

Übungsblatt 2

Abgabe: 24.11.2011

Aufgabe 1 (20 %)

Gegeben ist die Methoden `aForMethod(int)` mit einer *for*-Schleife. Erstellen Sie eine weitere Methode `theDoWhileMethod()`, die bei gleicher Eingabe den gleichen Wert zurückgibt, jedoch eine *do-while*-Schleife verwendet.

```
public int aForMethod(int k) {
    int j = 1;
    for (int i=1; i<k; i++) {
        j *= i+1;
    }
    return j;
}
```

Aufgabe 2 (25 %)

Sei `expression` ein beliebiger boolescher Ausdruck und die Methoden `doIf()` und `doElse()` gegeben. Formulieren Sie ausschließlich mit *while*-Schleifen einen zu äquivalenten Ausdruck.

```
if (expression) {
    doIf();
} else {
    doElse();
}
```

Ist es im Allgemeinfall möglich eine Schleife mit `if-else`-Anweisungen darzustellen? Begründen Sie Ihre Behauptung.

Aufgabe 3 (20 %)

Setzen Sie folgende Verzweigung mit einem *switch*-Statement in der Methode `performActionSwitch(char key)` um und erläutern Sie die Funktionsweise von `break`;

```
public int performActionBranch(char key) {
    int result = 0;
    if (key == 'w' || key == 'W') {
        result = 1;
    } else if (key == 'a' || key == 'A') {
        result = 2;
    } else if (key == 'd' || key == 'D') {
        result = 3;
    } else if (key == 's' || key == 'S') {
        result = 4;
    } else if (key == 'q' || key == 'Q') {
        result = 5;
    } else if (key == 'e' || key == 'E') {
        result = 6;
    } else {
```

```

    result = 7;
  }
  return result;
}

```

Aufgabe 4 (15%)

Gegeben ist folgende Methode:

```

public int getWithDoWhile ( int b ) {
    int a = 0;

    do {
        if ( b > 0 ) {
            a = a + 3*b;
        } else {
            a = a - 2*b;
        }
    } while ( a < b );

    return a;
}

```

Gibt es eine Eingabe `b`, für die die `do-while`-Schleife nie verlassen wird? Implementieren Sie eine zweite Methode `getWithWhile`, die äquivalent zu `getWithDoWhile` ist, aber stattdessen eine `while`-Schleife benutzt. Die Methode `getWithWhile` darf genau eine `if-else`-Anweisung enthalten.

Aufgabe 5 (20%)

Analysieren Sie die Kontrollstrukturen der folgenden Funktion und nennen Sie eine Eingabenbelegung, für die diese `true` zurück gibt. Erläutern Sie Ihren Lösungsansatz.

```

public boolean crucialFunction001(int x, int count) {
    boolean result;
    int n = 0;
    int v = 0;
    if (count < 0 || count > 100) {
        result = false;
    } else {
        for (int i = x; i < count; i++) {
            n = n + 1;
        }
        while (n > 200) {
            v = v + 1;
            n = n - 1;
        }
        result = v > 0;
    }
    return result;
}

```

Hinweise: Zur Lösung der Aufgaben erstellen Sie eine Klasse `Uebung2` und fügen dieser die geforderten Methoden hinzu. Für die Tests Ihrer selbstgeschriebenen Methoden erstellen Sie bitte die Klasse `Uebung2Test`. Sie können sich dabei an der gegebenen Struktur der Tests aus dem ersten Übungsblatt orientieren.