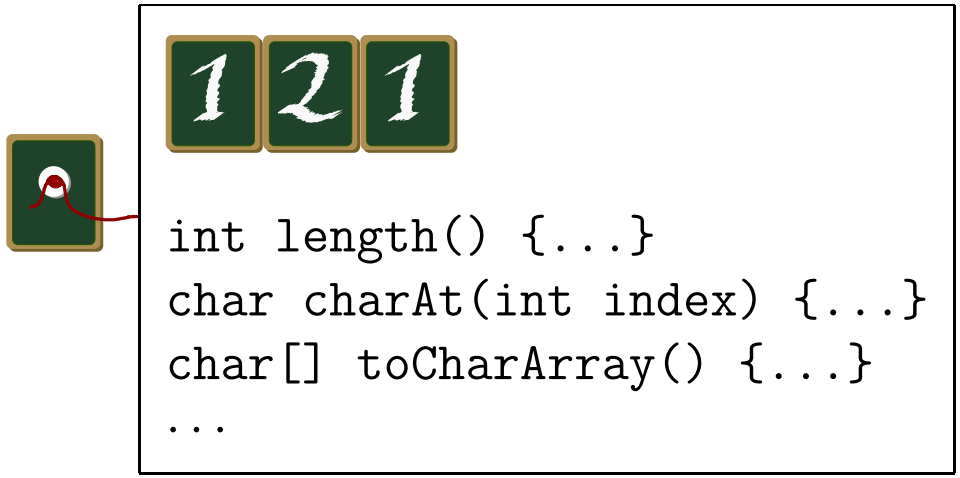
Tagesprogramm

Datenabstraktion

Queue and Stack

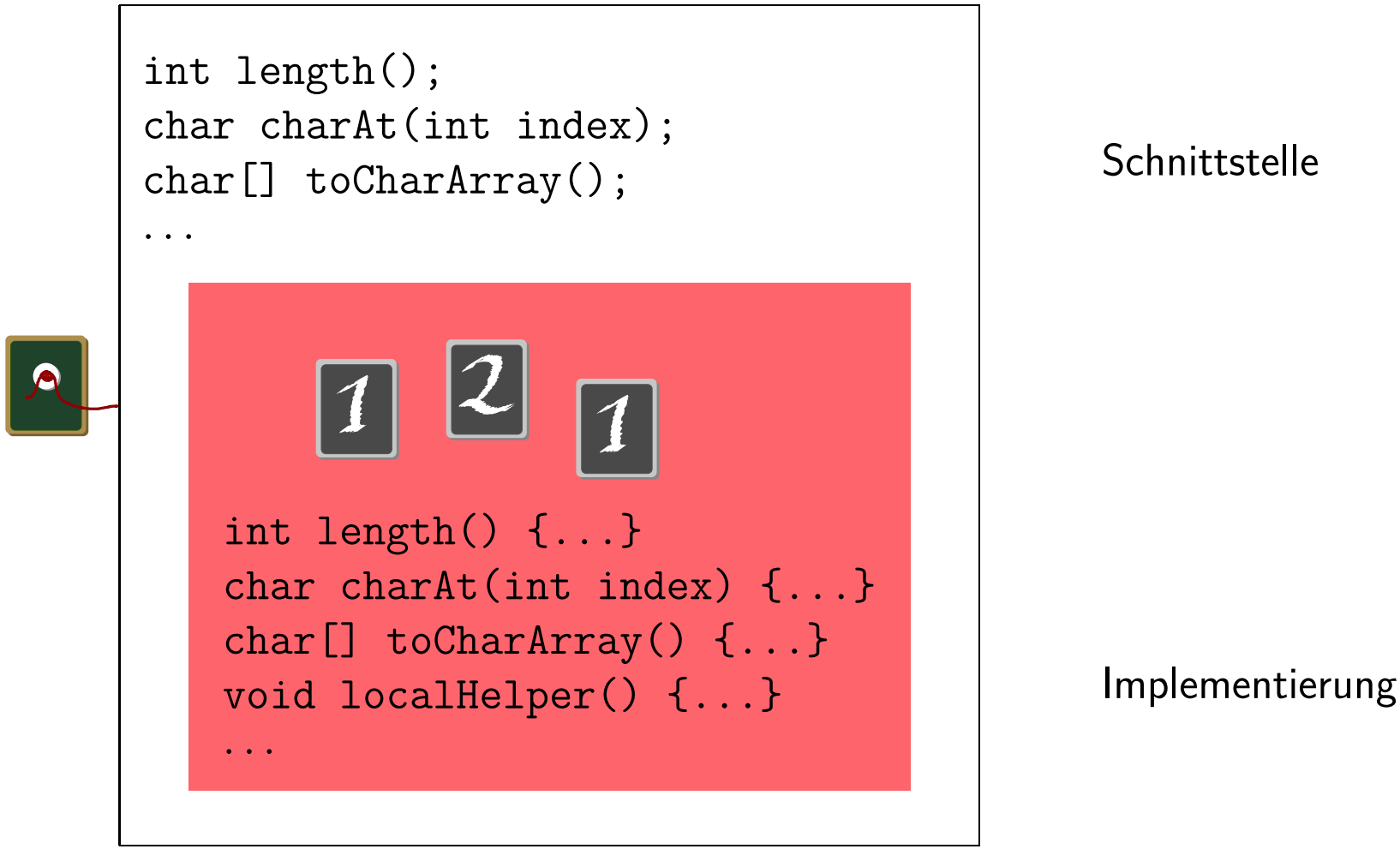
Datenkapselung

Datenkapselung = Daten und Methoden bilden Einheit



Data-Hiding

Data-Hiding = „verstecken“ von Daten vor direkten Zugriffen von außen



Datenabstraktion

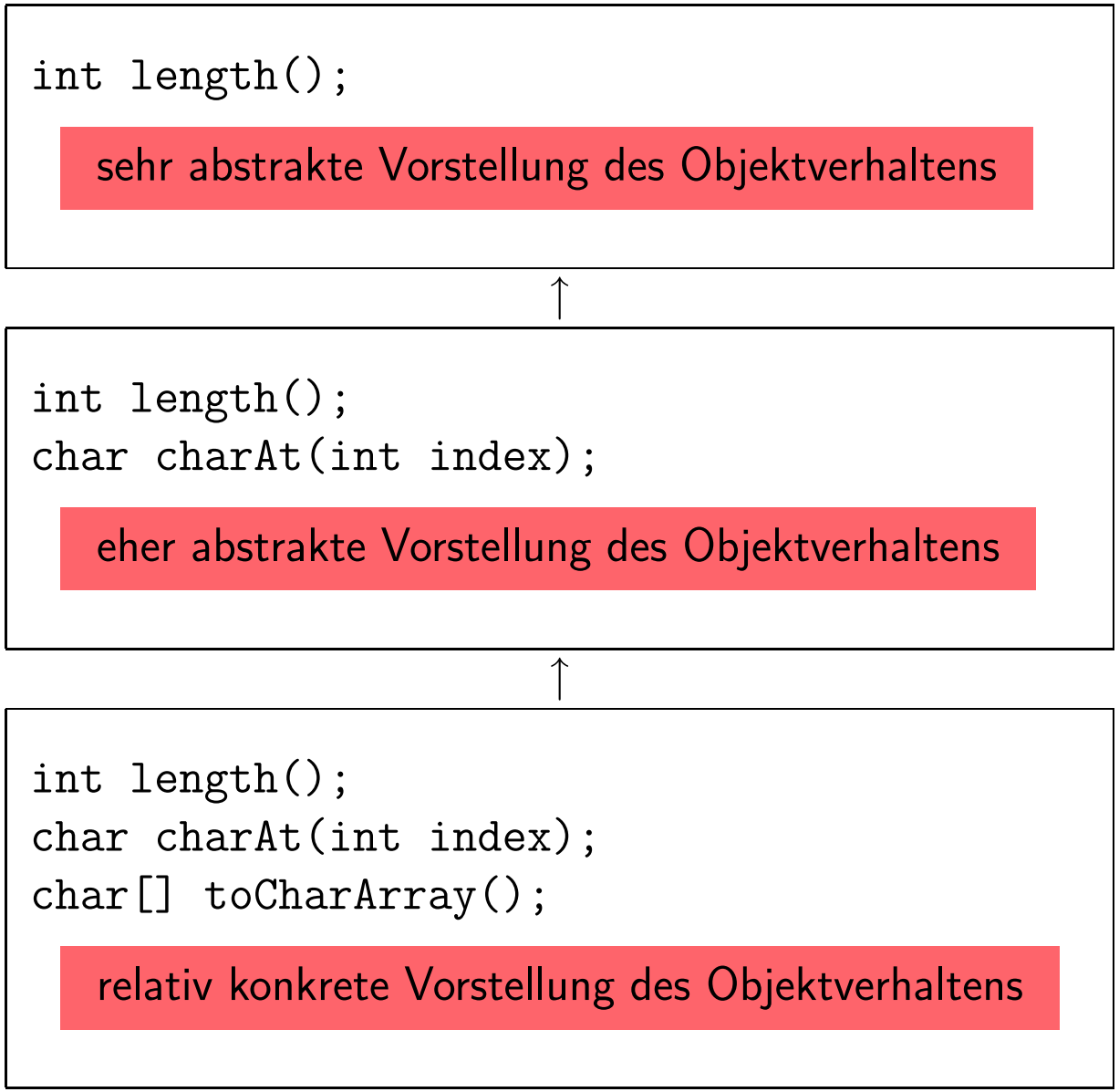
Datenabstraktion = Datenkapselung und Data-Hiding

```
int length();
char charAt(int index);
char[] toCharArray();
...
```

abstrakte Vorstellung des Objektverhaltens

Objekt

Abstraktionshierarchie



Aufgabe: Warum Abstraktion?

Suchen Sie in Gruppen zu 2 bis 3 Personen eine Antwort auf diese Frage:

