



6. Übung zu Softwaretechnik 2

Formale Spezifikation mit Z (II)

1 CheckOut verfeinern

Verfeinern sie die Versionskontrolle aus der ersten Vorlesung über Z so, daß sie auf einer Folge (Sequenz) `checked_out` anstelle der Menge arbeitet. Beweisen Sie, daß diese Verfeinerung gültig ist. `permission` kann eine Menge bleiben.

2 CheckOut implementieren

Implementieren sie das totale CheckOut durch Verfeinerung. Folgen und Mengen sollen jeweils zu einem Array über einen entsprechenden Datentyp verfeinert werden. Begründen sie die Korrektheit der Verfeinerung.

3 Axiomatisches Beweisen mit Invarianten

In dieser Aufgabe soll zu der for-Schleife im Programm mit den gegebenen Vor- und Nachbedingungen eine Invariante aufgestellt werden.

Der Kurs einer Aktie an n aufeinanderfolgenden Tagen ist als Feld $kurs[0..n-1]$ von Integers gegeben. Gesucht sind $kauftag$ und $verkauftag$, so daß der Gewinn $bestprofit$ (Kursdifferenz zwischen Kauf- und Verkauftag) maximal wird.

Vorbedingung:

$$n > 0 \wedge \forall i \in \{0, \dots, n-1\} : kurs[i] > 0$$

Nachbedingung:

$$\begin{aligned} bestprofit &= kurs[verkauftag] - kurs[kauftag] \\ &= \max\{kurs[i] - kurs[j] \mid i, j \in \{0, \dots, n-1\} \wedge j < i\} \end{aligned}$$

Realisierung (in Java):

```
public static int kauftag;
public static int verkaufstag;
public static int best_profit(int kurs[], int n)
{
    int min = Integer.MAX_VALUE; // Mindestkurs
    int profit = 0; // Erzielter Gewinn
    int mintag = 0; // Tag mit minimalem Kurs

    kauftag = 0;
    verkaufstag = 0;

    for (int tag = 0; tag < n; tag++) {
        if (kurs[tag] < min) then {
```

```
        min = kurs[tag];
        mintag = tag;
    }

    if (kurs[tag] - min > bestprofit) {
        profit = kurs[tag] - min;
        kaufstag = mintag;
        verkaufstag = tag;
    }
}

return profit;
}
```

Fragen?

Bei Fragen wenden sie sich an Holger Cleve <cleve@cs.uni-sb.de>.