

## Übungsblatt 9

Ausgabe: 14.12.2000

### Aufgabe 33 (Klassen): 30 Punkte

Schauen Sie sich nochmal die Klasse *Stack* der Vorlesung 9 auf Seite 30 an. Sie sollen nun auf analoge Art und Weise eine Klasse *Menge* definieren. Objekte der Klasse *Menge* sollen eine bestimmte Anzahl von `int`-Werten speichern können, jeden `int`-Wert maximal einmal. Im einzelnen soll die Klasse *Menge* folgende Methoden besitzen:

- einen Konstruktor mit einem `int`-Parameter, der die maximale Größe der Menge festlegt.
- eine Methode `numberOfElements`, die die Anzahl an `int`-Werten in der Menge liefert.
- eine Methode `isFull`, die überprüft, ob die Menge voll ist, d.h. keine weiteren `int`-Werte mehr aufgenommen werden können. Geliefert werden soll ein entsprechender boolescher Wert.
- eine Methode `isElement`, die überprüft, ob ein übergebener `int`-Wert bereits in der Menge gespeichert ist. Geliefert werden soll ein entsprechender boolescher Wert.
- eine Methode `add`, mit deren Hilfe ein neuer `int`-Wert in die Menge aufgenommen werden kann. Die Methode soll als Ergebnis `true` liefern, wenn das Aufnehmen geklappt hat. Falls die Menge vorher bereits voll oder der `int`-Wert bereits in der Menge vorhanden ist, soll der Wert `false` geliefert werden.
- eine Methode `remove`, mit deren Hilfe ein übergebener `int`-Wert aus der Menge entfernt werden soll. Die Methode soll als Ergebnis `true` liefern, falls das Entfernen geklappt hat. Falls der `int`-Wert gar nicht in der Menge vorhanden ist, soll der Wert `false` geliefert werden.
- eine Methode `isEqual`, die als Parameter ein Mengen-Objekt übergeben bekommt und die überprüft, ob die beiden Mengen (die, für die die Methode `isEqual` aufgerufen wird, und die, die als aktueller Parameter übergeben wird) dieselben Elemente enthalten. Geliefert werden soll ein entsprechender boolescher Wert.
- eine Methode `union`, die als Parameter ein Mengen-Objekt übergeben bekommt und die alle `int`-Werte dieser Menge - sofern noch nicht enthalten - in die Menge integriert, für die die Methode `union` aufgerufen wird (solange noch „Platz ist“). Es soll kein Wert geliefert werden.
- eine Methode `difference`, die als Parameter ein Mengen-Objekt übergeben bekommt und die alle `int`-Werte dieser Menge - sofern enthalten - aus der Menge entfernt, für die die Methode `difference` aufgerufen wird. Es soll kein Wert geliefert werden.

Schreiben Sie ein kleines Testprogramm!

### Aufgabe 34 (Klassen/Objekte): 30 Punkte

Kopieren Sie sich die Datei `Firma.java` aus dem Verzeichnis `/user/fb10/dibo/java` des ARBI-Clusters (alternativ können Sie die Datei auch über die Java-Homepage laden).

Schauen Sie sich den Inhalt der Datei an. Sie enthält u.a. den „Verbund“ *Angestellter* und die Klasse *Firma*. Die Klasse *Firma* enthält neben der `main`-Funktion sechs weitere Methoden. Wandeln Sie die Datei so um, dass aus dem Verbund *Angestellter* eine Klasse *Angestellter* (ohne Schlüsselwort `public`) wird und die sechs Funktionen der Klasse *Firma* zu Methoden der Klasse *Angestellter* werden. Passen Sie anschließend auch die `main`-Funktion so an, dass sie mit Objekten der Klasse *Firma* arbeitet.

### Aufgabe 35 (Arrays): 40 Punkte

Quadratien ist ein quadratisches Gebiet aus insgesamt  $10 \times 10$  quadratischen Feldern. Das Feld in der Nordwest-Ecke hat die Zeilennummer 0 und die Spaltennummer 0. Westlich von Quadratien befindet sich West-Quadratien und nördlich von Quadratien befindet sich Nord-Quadratien. Beide sind genauso groß wie Quadratien selbst.

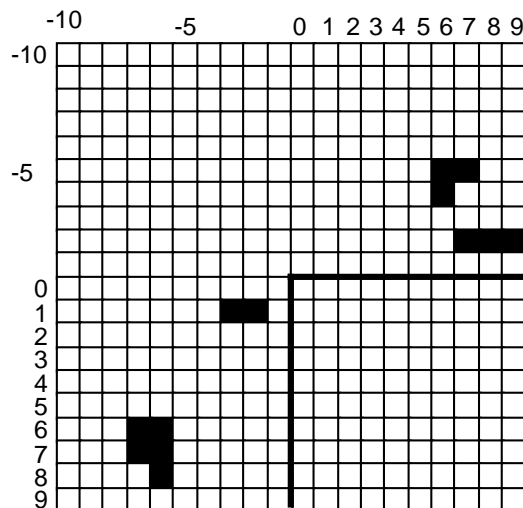
Das Wetter in Quadratien wird durch quadratische Wolken bestimmt, die genau ein Feld groß sind. Solche Wolken entstehen in West- und Nord-Quadratien und rücken getaktet über Quadratien vor, und zwar von Norden nach Süden 2 Felder pro Takt und von Westen nach Osten 3 Felder pro Takt. Es regnet überall dort, wo sich nach einem Vorrücken sowohl eine Nord-Süd- als auch eine West-Ost-Wolke befindet.

Die Wolkenvorhersage gibt an, an welchen Stellen (in der Form: Zeilennummer/Spaltennummer) in West- und Nord-Quadratien sich zum aktuellen Zeitpunkt Wolken befinden. Daraus lässt sich dann ermitteln, wo es in Quadratien regnen wird.

Beispiel:

Wolkenvorhersage:

-5/6 -4/6 -2/7 7/-6 -2/9 -5/7 1/-2 8/-6 6/-6 7/-7 6/-7 1/-3 -2/8



Es wird an folgenden Stellen regnen: 1/6, 1/7, 8/9

Entwickeln Sie ein Java-Programm, welches folgendes leistet:

1. Einlesen einer Wolkenvorhersage.
2. Ausgabe, wo in Quadratien Regen fällt.