Warum verwenden wir Abstraktion und Abstraktionshierarchien?

Zeit: 3 Minuten

Queue als Abstraktion

```
private static void useQueue(Queue<String> queue) {
    for (int i = 1; i <= 10; i++) {
        queue.offer("String " + i);
    }

    String s;
    while ((s = queue.poll()) != null) {
        System.out.println(s);
    }
}
```

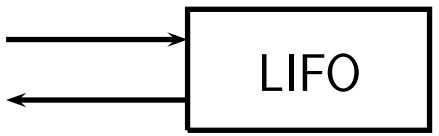
einige Methoden von Queue:

```
boolean offer(E e); // fügt e hinten ein, true bei Erfolg
E poll();           // entfernt erstes Element, gibt es zurück
E peek();           // gibt erstes Element zurück
```

Deque als Stack

```
private static void useStack(Deque<String> stack) {
    for (int i = 1; i <= 10; i++) {
        stack.offerFirst("String " + i);
    }

    String s;
    while ((s = stack.poll()) != null) {
        System.out.println(s);
    }
}
```



Methode von Deque zusätzlich zu denen von Queue:

```
boolean offerFirst(E e);
// fügt e vorne ein, true bei Erfolg
```

Aufgabe: LIFO oder FIFO?

In Deque gibt es neben `offer` (entspricht `offerLast`) auch `offerFirst`. Neben `poll` (entspricht `pollFirst`) gibt es auch `pollLast`.

Durch Kombinationen welcher `offer`-Varianten mit welchen `poll`-Varianten ergibt sich ein LIFO- bzw. FIFO-Verhalten?