

## Algorithmen und Datenstrukturen

### Übung 1 (AIN2)

1. Erstellen Sie mit einem beliebigen Texteditor (z.B. `vi` oder `kWrite`, `Edit`) eine Datei `Counter.java` mit dem in der Vorlesung vorgestellten Java-Quelltext der Klasse `Counter`.  
Übersetzen Sie die Klasse mit Hilfe des Compileraufrufs `javac Counter.java`.
2. Erweitern Sie die Klasse `Counter` um eine Methode `void decrement()`, mit der der Zählerstand um 1 verringert wird, sofern er noch positiv war.
3. Schreiben Sie eine eigene Klasse `CounterTest` mit einer statischen `main`-Methode (ähnlich wie in der Vorlesung), die einen Zähler anlegt, der in `for`-Schleifen mehrfach `incrementiert` und `decrementiert` wird.  
Übersetzen Sie auch diese Klasse. (Compileraufruf: `javac CounterTest.java`)  
Testen Sie mit diesem JAVA-Programm, ob die Implementierung der Klasse `Counter` korrekt ist. (Aufruf: `java CounterTest`)
4. Erweitern Sie Ihre Klasse `Counter` um eine Methode `void save()`, die den aktuellen Zählerstand in einer Hilfsgröße speichert und um eine Methode `void restore()`, mit der der Zähler wieder auf den zuletzt gesicherten Wert zurückgesetzt wird. Ein Aufruf von `restore` ohne vorheriges `save()` soll zu einer Fehlermeldung führen.

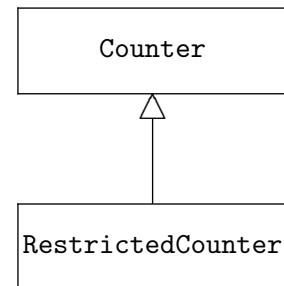
## 5. Ableitung einer Klasse `RestrictedCounter` aus `Counter`

Wenn man Zähler verwendet, um etwa die in einen Bus oder ein Flugzeug einsteigenden Personen zu zählen, sollten diese Zähler auch die maximal zulässige Anzahl von Passagieren kennen.

Schreiben Sie für solche Anwendungen eine aus `Counter` abgeleitete Klasse `RestrictedCounter`. Beim Erzeugen eines solchen Zählers soll der Anwender als Argument die maximale Kapazität mitgeben können. Die Methode `increment` muss dann beachten, dass die maximale Kapazität nicht überschritten werden darf und sollte gegebenenfalls eine Fehlermeldung ausgeben.

Implementieren Sie zusätzlich eine Methode `int freeCapacity()`, die die aktuell noch verfügbare freie Kapazität liefert.

Testen Sie Ihre Implementierung von `RestrictedCounter` mit folgendem Testprogramm:



```
public class RestrictedCounterTest {
    public static void main(String[] args) {
        RestrictedCounter c= new RestrictedCounter(15);
        for (int i=0; i<20; ++i) {
            c.increment();
            System.out.println("c=" + c.get() + " (free capacity=" +
                               c.freeCapacity() + ")" );
        }
        for (int i=0; i<20; ++i) {
            c.decrement();
            System.out.println("c=" + c.get() + " (free capacity=" +
                               c.freeCapacity() + ")" );
        }
    }
}
```