

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 11.6.-15.6.2018 besprochen.

## Übungsblatt 8

### Aufgabe 1

#### Entwurfsmuster – Strategie

Die Klasse `MysteryStructure` realisiert eine Datenstruktur, von der nur fünf Methoden bekannt sind. Die Klasse soll um eine öffentliche Methode zum Sortieren ergänzt werden, die den Inhalt der `MysteryStructure` mit der Hilfe eines zum Interface `Sorter` kompatiblen Objekt sortiert. Geben Sie die Implementierung der folgenden Methode an:

```
public void sort( Sorter<E> s )
```

```
public class MysteryStructure<E extends Comparable<E>> implements Iterable<E>
{
    public MysteryStructure() {...}
    public Iterator<E> iterator()           // gibt einen passenden Iterator zurück
    public int size() {...}                 // gibt die Anzahl der abgelegten E-Objekte zurück
    public E getAt( int i ) {...}           // gibt das E-Objekt an der Position i zurück,
                                           // Zugriff erfolgt in annähernd konstanter Zeit
    public void clear() {...}               // löscht alle abgelegten E-Objekte
    public void add( E content ) {...}      // nimmt das Argument in die Datenstruktur auf
    ...
}
```

```
public Interface Iterator<V>
{
    V next();
    boolean hasNext();
}
```

```
public interface Iterable<T>
{
    Iterator<T> iterator();
}
```

```
public interface Sorter<E extends Comparable<E>>
{
    public Iterable<E> sort( Iterable<E> c );
}
```

### Aufgabe 2

#### Entwurfsmuster – Dekorierer

Der Online-Händler **Orinoco** hat ein Versandkonzept entwickelt, das beim Anlegen und Abrechnen durch ein Dekorierermuster unterstützt werden soll. Ein Besteller kann zwischen verschiedenen Transportleistungen auswählen, die in der folgenden Tabelle stehen.

Bezeichnung	Kosten
Paket Größe M (Lieferdauer 14 Tage)	€ 2,00
Paket Größe L (Lieferdauer 14 Tage)	€ 3,00
Lieferdauer verkürzen um je 1 Tag	€ 0,50
Lieferzeitraum festlegen auf +- 2 Stunden	€ 3,00
Lieferzeitraum präzisieren um je +- 30 Minuten	€ 0,50
persönliche Übergabe durch den Zusteller	€ 2,00
Versicherung für einen Versandwert von je 50 €	€ 1,00

*Beispiel:*

Bei Bestellung eines 430 € teuren SmartPhones (passt in Paket der Größe M), das voll versichert morgen zwischen 11 und 12 Uhr geliefert und persönlich übergeben werden soll, fallen als Kosten an:

$$2,00 + 13 \cdot 0,50 + 3,00 + 3 \cdot 0,50 + 2,00 + 9 \cdot 1,00 = 24,00 \text{ €}$$

- Geben Sie ein Klassendiagramm für ein geeignetes Dekorierer-Muster an.
- Skizzieren Sie die Methoden zum Berechnen der Kosten.
- Skizzieren Sie die Methoden zum Berechnen des Lieferzeitraums in Tagen.
- Skizzieren Sie Methoden zum Aufbau des Stapels von Dekorierer-Objekten. Die Methoden sollen sicherstellen, dass
  - (1) die Lieferdauer nicht kürzer als 1 Tag werden darf,
  - (2) der Lieferzeitraum nicht präziser als ein Zeitpunkt werden darf und
  - (3) das Paket nur einmal übergeben werden darf.

## Aufgabe 3

### Entwurfsmuster – Kompositum

Zeichnen Sie ein Klassendiagramm für ein *Kompositum*-Muster, das die im Folgenden beschriebenen Beziehungen zwischen verschiedenen Strukturobjekten geeignet umsetzt:

Ein *Verzeichnis* kann *Dateien*, *Archive* oder andere *Verzeichnisse* enthalten.

Ein *Archiv* kann *Dateien*, *Verzeichnisse* oder andere *Archive* enthalten.

Jede *Datei* besitzt ein Attribut *groesse*, das die Größe der Datei in Bytes als **int**-Wert angibt.

- Für das oben modellierte Kompositum-Muster soll eine Methode `int groessteDatei()` angelegt werden. Die Methode `groessteDatei` soll für eine mit dem Kompositum-Muster modellierte Struktur die Größe der größten an ihr beteiligten Datei liefern. Skizzieren Sie für jede der von Ihnen vorgesehenen Klassen *entweder* als informale Beschreibung *oder* als Java-Programmcode, wie die Methode `groessteDatei` implementiert werden müsste.
- Für das oben modellierte Kompositum-Muster soll eine Methode `int ueberLimit(int lim)` angelegt werden. Die Methode `ueberLimit` soll für eine mit dem Kompositum-Muster modellierte Struktur die Zahl der an ihr beteiligten Dateien liefern, die größer als `lim` sind. Skizzieren Sie für jede der von Ihnen vorgesehenen Klassen *entweder* als informale Beschreibung *oder* als Java-Programmcode, wie die Methode `ueberLimit` implementiert werden müsste.
- Für das oben modellierte Kompositum-Muster soll eine Methode `boolean expandiert()` angelegt werden. Die Methode `expandiert` soll für eine mit dem Kompositum-Muster modellierte Struktur den Wert `true` liefern, falls diese kein Archiv enthält. Skizzieren Sie für jede der von Ihnen vorgesehenen Klassen *entweder* als informale Beschreibung *oder* als Java-Programmcode, wie die Methode `expandiert` implementiert werden müsste.