

Übung zu Algorithmen und Programmieren III, WS 2001/2

Übung 8

Ausgabe: 4.12.01

Abgabe: 13.12.01 bis 14.00 Uhr

Aufgabe 1 (6 P)

a) Entwerfen und implementieren Sie bitte eine Hierarchie für geometrische Objekte (ein Interface/abstrakte Klasse für 2-dimensionale Objekte und Klassen für Dreieck, Kreis und Rechteck). Die minimale Implementierung (Entwurf) soll die Fläche berechnen können und das geometrische Objekt bewegen können.

b) Schreiben Sie eine Klasse, die über eine Menge der 2-dimensionalen Objekte iteriert und die Summe der Flächeninhalte bestimmt.

Aufgabe 2 (4 P)

Gegeben sei eine Klasse Counter mit folgenden Methoden:

```
public class Counter {
    private int count;
    public Counter(){ ... } //effects: makes count = 0
    public int getCount(){ ... } //effects: return the value of count
    public void inc(){ ... } //effects: value of count incremented by one
}
```

Betrachte jetzt 2 mögliche Unterklassen Counter2 und Counter3 mit den folgenden zusätzlichen Operationen:

```
public class Counter2 {
    public Counter2(){ ... } //effects: makes count = 0
    public void inc(){ ... } //effects: value of count doubled
}
```

und

```
public class Counter3 {
    public Counter3(int n ){ ... } //effects: makes count contain n
    public void inc(int k){ ... } //effects: if n > 0 value of count
                                increased by k
}
```

Entscheiden und begründen Sie, ob Counter2 bzw. Counter3 legitime Unterklassen von Counter im Sinne der Vererbung von Spezifikationen sind.

Bitte wenden

Aufgabe 3 (13 P)

Gegeben sei das folgende Interface:

```
import java.util.*;

interface TraversableList extends List {
    void map(Function1 f); //imperatives Analogon zum funktionalen map
    Object foldl(Function2 f, Object init); // .... zum funktionalen foldl
    List filter(Predicate p); // .... zum funktionalen filter
}

interface Function1 { ... }
interface Function2 { ... }
interface Predicate { ... }
```

Vervollständigen Sie die angegebenen Schnittstellen, geben Sie Beispiel-Klassen für deren Implementierung an und implementieren Sie die Schnittstelle `TraversableList`!