

Arbeitsblatt: Lock-Free Number Range

In diesem Arbeitsblatt untersuchen und reparieren Sie eine Lock-Free Implementierung der Klasse `NumberRange`. Die Klasse `NumberRange` implementiert ein Intervall und stellt sicher, dass die untere Intervallgrenze immer kleiner oder gleich der oberen Intervallgrenze ist.

Die folgende Klasse finden Sie auch im Wochenprojekt 06_LE_Lock_Free.zip.

```
public class NumberRangeWrong {

    // INVARIANT: lower <= upper is NOT GUARANTEED
    private final AtomicInteger lower = new AtomicInteger(0);
    private final AtomicInteger upper = new AtomicInteger(0);

    public int getLower() {
        return lower.get();
    }

    public void setLower(int i) {
        while (true) {
            int l = lower.get();
            int u = upper.get();
            if (i > u)
                throw new IllegalArgumentException();
            if (lower.compareAndSet(l, i))
                return;
        }
    }

    public int getUpper() {
        return upper.get();
    }

    public void setUpper(int i) {
        while (true) {
            int l = lower.get();
            int u = upper.get();
            if (i < l)
                throw new IllegalArgumentException();
            if (upper.compareAndSet(u, i))
                return;
        }
    }

    public boolean contains(int i) {
        return lower.get() <= i && i <= upper.get();
    }
}
```

Aufgaben:

- Zeigen Sie das Problem der obigen Implementierung.
- Implementieren Sie einen korrekten Lock-Free `NumberRange`. Zur Lösung muss eine `AtomicReference` verwendet werden. Überlegen Sie sich, welche Werte zusammengehören. Möglicherweise müssen Sie eine kleine Hilfsklasse implementieren.