FACH
HOCHSCHULE
LÜBECK
University of Applied Sciences

Informatik II

Präsentation
Stack

Thomas Jürgensen

popTop()

- PopTop : Stack → T x Stack
  - In-Place → ausgabe Stack entfällt

- Axiom: popTop(s) = top(s), pop(s), falls s nicht leer
  - top() speichern, pop(), gespeichertes top() zurückgeben
    - Pop(): O(1); top(): O(1) → poptop() O(1)

AbstractStack<E>

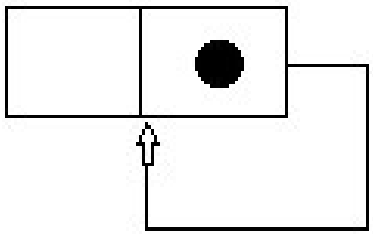
- Implementiert interface Stack<E>
  - → da List- und ArrayStack erben, implementieren diese ebenso das Interface
- Gemeinsame Methoden:
  - poptop() , da lediglich Methoden aufgerufen werden
  - isEqualTo(Stack<E> s)

ListStack<E>

- Erstelle ListStack mit einer „Null“-Zelle 'top', der die als Zeiger dient.

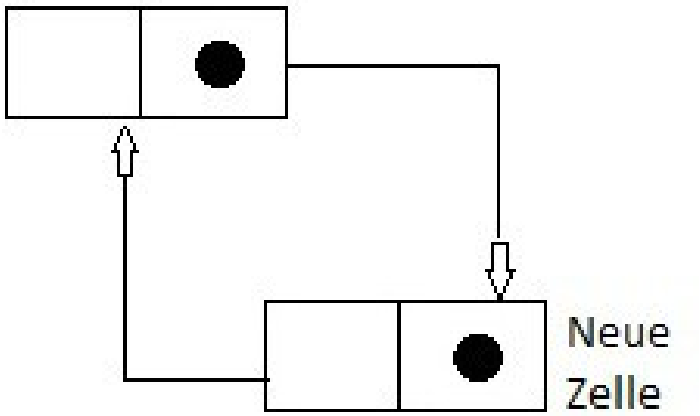
ListStack<E>
push(E e)

- Neue Zelle mit Verweis auf sich selbst ('top' zeigt nun auf neue Zelle),



ListStack<E>
push(E e)

- Füge neue Zelle an next hinzu und setze dessen Verweis auf alte Zelle
- Setze top auf neue Zelle
  - Durch Zeiger: O(1)



ListStack<E>
pop(), top()

- Falls next == null ist, setze top auf null

  → leerer Stack

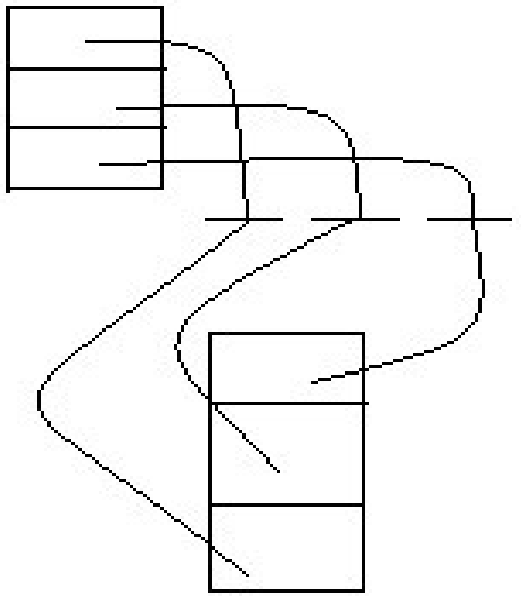
- Setze top einen weiter auf top.next
  - Nachteil: Die Liste wird immer größer, da keine Verweise direkt gelöscht werden; es wird lediglich der Zeiger verschoben

- top(): gebe Inhalt des Elements aus, auf das der Zeiger verweist
  - O(1)

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

StackTest und toList()

- Für beide Klassen identisch:
  - Erstelle Testklasse
  - Erstelle Stack und befülle ihn mit Test-String
  - Um auszulesen: Temporären Stack erstellen und beim auslesen des Originalstacks befüllen, danach wieder (umgekehrt) befüllen.
  - Da 2 Schleifen je O(n)
    - → O(2n) = O(n)



IsEqualTo (Stack<E> s)

- In AbstractStack
  - Jeder Stack hat eigene ToList() (ArrayList, LinkedList)
- Prüfen, ob eine oder beide Listen leer
- Falls Liste und poptop() übereinstimmen, weiter machen, bis Stack leer
  - Falls nicht, Stack wieder mit Hilfe der Liste befüllen
- Leeren Stack wieder mit Hilfe der Liste komplett befüllen

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

Junit Test

- Testklasse erstellen
- Identische Stacks erstellen
- Axiom auf beide Stacks ausführen
- Auf Gleichheit bzw error überprüfen
  - AssertTrue, assertEquals, assertThrows

PostFixInterpreter

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

- Beinhaltet 2 Stacks, 1 TempVariable, 1 Methode
  - Tmp speichter poptop(), um push(poptop()+tmp) auszuführen
- Interpreter:
  - Prüft auf zahlen, +,-,*,/,^,! sowie öffnende und schließende Klammern

PostFixInterpreter

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

```
} else if (s.charAt(i) == '+') {
    tmp = stack.poptop();
    stack.push(stack.poptop() + tmp);
```

PostFixInterpreter

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

```
if (s.charAt(i) >= '0' && s.charAt(i) <= '9') {
    stack.push((double) s.charAt(i) - '0');
```

048	30	0
049	31	1
050	32	2
051	33	3
052	34	4
053	35	5
054	36	6
055	37	7
056	38	8
057	39	9