `"a-lo-ha".split("-") -> {"a", "lo", "ha"}`

La première ligne correspond à une mesure de 3 watts à 50 Hz, la seconde ligne à une mesure de 10 watts à 51 Hz (notez que les valeurs en Hz sont multipliées par 10 dans le fichier). Chaque fichier contient un nombre variable de mesures et à chaque intervalle de mesure, une nouvelle ligne est ajoutée dans ce fichier. Chaque ligne doit commencer par le caractère `#` et les différentes valeurs sont séparées par des virgules.

Comme vous le voyez à la ligne 3 et 4, il se peut que le fichier contienne des caractères `?` qui correspondent à des valeurs de puissance indisponibles qui doivent être ignorées. Même en présence de mesures inconnues, la puissance vient toujours avant la fréquence et, dans tous les cas, la fréquence apparaît toujours en dernier.

Pour rappel, la méthode `split` de la classe `String` permet de découper un `String` à l'aide d'un caractère donné. Le type de résultat est un tableau de `String`. Exemple : `"a-lo-ha".split("-") -> {"a", "lo", "ha"}`

[2 Pt] (a) *Es ist möglich, das Linien mit Messungen nicht gültig sind. Die zwei möglichen Ursachen sind, dass die Zeile nicht mit `#` beginnen order dass die Zeile nur unbekannte Werte, in Folge eines Problems mit dem Sendor, enthält. Beachte, das angenommen wird, dass wenn ein numerischer Wert in der Linie enthalten ist, muss der andere Wert auch da sein. Implemetieren die Funktion `isValid`, die als Parameter eine Linie der Datei mit den Messungen erhält und `true` zurückgibt, wenn die Linie gültig ist.*

Il se peut que des lignes de mesures ne soient pas valides. Les deux problèmes possibles sont que la ligne ne commence pas par `#` ou que la ligne ne contient que des valeurs inconnues en raison d'un problème de capteurs. Notez que l'on fait l'hypothèse que si une valeur numérique est présente dans la ligne, la seconde le sera également. Implémentez la fonction `isValid` qui prend comme paramètre une ligne du fichier de mesure et retourne `true` lorsque celle-ci est valide.

Solution:

```
1  public static boolean isValidAlternate(String m){
2    if(m.charAt(0) != '#' || m.charAt(m.length()-1) == '?')
3      return false;
4
5    return true;
6  }
```

other solution

```
1  public static boolean isValid(String m){
2    if(m.length() == 0 || m.charAt(0) != '#')
3      return false;
4
5    String[] tbl = m.split(",");
6
7    // We need at least 3 values split (#, val1, val2)
8    if (tbl.length < 3)
9      return false;
10
11   // Check if only ? are present
12   for(int i = 1; i < tbl.length; i++){
13     if(tbl[i].equals("?") == false)
14       return true;
15   }
16
17   return false;
18 }
```

[3 Pt] (b) *Implementieren Sie nun eine Funktion `decodeMeasure`, welche den Inhalt einer kompletten Messung, durch dekodieren der Linie zurückgibt. Die Linie wird als gültig angenommen. Der Rückgabewert ist vom Typ `Measure`.*

Implémentez maintenant une fonction `decodeMeasure` qui doit retourner le contenu d'une mesure complète en décodant une ligne que l'on supposera valide. La valeur retournée est de type `Measure`.

Solution:

```java
public static Measure decodeMeasure(String input) {
  String[] t = input.split(",");

  int power = 0;
  int freq = 0;

  for(int i = 1; i < t.length-1; i++){
    if(t[i].contains("?"))
      continue;

    power = Integer.parseInt(t[i]);
  }

  freq = Integer.parseInt(t[t.length-1]) / 10;
  return new Measure(power, freq);
}
```

[3 Pt] (c) *Implementieren Sie zum Abschluss eine Methode namens* `decodeFile` *welche als Parameter den Namen einer Datei mit Messungen erhält. Diese Datei soll geöffnet werden und die messungen jeder gültigen Linie soll in einem Vektor abgespeichert werden. Dieser Vektor wird am Schluss der Funktion zurückgegeben.*

Implémentez finalement une méthode nommée `decodeFile` qui prend en argument le nom d'un fichier de mesures, l'ouvre, et insère la mesure de chaque ligne valide dans un vecteur qui est retourné à la fin.

