

Einzelprüfung „Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft)“

Einzelprüfungsnummer 46115 / 2021 / Frühjahr

Thema 2 / Teilaufgabe 2 / Aufgabe 2

(Java-Klasse *Stack*)

Stichwörter: Stapel (Stack), Implementierung in Java

Gegeben sei die folgende Java-Implementierung eines Stacks.

```
class Stack {
    private Item head;

    public Stack() {
        head = null;
    }

    public void push(int val) {
        if (head == null) {
            head = new Item(val, null);
        } else {
            head = new Item(val, head);
        }
    }

    public int pop() {
        // ...
    }

    public int size() {
        // ...
    }

    public int min() {
        // ...
    }
}

class Item {
    private int val;
    private Item next;

    public Item(int val, Item next) {
        this.val = val;
        this.next = next;
    }
}
```

- (a) Implementieren Sie die Methode `pop` in einer objektorientierten Programmiersprache Ihrer Wahl, die das erste Item des Stacks entfernt und seinen Wert zurückgibt. Ist kein Wert im Stack enthalten, so soll dies mit einer `IndexOutOfBoundsException` oder Ähnlichem gemeldet werden.

Beschreiben Sie nun jeweils die notwendigen Änderungen an den bisherigen Implementierungen, die für die Realisierung der folgenden Methoden notwendig sind.

Lösungsvorschlag

```
public int pop() {
    if (head != null) {
        int val = head.val;
        size--;
        head = head.next;
        return val;
    } else {
        throw new IndexOutOfBoundsException("The stack is empty");
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java)

- (b) `size` gibt in Laufzeit $\mathcal{O}(1)$ die Anzahl der enthaltenen Items zurück.

Lösungsvorschlag

```
public void push(int val) {
    if (head == null) {
        head = new Item(val, null);
    } else {
        head = new Item(val, head);
    }
    if (min > val) {
        min = val;
    }
    size++;
}

public int pop() {
    if (head != null) {
        int val = head.val;
        size--;
        head = head.next;
        return val;
    } else {
        throw new IndexOutOfBoundsException("The stack is empty");
    }
}

public int size() {
    return size;
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java)

- (c) `min` gibt (zu jedem Zeitpunkt) in Laufzeit $\mathcal{O}(1)$ den Wert des kleinsten Elements im Stack zurück.

Lösungsvorschlag

```
public void push(int val) {  
    if (head == null) {  
        head = new Item(val, null);  
    } else {
```

```
        head = new Item(val, head);
    }
    if (min > val) {
        min = val;
    }
    size++;
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java)

```
public int min() {
    return min;
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java)

Sie dürfen jeweils alle anderen angegebenen Methoden der Klasse verwenden, auch wenn Sie diese nicht implementiert haben. Sie können anstelle von objektorientiertem Quellcode auch eine informelle Beschreibung Ihrer Änderungen angeben.

Additum: Kompletter Java-Code

```
class Stack {
    private Item head;

    private int size;
    private int min = Integer.MAX_VALUE;

    public Stack() {
        head = null;
    }

    public void push(int val) {
        if (head == null) {
            head = new Item(val, null);
        } else {
            head = new Item(val, head);
        }
        if (min > val) {
            min = val;
        }
        size++;
    }

    public int pop() {
        if (head != null) {
            int val = head.val;
            size--;
            head = head.next;
            return val;
        } else {
            // TODO: Handle empty stack
        }
    }
}
```

```
        throw new IndexOutOfBoundsException("The stack is empty");
    }
}

public int size() {
    return size;
}

public int min() {
    return min;
}

class Item {
    private int val;
    private Item next;

    public Item(int val, Item next) {
        this.val = val;
        this.next = next;
    }
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java](https://github.com/src/main/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/Stack.java)

```
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertThrows;

import org.junit.Test;

public class StackTest {

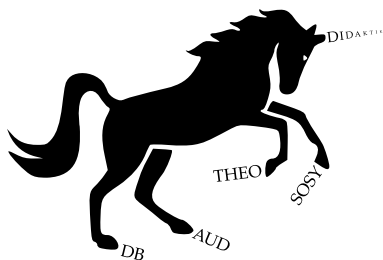
    private Stack makeStack() {
        Stack stack = new Stack();
        stack.push(1);
        stack.push(2);
        stack.push(3);
        return stack;
    }

    @Test
    public void methodPop() {
        Stack stack = makeStack();
        assertEquals(3, stack.pop());
        assertEquals(stack.size(), 2);
        assertEquals(2, stack.pop());
        assertEquals(stack.size(), 1);
        assertEquals(1, stack.pop());
        assertEquals(stack.size(), 0);
        assertThrows(IndexOutOfBoundsException.class, () -> {
            stack.pop();
        });
    }
}
```

```
@Test
public void methodSize() {
    Stack stack = makeStack();
    assertEquals(stack.size(), 3);
}

@Test
public void methodMin() {
    Stack stack = makeStack();
    assertEquals(stack.min(), 1);
}
}
```

Code-Beispiel auf Github ansehen: [src/test/java/org/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/StackTest.java](https://github.com/bschlangaul/examen/examen_46115/jahr_2021/fruehjahr/StackTest.java)



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben-text/blob/main/Examen/46115/2021/03/Thema-2/Teilaufgabe-2/Aufgabe-2.tex>