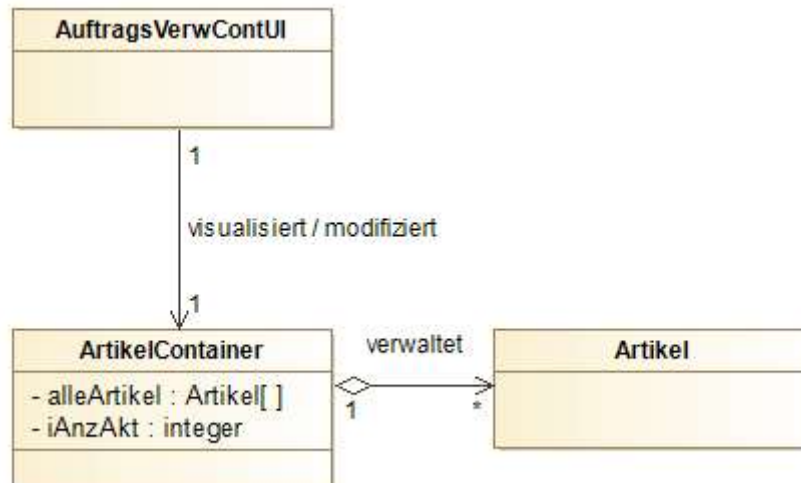


### 3 Containerklassen und 3-Schichten-Architektur

*Ziel: Entsprechend der 3-Schicht-Architektur soll eine Containerklasse zum Verwalten von Artikel Objekten erstellt werden.*

- 3.1. Gegeben ist die bereits früher erstellte Fachkonzeptklasse Artikel und eine einfache UI-/Startklasse. Die Beschreibung der Methoden liegt in Form JavaDoc vor.  
(Siehe Moodle: DatenArtikelContainer)



Kopieren Sie sich die Klassen `Artikel` und `AuftragsVerwContUI` in eine neu erstelltes Package. Implementieren Sie die Klasse `ArtikelContainer` und testen Sie die Methoden. Gehen Sie schrittweise vor indem Sie Teile von `AuftragsVerwContUI` auskommentieren.

```

public class ArtikelContainer
{
    private Artikel[] alleArtikel;
    private int iAnzAkt;

    * Konstruktor, erzeugt Array für max 20 Artikel (wird dynamisch verlängert)
    public ArtikelContainer()
    {
        // this.alleArtikel = new Artikel[20];
        // iAnzAkt = 0;
        this(20); // Konstruktor-Verkettung
    }

    * Konstruktor, erzeugt Array für iAnzArtikel Artikel
    public ArtikelContainer(int iMaxArtikel)
    {
        this.alleArtikel = new Artikel[iMaxArtikel];
        iAnzAkt = 0;
    }

    * @return liefert aktuelle Anzahl gespeicherter Artikel
    public int getIANzAkt()
    {
        return this.iAnzAkt;
    }

    * neuerArtikel wird in Array gespeichert und die Anzahl aktualisiert
    public void speichereArtikel(Artikel neuerArtikel)
    {
        if (this.iAnzAkt == this.alleArtikel.length) // wenn Array voll
        {
            // Verlängern auf doppelte Länge
            this.alleArtikel = Arrays.copyOf(this.alleArtikel, alleArtikel.length * 2);
        }
        this.alleArtikel[iAnzAkt] = neuerArtikel;
        iAnzAkt++;
    }

    * Sucht den Artikel der die gegebene iArtikelNr besitzt;
    public Artikel sucheArtikelNachNr(int iArtikelNr)
    {
        Artikel gesucht = null;
        int iInd = 0;

        while (iInd < this.iAnzAkt && this.alleArtikel[iInd].getINr() != iArtikelNr)
        {
            iInd++;
        }

        if (iInd < this.iAnzAkt) // ArtikelNr wurde gefunden
        {
            gesucht = this.alleArtikel[iInd];
        }

        return gesucht; // null, wenn nicht gefunden
    }
}

```

```

    * sucht alle Artikel mit sBezeichnung == sSuchBezeichnung
    public Artikel[] sucheArtikelNachBezeichnung(String sSuchBezeichnung)
    {
        Artikel[] gefundeneArtikel = null;
        int iAnz = 0;
        for (int i = 0; i < this.iAnzAkt; i++)
        {
            if (this.alleArtikel[i].getSBezeichnung().equals(sSuchBezeichnung))
            {
                if (gefundeneArtikel == null) gefundeneArtikel = new Artikel[1];
                if (iAnz == gefundeneArtikel.length) // gilt ab dem 2. gefundenen Artikel
                    gefundeneArtikel = Arrays.copyOf(gefundeneArtikel,
                                                        gefundeneArtikel.length + 1);
                gefundeneArtikel[iAnz] = this.alleArtikel[i];
                iAnz++;
            }
        }
        return gefundeneArtikel; // null, wenn nichts gefunden
    }
}

```

#### 4 Zwei-dimensionale-Arrays

*Ziel: Der Umgang mit zwei-dimensionalen Arrays soll am Beispiel eines Memory-Spiels geübt werden.*

- 4.1. Erstellen Sie ein neues Package `memorySpiel`. Laden Sie von Moodle die Datei `DatenMemorySpiel` herunter und entpacken Sie die Dateien in den Ordner des Packages. Nach einem „Refresh“ sollten die Klassen sichtbar sein.

Ergänzt werden muss nur in der Klasse `Memory` der Inhalt des Konstruktors, entsprechend der dort angegebenen Kommentare.

Das Array `aMaske` hat folgenden Inhalt:      Das Array `aVorlage` z.B. folgenden:  
(jeweils ohne Ränder):

```

-1-2-3-4-5-6-7-8-
1 #|#|#|#|#|#|#|
2 #|#|#|#|#|#|#|
3 #|#|#|#|#|#|#|
4 #|#|#|#|#|#|#|
5 #|#|#|#|#|#|#|
6 #|#|#|#|#|#|#|
7 #|#|#|#|#|#|#|
8 #|#|#|#|#|#|#|
-----

```

```

-1-2-3-4-5-6-7-8-
1 2|B|9|8|2|1|)|-|
2 .|=|>|;|C|&|,|@|
3 5|?|/|>|+|'|;|0|
4 6|,|'|4|@|A|8|(|
5 %|3|*|)|$|0|5|(|
6 7|%|6|&|.|<|?|B|
7 C|A|/|9|:|1|=|:|
8 4|7|*| -|<|3|$|+|
-----

```