

Übung zur Vorlesung Programmierung

Aufgabe T6

Schreiben Sie ein Codesegment P , das eine Integer-Variable x auf folgende Art und Weise verändert:

- Wenn $x = 0$ gilt, dann bleibt x unverändert.
- Wenn $x < 0$ gilt, dann wird x dekrementiert.
- Wenn $x > 0$ gilt, dann wird x inkrementiert.

Verifizieren Sie anschließend mit Hilfe des Hoare-Kalküls, dass folgende Aussage gilt:

$\langle x = x' \rangle$

P

$\langle (x' = 0 \wedge x = x') \vee (x' < 0 \wedge x = x' - 1) \vee (x' > 0 \wedge x = x' + 1) \rangle$

Aufgabe T7

Implementieren Sie eine Klasse *Set*, deren Instanzen Mengen natürlicher Zahlen darstellen. Die Klasse *Set* soll die Methoden *add(int x)*, *contains(int x)* und *remove(int x)* zur Verfügung stellen. Dabei sollen die Attribute jeder Instanz der Klasse *Set* unveränderbar sein, nachdem sie im Konstruktor gesetzt wurden. Die Menge wird dafür durch eine Liste von *Set* Objekten repräsentiert. Jedes *Set* Objekt enthält eine Operation und eine Referenz zu einem anderen *Set* Objekt (oder *null*, wenn es das letzte Element der Liste ist). Jede Operation soll entweder eine Einfüge- oder eine Löschoption sein. Die Methoden *add(int x)* und *remove(int x)* geben ein neues *Set* Objekt zurück, welches der neue Kopf der Liste ist und eine Einfüge- oder Löschoption repräsentiert. Die Methode *contains(int x)* entscheidet, ob ein Element in der Menge ist oder nicht, indem sie die Einfüge- und Löschoptionen, die bereits ausgeführt wurden, durchläuft.

Aufgabe H6 (10 Punkte)

Implementieren Sie ein Codesegment P , das eine Integer-Variable x derart verändert, dass sie nach Ausführung von P den Betrag jener Zahl enthält, die vor Ausführung von P in x gespeichert war. Verifizieren Sie anschließend mithilfe des Hoare-Kalküls, dass folgende Aussage gilt:

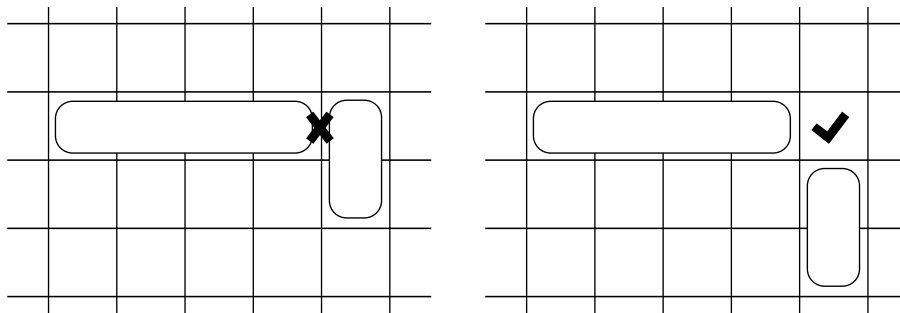
$$\langle x = x' \rangle P \langle x = |x'| \rangle$$

Aufgabe H7 (10 Punkte)¹

Auf der Homepage zur Vorlesung finden Sie eine Archivdatei `battleship.zip`. Darin befindet sich eine fast vollständige Implementierung des Spiels "Schiffe versenken": lediglich in der Klasse `lufti.battleship.Ship` fehlen sinnvolle Implementierung zweier Methoden. Zum Starten des Spiels kompilieren sie die Klasse `main.BattleshipMain` und starten diese (starten Sie das Spiel ruhig einmal, um die fehlende Funktionalität zu sehen).

Implementieren Sie folgenden Methoden der Klasse `Ship`, um das Spiel zu vervollständigen:

- `isHitBy(Point shot)`: Diese Methode soll `true` zurückgeben, wenn die Koordinate `shot` von diesem Schiff belegt wird. Es ist ratsam, dazu die Methode `getCoordinates()` zu Hilfe zu nehmen.
- `toCloseTo(Ship otherShip)`: Diese Methode soll `true` zurückgeben, wenn dieses Schiff und das übergebene Schiff `otherShip` zu nahe beieinander liegen. Dies ist dann der Fall, wenn nicht mindestens ein Feld zwischen beiden Schiffen liegt; allerdings dürfen sich die Schiffe "über Eck" berühren:



Für die schriftliche Abgabe reicht es aus, wenn Sie die beiden obigen Methoden angeben. Senden Sie Ihrem Tutor bitte lediglich die von Ihnen modifizierte Klasse `Ship`.

Abgabe zum 19.11.2013

¹Bitte Quelltext für die Abgabe ausdrucken und zusätzlich per E-mail an den jeweiligen Tutor senden.