

<input type="checkbox"/> Gr. 1, Dr. G. Kronberger	Name <u>Daniel Kreuzer</u>	Aufwand in h <u>11</u>
<input type="checkbox"/> Gr. 2, Dr. H. Gruber		
<input type="checkbox"/> Gr. 3, Dr. D. Auer	Punkte _____	Kurzzeichen Tutor / Übungsleiter _____ / _____

Pascal bietet zwar einen Datentyp zur Repräsentation von Mengen (*SET OF ...*), allerdings können die Elemente solcher Mengen nur ganze Zahlen aus dem Bereich von 0 .. 255, Zeichen (also Werte des Datentyps *CHAR*) oder Werte eines Aufzählungsdantentyps sein. – Wir haben diesen *SET*-Datentyp nicht behandelt, da er zu restriktiv ist und in anderen Sprachen (z. B. in C, C++ und Java) so nicht zur Verfügung steht.

Klassenbasierte objektorientierte Programmiersprachen (wie Borland Pascal, C++, Java und C#) bieten mit eben diesen Klassen aber die Möglichkeit, benutzerdefinierte Datentypen auf einfachere Art und Weise zu realisieren, als es bisher mit dem ADT-Konzept möglich war.

## 1. Zeichenketten-Menge als Klasse (10 Punkte)

Entwerfen Sie eine Klasse *SOS* (für *set of strings*), deren Objekte eine Realisierung des mathematischen Konzepts einer Menge von Elementen des Datentyps *STRING* darstellen. Verwenden Sie zur internen Speicherung der Elemente eine Datenkomponente *elements* (ein Feld von Zeichenketten) und eine Datenkomponente *n* (Zähler für die Anzahl der Elemente).

Stellen Sie mindestens folgende Methoden zur Verfügung:

- Einen Konstruktor und einen Destruktor,
- die Methoden *Empty*, *Cardinality*, *Add*, *Remove* und *Contains* sowie
- Methoden, welche die aus der Mathematik bekannten Mengenoperationen *Union*, *Intersection*, *Difference* und *Subset* implementieren.

Testen Sie Ihre Klasse ausführlich, indem Sie statische und dynamische Objekte anlegen und alle Operationen darauf ausführen.

## 2. Säcke (14 Punkte)

Jeder kennt einen Sack. – Denken Sie z. B. an einen Plastiksack: In einen solchen kann man mehrere, auch gleiche Dinge hineinstecken.

Im mathematischen Sinn gilt: Ein Sack (*bag*) ist eine Menge (*set*), die Elemente auch mehrfach enthalten kann. Somit könnte man auf die Idee kommen, die Klasse *BOS* (für *bag of strings*) von der Klasse *SOS* aus Aufgabe 1 abzuleiten<sup>1</sup>.

Versuchen Sie, möglichst viel von der Basisklasse *SOS* zu nutzen, indem Sie in der abgeleiteten Klasse *BOS* nur eine weitere Datenkomponente *counters* (ein Feld von ganzen Zahlen) hinzufügen, welche für alle Elemente in *elements* angibt, wie oft diese Elemente im Sack vorkommen.

Passen Sie alle Mengenoperationen in *BOS* an die Semantik von Säcken an.

Testen Sie Ihre Klasse ausführlich, indem sie statische und dynamische Objekte anlegen und alle Operationen darauf ausführen.

---

<sup>1</sup> Ob das im Sinne der OOP besonders klug ist, werden wir in einem höheren Semester bei der Behandlung des Liskov'schen Substitutionsprinzips näher besprechen.