

3. Übung zur Vorlesung Einführung in die Programmierung

A3-1 *Simpler Taschenrechner* Implementieren Sie zum Aufwärmen einen einfachen Taschenrechner! Es sollen solange Zahlen von der Tastatur eingelesen und aufaddiert werden, bis der Benutzer den String "end" eingetippt hat. *Beispiel:*

```
Dies ist ein Taschenrechner. (Eingabe "end" zum Beenden)
Zustand: 0 Gib bitte eine ganze Zahl ein: 77
Zustand: 77 Gib bitte eine ganze Zahl ein: -2
Zustand: 75 Gib bitte eine ganze Zahl ein: acht
Das war keine ganze Zahl! (Eingabe "end" zum Beenden)
Zustand: 75 Gib bitte eine ganze Zahl ein: -6
Zustand: 69 Gib bitte eine ganze Zahl ein: end
Endstand ist: 69
```

Dem Übungsblatt liegt ein unverbindliche Vorlage bei, um Ihnen etwas Tipparbeit zu ersparen. *Zusatzaufgabe um die Schwierigkeit zu erhöhen:* Wenn weder eine Zahl noch "end" eingegeben wurde, wird jedes Zeichen der Eingabe einzeln ausgegeben und 1 addiert:

```
Zustand: 0 Gibt bitte eine ganze Zahl ein: EiP
Gibt mir ein E!
Gibt mir ein i!
Gibt mir ein P!
Zustand: 3 Gibt bitte eine ganze Zahl ein:
```

A3-2 *Zustandshölle* Berechne mit Papier und Bleistift die Ausgabe dieses Programms:

<pre>public class Main { private static String output = "Start"; public static void main(String[] args) { int a = 1; MyInt b = new MyInt(); crazyCount(a,b); crazyCount(a,b); crazyCount(a,b); print("Ende"); System.out.println(output); } public static void crazyCount(int number, MyInt other){ number = number + 1; other.add(number); print(number + " + " + other.getMyint() + " = " + (other.getMyint() + 2)); } }</pre>	<pre>// Definition der Klasse Main geht hier weiter: public static void print(String message) { message= System.lineSeparator()+message; output = output + message; } } // Ende der Definition von Klasse Main</pre> <hr/> <pre>public class MyInt { private int myint = 0; public int getMyint() { return myint; } public void add(int myint){ this.myint = this.myint + myint; } } // Ende Klasse MyInt</pre>
---	--

H3-1 Zustandshölle (8 Punkte; Abgabe: H3-1.txt oder H3-1.pdf)

Fortsetzung von Aufgabe A3-2, das dort gegebene Programm wird wie folgt geändert:

- (i) Die Klasse `Main` wird um eine Klassenvariable des Typs `boolean` mit dem Namen `state` erweitert, welche mit `false` initialisiert wird.
- (ii) Der Rumpf der `main`-Methode wird am Ende erweitert um das Statement `main2("Anfang");`
- (iii) Die Klasse `Main` wird um folgende Methodendefinitionen erweitert:

```
public static void main2(String msg) {
    output = msg;
    boolean state = true;
    crazyBool(state);           // 1. Aufruf
    crazyBool(state);           // 2. Aufruf
    crazyBool(false);           // 3. Aufruf
    crazyBool(false);           // 4. Aufruf
    System.out.println(output); // Was wird hier ausgegeben?
}

public static void crazyBool(boolean argument) {
    boolean r = implies(state, argument);
    print(state + " -> " + argument + " = " + r);
    if (state != argument) {
        argument = !argument;
        state = !argument;
    }
}

public static boolean implies(boolean first, boolean second) {
    return !first || second;
}
```

Ihre Aufgabe (möglichst nur mit Papier & Bleistift lösen):