Solution:

```java
public static Vector<Measure> decodeFile(String fn) {
  Vector<Measure> measures = new Vector<Measure>();

  try {
    BufferedReader input = new BufferedReader(new FileReader(fn));
    String line = input.readLine();
    while (line != null) {
      if(isValid(line)){
        Measure m = decodeMeasure(line);
        measures.add(m);
      }

      line = input.readLine();
    }
    input.close();
  } catch (IOException e) {
    e.printStackTrace();
  }

  return measures;
}
```

Question 8 – MMORPG (9 points)

Eine Informatik Firma hat entschieden, ein neues MMORPG (Multi-Players Online Roleplaying Game) zu realisieren. Eine Objekt-Orientierte Struktur spezifiziert, welche die verschiedenen Einheiten des Spiels zu erstellen.

Une firme informatique a décidé de créer un nouveau MMORPG (Multi-Players Online Roleplaying Game) et une structure a été a été spécifiée pour réaliser les différentes unités du jeu.

[3 Pt] (a) *Schreiben Sie die Implementation der Klasse Unit welche die Spitze der Hierarchie darstellt. Diese Klasse besitzt vier Attribute, der Namen der Einheit (*`name`*), die Erfahrung (*`XP`*), die Punkte der Gesundheit (*`HP`*) und die Punkte des Schutzes (*`armor`*). Die letzten drei Attribute sind als* `int` *abgespeichert und müssend als default Wert: 0 für die Erfahrung, 10 für die Gesundheit und 5 für den Schutz besitzen. Die Klasse, welche nicht instanziert werden kann, besitzt ausserdem einen Konstruktor, welcher den Namen der Einheit spezifiziert.*

Écrivez l'implémentation de la classe `Unit` qui représente le sommet de la hiérarchie. Cette classe comporte quatre attributs, le nom de l'unité (`name`), l'expérience (`XP`), les points de vie (`HP`) et les points d'armure (`armor`). Les trois derniers attributs, stockés sous forme de `int`, doivent avoir des valeurs par défaut : 0 pour l'expérience, 10 pour les points de vie et 5 pour l'armure. Cette classe, qui ne doit pas être instanciable, doit aussi comporter un constructeur qui permet de spécifier le nom de l'unité.

Solution:

```java
public abstract class Unit {
  public String name;
  public int XP = 0;
  public int HP = 10;
  public int armor = 5;

  public Unit(String name) {
    this.name = name;
  }
}
```

[3 Pt] (b) *Schreiben Sie nun eine Klasse* `MovingUnit` *welche von der Klasse* `Unit` *erbt. Diese Klasse, welche auch nicht instanziert werden kann, besitzt ein zusätzliches Integer Attribut namens* `MP`*, welches die Zahl der Bewegungspunkte darstellt. Ausserdem muss diese Klasse sicherstellen, dass alle Klassen, welche von dieser Klasse erben, eine Methode move implementieren. Die Methode* `move` *erhält zwei Ganzzahlen als Parameter.*

Écrivez maintenant une classe `MovingUnit` qui hérite de la classe `Unit`. Cette classe, qui elle aussi ne doit pas être instanciable, possède un attribut entier supplémentaire nommé `MP` pour représenter le nombre de points de mouvement. De plus, cette classe doit faire en sorte que toutes les classes qui hériteront de celle-ci seront obligées d'avoir une procédure `move` qui prend deux entiers en paramètres.

Solution:

