

Übungsblatt 10

Abgabe bis Donnerstag, 19.01.12, 12:00 Uhr

Hinweis: Kompilieren und testen Sie Ihre Programme mit Hilfe des `ant`-Buildsystemes. Schicken Sie Ihrem Tutor eine Kopie des kompletten Projektordners, der alle benötigten Dateien enthält.

Aufgabe 10.1

WHILE-Programme können nur mit `while`-Schleifen, Zuweisungen (`=`) und den Operatoren `+`, `-` und `≠` beschrieben werden. Schreiben Sie folgende Methoden so um, dass Sie **nur** diese Anweisungen verwenden.

1.

```
void ifThenElse(int x) {
    if (x != 0) {
        a();
    } else {
        b();
    }
}
```
2.

```
boolean greaterEqualThan(int x) {
    return x >= 0;
}
```
3.

```
int modulo(int a, int b) {
    return a % b;
}
```

Sie können davon ausgehen, dass die Eingaben `a` und `b` positiv sind.

Aufgabe 10.2

Beweisen Sie per Induktion, dass folgender Algorithmus zur Berechnung der Fakultät der positiven ganzen Zahl n die Eigenschaft der totalen Korrektheit besitzt.

```
erg := 1;           // 1
i := 1;            // 2
while i <= n do    // 3
    erg := erg * i; // 4
    i := i + 1;    // 5
end                // 6
print erg;        // 7
```

Aufgabe 10.3

Entscheiden Sie, ob die folgenden Aussagen richtig oder falsch sind. Betrachten Sie dabei die Definition und Eigenschaften von Algorithmen entsprechend der Vorlesung.

	Richtig	Falsch
Jedes Problem, das mit einem Algorithmus gelöst werden kann, ist berechenbar.	<input type="checkbox"/>	<input type="checkbox"/>
Jedes Programm, das einen Algorithmus implementiert, ist korrekt.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt Java-Programme, für die entscheidbar ist, ob sie für beliebige Eingaben terminieren.	<input type="checkbox"/>	<input type="checkbox"/>
Ein Algorithmus belegt zu jedem Zeitpunkt nur endlich viele Ressourcen.	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt einen Algorithmus, der beliebige Programme auf Korrektheit überprüft.	<input type="checkbox"/>	<input type="checkbox"/>

Die folgende Methode soll die Eingabewerte a und b addieren.

```
public static int add(int a, int b) {  
    if(a % 2 == 0) {  
        return a + b;  
    } else {  
        while(true){ }  
        return 0;  
    }  
}
```

	Richtig	Falsch
Die Methode <code>add(int a, int b)</code> ist partiell korrekt.	<input type="checkbox"/>	<input type="checkbox"/>
Die Methode <code>add(int a, int b)</code> ist total korrekt.	<input type="checkbox"/>	<input type="checkbox"/>
Die Methode <code>add(int a, int b)</code> löst das Halteproblem.	<input type="checkbox"/>	<input type="checkbox"/>
Die Methode <code>add(int a, int b)</code> ist ein Algorithmus.	<input type="checkbox"/>	<input type="checkbox"/>