

Übungen zu Einführung in die Informatik II

Aufgabe 3 **Generische Klassen**

Entwickeln Sie eine generische Klasse `Set<E>`. Ein Objekt der Klasse `Set<E>` soll eine Menge von Elementen vom Typ `E` darstellen. Zur Implementierung können Sie die Klasse `List<E>` benutzen, die wie folgt definiert ist:

```
public class List<E>{
    public E info;
    public List<E> next;

    public List(E e){
        info = e;
        next = null;
    }
    public List(E e, List<E> n){
        info = e;
        next = n;
    }
    public void insert(E e){
        next = new List<E>(info,next);
        info = e;
    }
}
```

Die Klasse `Set<E>` muss folgende Methoden bereitstellen:

- eine Objekt-Methode `public boolean isIn(E elem)`, die testet, ob ein Element mit dem selben Inhalt wie `elem` in dem `Set`-Objekt enthalten ist.
- eine Objekt-Methode `public void add(E elem)`, die ein Element `elem` dem `Set`-Objekt hinzufügt, falls es nicht schon vorhanden ist.
- eine Objekt-Methode `public void intersect(Set<E> s)` zur Berechnung der Schnittmenge zwischen dem aktuellen Objekt (für das die Methode aufgerufen wurde) und dem angegebenen Argument. Anschließend soll das aktuelle Objekt nur noch die Elemente enthalten, die in beiden Ursprungsmengen enthalten waren.

Verwenden Sie zum Vergleichen der Elemente den Inhaltsvergleich. Nehmen Sie dabei an, dass die Methode `equals` des Objekts `elem` den Wert `true` zurückliefert, wenn sie mit einem Element gleichen Inhalts aufgerufen wird.

Aufgabe 4 **GUI: Puzzle**

Implementieren Sie inklusive graphischer Benutzeroberfläche (GUI) ein *Schiebe-Puzzle*. Für ein Spiel der Größe 3×3 sind 8 Spielsteine mit den Werten $1, \dots, 8$ auf dem Spielfeld und das neunte

Feld bleibt frei. Ein Stein kann genau dann bewegt werden, wenn er direkter Nachbar des freien Feldes ist.

4	3	5
2	1	6
7	8	→

4	3	5
2	1	6
7		8

Implementieren Sie Ihr Puzzle so, dass man die Größe des Feldes frei wählen kann (in unserem Beispiel ist das 3). Um das Spiel spannender zu machen, ist es natürlich sinnvoll mit einem unsortierten Feld zu beginnen. Stellen Sie dazu eine Methode `shuffle()` zur Verfügung, die das Feld durcheinander bringt. In dieser Methode werden Sie vermutlich Zufallszahlen benötigen, die man mit den Methoden der Klasse `Random` erzeugen kann:

```
import java.util.Random;
...

Random random = new Random(System.currentTimeMillis());
/* Creates a new random number generator using a single long
 * seed. */
...

int r = random.nextInt(4);
/* Returns a pseudorandom, uniformly distributed int value
 * between 0 (inclusive) and the specified value (exclusive),
 * drawn from this random number generator's sequence. */
```