

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 8.5.-12.5.2017 besprochen.

## Übungsblatt 2

### Aufgabe 1

#### Sequenzdiagramm

Nehmen Sie an, dass die in den folgenden Methoden benutzten Klassen und Methoden ihren Definitionen entsprechend verwendet und aufgerufen werden. Erstellen Sie jeweils ein **Sequenzdiagramm**, das den Ablauf eines Aufrufs der in der Teilaufgabe genannten Methode zeigt. Alle Methodenaufrufe innerhalb der Sequenzdiagramme sollen durch Pfeile visualisiert werden.

### Aufgabe 1

Visualisieren Sie einen Aufruf der Methode `remove`.

```
public Item remove(List l, int x, DataPool p) {
    Item it = null;
    if (x < l.size()) {
        it = p.detach(l.atPosition(x));
    }
    return it;
}
```

### Aufgabe 2

Visualisieren Sie einen Aufruf der Methode `average`.

```
public double average(Iterator<Data> it) {
    double result = 0.0;
    double count = 0.0;
    while (it.hasNext()) {
        result += it.next().getValue();
        count++;
    }
    return result / count;
}
```

### Aufgabe 3

#### Sequenzdiagramm

Zeichnen Sie ein Sequenzdiagramm, das alle aus einem Aufruf von `run` resultierenden Konstruktor- und Methodenaufrufe zeigt. Die Abläufe in den Methoden `exec()`, `get()` und `doIt()` sind ohne Bedeutung.

```
class C1 {
    public boolean exec() { ... }
}
```

```
class C2 {
    public int get() { ... }
    public int doIt() { ... }
}
```

```
class C3 {
    public int compute(C1 p1, C2 p2) {
        if (p1.exec()) {
            return p2.doIt();
        }
        return -1;
    }
}
```

```
public class C4 {
    public int run(C2 c) {
        C3 ref = new C3();
        return ref.compute(new C1(), c)
            + c.get();
    }
}
```