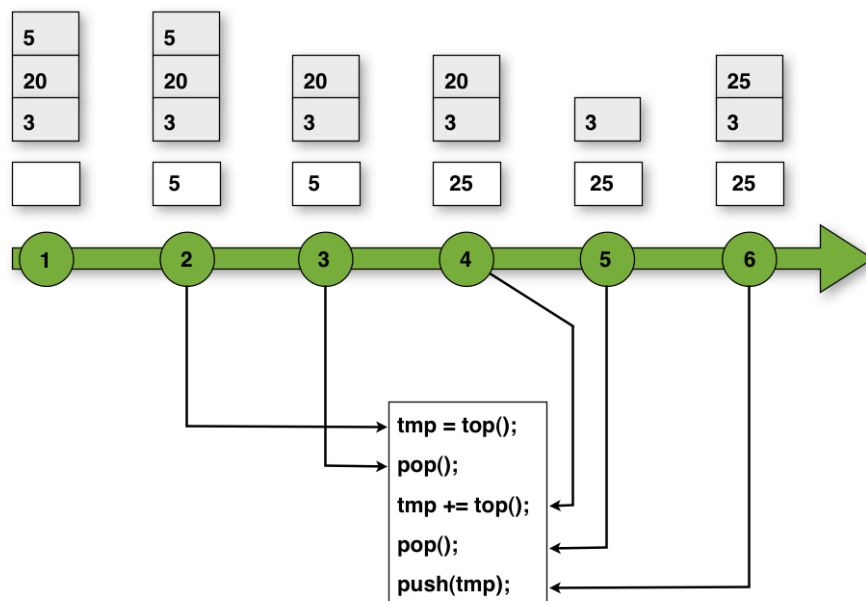


## 21.4 Einfache Berechnungen, Teil 1



Die Abbildung zeigt die Arbeitsweise der `add()`-Methode. Die Klasse **Stackmaschine** hat als Tochterklasse von **Stack** Vollzugriff auf alle öffentlichen Methoden der Klasse **Stack** wie `push()` und `pop()`. Die Klasse **Stackmaschine** hat dagegen überhaupt keine Ahnung, wie innerhalb von Stack die Daten verwaltet werden. Daher kann **Stackmaschine** zum Addieren oder Multiplizieren auch nicht direkt auf einen Array oder irgendwelche Knoten einer dynamischen Liste zugreifen,

denn solch ein Array oder solche Knoten existieren ja eventuell überhaupt gar nicht. Die **Stackmaschine** kann ausschließlich die vereinbarten öffentlichen Methode des **Stacks** verwenden.

### Übung 21.4-1

1. Erstellen Sie für den arithmetischen Ausdruck  $(4 + 6) / (8 - 3) + 2$  den Stackcode.
2. Erstellen Sie den Stackcode für den Ausdruck  $4 + 6 / 8 - 3 + 2$ .

### Übung 21.4-2

Schreiben Sie eine Klasse **Test** für Ihre Stackmaschine, die Sie bereits programmiert haben. In dieser Klasse können Sie die Methoden der Stackmaschine auf arithmetische Ausdrücke anwenden. Eine solche Testmethode könnte beispielsweise so aussehen:

```

public void test1()
{
    sm = new Stackmaschine();
    sm.push(7);
    sm.push(3);
    sm.add();
    sm.push(12);
    sm.push(3.14);
    sm.sub();
    sm.div();
    System.out.println("Ergebnis = " + sm.top());
}
    
```