

## Formale Systeme II: Anwendung (Sommer 2019) Übungsblatt „KeY“

### Aufgabe 1 Bounded Stack

Sie erhalten die Aufgabe, eine Komponente zu entwerfen, die einen begrenzten Kellerspeicher (Stack, LIFO) für **int**-Werte implementiert. Die Begrenzung bezieht sich auf die maximale Größe des Speichers, also die Zahl der darauf befindlichen Daten-Elemente.

Sie sollen dazu die Klasse `BoundedStack`, deren Gerüst unten abgebildet ist,

1. spezifizieren,
2. implementieren, und dann
3. verifizieren, dass die Implementierung die Spezifikation erfüllt.

```
class BoundedStack {  
    // Felder dürfen hinzugefügt werden ...  
  
    BoundedStack(int capacity) { /* ... */ }  
    int pop() { /* ... */ }  
    void push(int value) { /* ... */ }  
    int size() { /* ... */ }  
    int capacity() { /* ... */ }  
}
```

Die natürlich-sprachlichen Spezifikationen der Methoden der Klasse sind dabei:

**Konstruktor** Ein neuer leerer Kellerspeicher wird erzeugt, dessen maximal Kapazität dem Argument entspricht.

**pop** Wenn der Speicher nicht leer ist, wird das oberste Element zurückgeliefert.  
Wenn der Speicher leer ist, wird eine `IndexOutOfBoundsException` geworfen.

**push** Wenn der Speicher nicht voll<sup>1</sup> ist, wird das Argument als oberstes Element auf den Kellerspeicher gelegt.  
Wenn der Speicher voll ist, wird eine `IndexOutOfBoundsException` geworfen.

**size** Liefert die Anzahl der momentan auf dem Kellerspeicher gespeicherten Daten zurück. Diese Methode ist *pur*.

**capacity** Liefert die maximale Kapazität dieses Kellerspeichers zurück. Diese Methode ist *pur*.

---

<sup>1</sup>d.h., die Zahl der Elemente gerade der maximalen Kapazität entspricht

Sie finden das Gerüst der Klasse zum Herunterladen auf der Homepage der Vorlesung. Gehen Sie nun folgendermaßen vor:

1. *Spezifizieren Sie zunächst die Methoden mittels JML.*
2. *Verifizieren Sie die Testmethode, die mitgeliefert wird.*
3. *Implementieren Sie dann die Methoden.*
4. *Verifizieren Sie, dass alle Methoden ihren Kontrakt einhalten.*

Im zweiten Schritt ist es möglich, Code der die Methoden benutzt zu verifizieren, obwohl die Methoden noch gar nicht implementiert sind. Dies funktioniert, da KeY die Methoden-Aufrufe durch den Vertrag abstrahiert – und die Verträge sind ja vorhanden.

## **Aufgabe 2 SaddlebackSearch**

Laden Sie die Datei `SaddlebackSearch.java` vom Server herunter und beweisen Sie den Vertrag der einzigen Methode. Die Datei enthält weitere Hinweise.