- a) Um die erste Änderung (i) zu implementieren, genügt es, eine einzige Zeile ganz am Anfang in die Deklaration der Klasse `main` einzufügen!
Geben Sie diese einzufügende Zeile an!
- b) Wenn alle Änderungen implementiert sind, sollte das Programm ohne Fehler ablaufen. Welche Ausgabe produziert das letzte Statement in Methode `main2` bei der Ausführung?
- c) Welche Art von Variable (Instanzvariable, Klassenvariable, Parameter, lokale Variable) sind `argument`, `state` und `r` innerhalb der Methode `crazyBool` jeweils?
- d) Erklären Sie, warum alle vier Aufrufe der Methode `crazyBool` unterschiedliche Ausgaben produzieren, obwohl jeweils die ersten beiden und die letzten beiden Aufrufe mit dem gleichen Argument (`state` und `false`) erfolgen!

Beantworten Sie dabei insbesondere: Hat das Argument `state` beim zweiten Aufruf einen anderen Wert als beim ersten Aufruf der Methode? Was ändert sich zwischen dem dritten und dem vierten Aufruf von `crazyBool`, und warum?

H3-2 Bankkonto mit PIN (0 Punkte; Alle .java-Dateien Ihrer Lösung abgeben)

In der Vorlesung am 26.10.17 wurde eine Klasse `Bankkonto` entwickelt, welche Sie von der Vorlesungshomepage herunterladen können. Enthalten sind 2 Dateien: `Bankkonto.java` und `Main.java`; letztere dient nur zum testen.

Ihre Aufgabe: Entwickeln Sie ein PIN-geschütztes Bankkonto! Schreiben Sie eine Klasse `BankkontoPIN`, welche bei Eröffnung eines Bankkontos zusätzlich eine PIN des Typs `String` benötigt.

- a) Die Klasse soll ansonsten die gleiche Funktionalität bieten wie `Bankkonto`, mit dem Unterschied, dass die Methoden `einzahlen` und `getKonto` einen zusätzlichen Parameter des Typs `String` anfordern, und nur dann ausgeführt werden, wenn die angegebene PIN mit der intern gespeicherten PIN übereinstimmt. Bei einem positiven Geldbetrag soll `einzahlen` jedoch auch mit einer falschen PIN ausgeführt werden!

Da wir Fehlermeldung noch nicht behandelt haben, soll bei einer falschen PIN `einzahlen` einfach keine Wirkung haben und `getKonto` immer 0 zurückliefern.

- b) Schreiben Sie eine neue Methode `neuePin`, welche das Auswechseln der PIN erlaubt – natürlich nur, wenn auch die alte PIN korrekt angegeben wurde. Diese Methode soll `true` zurückliefern, falls das Auswechseln der PIN erfolgreich war, und `false` sonst. Überlegen Sie sich zuerst den Kopf der geforderten Methodendefinition (“Signatur”).

Hinweis: Erweitern Sie auch die `main`-Methode in `Main` um weitere Kontotransaktionen zum Testen. Es geht in dieser Aufgabe um Methoden und If-Statements, d.h. Ihre neue Klasse `BankkontoPIN` soll selbst keine Bildschirmausgabe oder Tastatureingabe durchführen.

Wenn Sie möchten, können Sie jedoch Konto-Aktivitäten mit `System.out.println` protokollieren und Eingaben mit `Scanner` durchführen (siehe A2-1), damit Sie leichter sehen können, wie das Konto arbeitet.

Abgabe: Lösungen zu den Hausaufgaben können bis Sonntag, den 12.11.17, mit UniWorX nur als `.zip` abgegeben werden. Aufgrund des Klausurbonus müssen die Hausaufgaben von Ihnen alleine gelöst werden. Abschreiben bei den Hausaufgaben gilt als Betrug und kann zum Ausschluss von der Klausur zur Vorlesung führen. Bitte beachten Sie auch die Hinweise zum Übungsbetrieb auf der Vorlesungshomepage (www.tcs.ifi.lmu.de/lehre/ws-2017-18/eip/).