

Klausur: Programmiermethoden und Techniken

WS12/13

Erlaubte Hilfsmittel: keine

Lösung ist auf den Klausurbögen anzufertigen. (eventuell Rückseiten nehmen)

Bitte legen Sie den Studentenausweis auf den Tisch.

Bearbeitungszeit: 90 Minuten

Unterschrift

Benotung

Aufgabe:	1	2	3	4	5		Gesamt	Note
Punkte:	20	20	20	20	20		100	
erreicht:								

Aufgabe 1 (20 Punkte)
(XML)

- a) Schreiben Sie mit dem DOM API folgende Methode:

```
public static int maxWidth(org.w3c.dom.Node node);
```

Sie soll die maximale Anzahl, die ein Knoten des übergebenen XML-Baums als Kindknoten hat, zurück geben.

- b) Schreiben Sie mit dem DOM API folgende Methode:

```
public static void getNames(Node node, Set<Node> result)
```

Alle Tagnamen von Elementen im übergebenen XML-Baum sollen in die Menge `result` eingefügt werden.

- c) Schreiben Sie einen XPath-Ausdruck, der alle Elementknoten mit dem Tagnamen `kapitel` aus einem Dokument selektiert, die das Attribut `@titel` mit dem Wert "Einführung" haben.

Name:

Matrikelnummer:

Name:

Matrikelnummer:

Aufgabe 2 (20 Punkte)
(Iteratoren in Java)

In dieser Aufgabe sollen Sie zwei Klassen schreiben, die die Schnittstelle `Iterator` implementieren. Die Methode `remove` soll dabei vernachlässigt werden.

- a) Schreiben Sie eine Klasse `StringCharacterIterator`. Die Klasse soll die Schnittstelle `Iterator<Character>` implementieren. Es soll einen Konstruktor `StringCharacterIterator(String str)` geben. Der Iterator soll beim Aufruf von `next` alle Zeichen des Strings `str` nacheinander zurück geben.

Name:

Matrikelnummer:

b) Gegeben sei die folgende Schnittstelle:

```
Function.java  
1 interface Function<A>{  
2     A apply(A e);  
3 }
```

Schreiben Sie eine Klasse `MapIterator<A>`, die `Iterator<A>` implementiert. Es soll einen Konstruktor der folgenden Form geben:

```
MapIterator(Iterator<A> orgIterator, Function<A> f)
```

Der so erzeugte Iterator soll beim Aufruf von `next` jeweils nacheinander die Elemente des Iterators `orgIterator` zurück geben, allerdings nicht das Element von `orgIterator` direkt, sondern das Ergebnis von der Methode `f.apply(next())`.

Name:

Matrikelnummer:

Aufgabe 3 (20 Punkte)
(Strings in C)

Gegeben sei die folgende Struktur, die String-Objekte realisiert:

```
String.c  
1 typedef struct{  
2     unsigned int length;  
3     char* data;  
4 } String;
```

Schreiben Sie folgende Funktionen in C:

a) `int indexOfFirstVowel(String this)`

Es soll der Index im String zurück gegeben werden, an dem sich der erste Vokal befindet. Ein Vokal ist eines der folgenden Zeichen: `aAoOuUeEiI`. Wenn kein Vokal im String ist, soll `-1` zurück gegeben werden.

Name:

Matrikelnummer:

b) String rotate(String this)

Es soll ein neuer String erzeugt werden, der aus dem Parameter `this` gebildet wird, in dem der letzte Buchstabe des Ausgangsstrings der erste im Ergebnis wird und jeder weitere Buchstabe einen Index nach hinten rutscht.

Beispiel: `rotate({"hallo"},5) = {"ohall"},5}`

Name:

Matrikelnummer:

Aufgabe 4 Entwerfen Sie in dieser Aufgabe eine kleine Bibliothek zur Darstellung von Punkten im zweidimensionalen Raum.

Ein Punkt habe dabei eine x- und eine y-Koordinate, die durch eine Fließkommazahl dargestellt werden soll..

a) Schreiben Sie eine geeignete Struktur `struct Punkt` für die Bibliothek.

b) Schreiben Sie eine geeignete Konstruktorfunktion, mit der ein Zeiger auf ein Punkt-datenobjekt erzeugt und dieses Objekt initialisiert wird.

Name:

Matrikelnummer:

c) Schreiben Sie eine Funktion, die ein Punktobjekt sowohl in x - als auch in y -Richtung um einen Faktor verschiebt.

d) Schreiben Sie eine Funktion, die zwei Punktobjekte als Parameter erhält und testet, ob der zweite Punkt auf der x -Achse weiter rechts als der erste Punkt liegt.

Name:

Matrikelnummer:

Aufgabe 5 (20 Punkte)

Erklären Sie in kurzen Worten.

a) Was ist der Unterschied zwischen einem InputStream und einem Reader in Java.

b) Was passiert im Speicher, wenn in Java der Operator + auf zwei Strings ausgeführt wird.

Name:

Matrikelnummer:

c) Was findet sich im Java Paket `org.w3c.dom`. Erklären Sie die Gründe.

d) Was genau ist ein Array in C. Wie unterscheidet sich das zu Arrays in Java?

Name:

Matrikelnummer:

e) Wenn man in C eine Liste programmieren möchte, die möglichst generisch für unterschiedliche Daten funktioniert, wie könnte man vorgehen?

f) Was ist ein `typedef` und welchen Nutzen hat dieses Konstrukt in C?