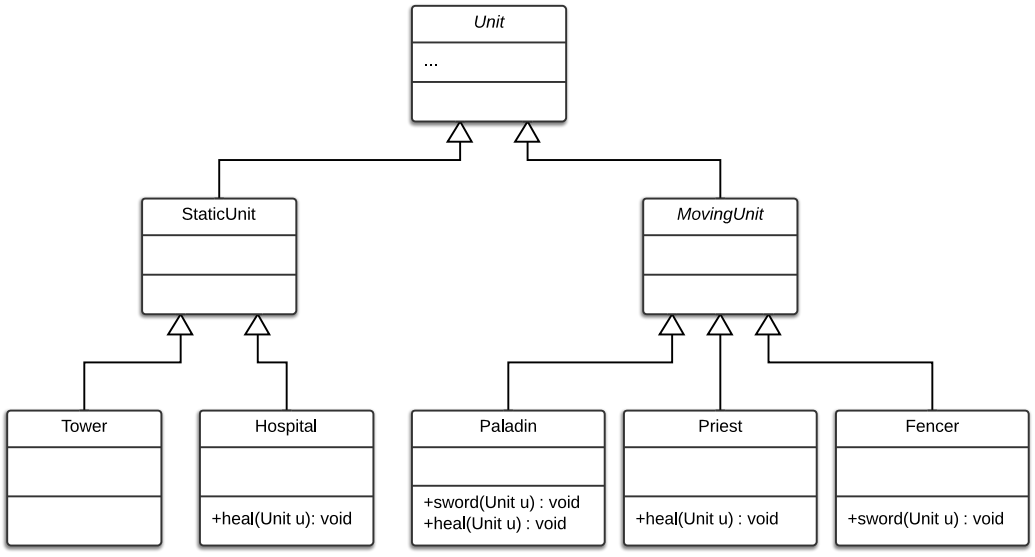
```java
public abstract class MovingUnit extends Unit {
  int MP = 1;

  public MovingUnit(String name) {
    super(name);
  }

  abstract public void move(int x, int y);
}
```

[1 Pt] (c) *Das unten aufgeführte UML Diagramm repräsentiert einen Teil der Hierarchie des Spiels. Dies ermöglicht es Ihnen, einen besseren Überblick über die Vererbung der Klassen zu erhalten. Es wird darauf hingewiesen, dass die Beschreibung der Klassen* ***unvollständig*** *ist.*

Man möchte nun Spezifikationen für zwei Kategorien von Einheiten hinzufügen. Die erste sind die Pfleger (`Healer`*) welche eine Methode namens* `heal` *besitzen müssen. Die Methode* `heal` *erhält als Parameter eine Einheit. Beispiele dieser Klasse im oben aufgeführten UML-Diagramm sind* `Hospital`*,* `Priest` *und* `Paladin`*. Die zweiten welche spezifiziert werden sind die Krieger (*`Warrior`*) welche eine Methode namens* `sword` *besitzen müssen. Die Methode* `sword` *erhält auch eine Einheit als Parameter (zum Beispiel* `Paladin` *oder* `Fencer`*). So gibt es einige Klassen welche* `Healer` *oder* `Warrior` *oder beides sind. Diese Klassen können sowohl in der Kategorie* `MovingUnit` *oder* `StaticUnit` *sein. Erklären Sie in einem Satz welche Lösung benutzt werden könnte, um diese Spezifikation zu realisieren.*



La structure UML ci-dessus représente une partie de la hiérarchie des unités du jeu. Cela permet de voir un peu mieux la structure d'héritage des classes désirée. Il est à noter que la description des classes est **incomplète**.

On voudrait maintenant rajouter des spécifications pour deux catégories d'unités, celles qui soigent (`Healer`) et qui doivent posséder une méthode nommée `heal` qui reçoit une unité en paramètre. Des exemples de classes dans le diagramme ci-dessus sont `Hospital`, `Priest` ou `Paladin`. On voudrait aussi spécifier celles qui se sont des guerriers (`Warrior`) et qui doivent posséder une méthode `sword` qui reçoit aussi une unité en paramètre (p. ex. `Paladin` ou `Fencer`). Il y a donc que certaines classes qui sont `Healer` ou `Warrior`), ou les deux. Ces classes peuvent être autant dans la catégorie des `MovingUnit` ou des `StaticUnit`. Expliquez en une phrase quelle solution pourrait être utilisée pour créer ces spécifications.

Solution: To create the transversal specifications, interfaces could be used.

[2 Pt] (d) *Schreiben Sie nun den Code um diese Spezifikation zu realisieren.*
Écrivez maintenant le code qui permet de créer ces spécifications.

Solution:

```
1  public interface Warrior {
2    public void sword(Unit unit);
3  }
4
5  public interface Healer {
6    public void heal(Unit unit);
7  }
```