
Aufgaben zur Übergangsprüfung **Grundlagen der Programmierung** (Wiederholung) im
WS 96/97 (WI)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

Aufgabe 1:

Beweisen Sie durch Transformation, daß die folgende Formel ein Satz der Aussagenlogik ist. Begründen Sie die einzelnen Beweisschritte.

$$(\neg w \Rightarrow \neg u) \wedge \neg(u \oplus v) \Rightarrow (v \Rightarrow w)$$

Nutzen Sie diese Seite für die Kladde, die nächste Seite für die fertige Lösung.

$$(\neg w \Rightarrow \neg u) \wedge \neg(u \oplus v) \Rightarrow (v \Rightarrow w)$$

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

Aufgabe 2:

Analysieren Sie die folgenden Aussagen. Dabei ist die Grundmenge, über die Aussagen gemacht wird, die Menge aller Software, hier mit *Software* bezeichnet, diese ist nicht leer.

Es werden folgende einstellige Elementaraussagen verwendet:

$vonWW(sw)$

für ein Stück Software, das von der Firma *WinzigWeich* gebaut wurde

$preiswert(sw)$

für ein Stück preiswerter Software, nicht preiswerte Software ist teuer

$fehleranfällig(sw)$

für fehleranfällige Software, nicht fehleranfällige Software ist zuverlässig

Die Aussagen über Software als prädikatenlogische Formeln

1. $\exists sw \in Software \bullet (vonWW(sw) \wedge fehleranfällig(sw)) \Rightarrow preiswert(sw)$
2. $\exists sw \in Software \bullet vonWW(sw) \wedge (preiswert(sw) \wedge fehleranfällig(sw))$
3. $\exists sw \in Software \bullet (\neg vonWW(sw) \wedge preiswert(sw)) \Rightarrow \neg fehleranfällig(sw)$
4. $\exists sw \in Software \bullet preiswert(sw) \Rightarrow (vonWW(sw) \wedge fehleranfällig(sw))$
5. $\exists sw \in Software \bullet preiswert(sw) \Rightarrow \neg(vonWW(sw) \wedge \neg preiswert(sw))$
6. $\forall sw \in Software \bullet vonWW(sw) \Rightarrow preiswert(sw) \wedge fehleranfällig(sw)$
7. $\forall sw \in Software \bullet (\neg fehleranfällig(sw) \vee \neg preiswert(sw)) \Rightarrow \neg vonWW(sw)$
8. $\forall sw \in Software \bullet \neg vonWW(sw) \vee (preiswert(sw) \wedge \neg fehleranfällig(sw))$
9. $\forall sw \in Software \bullet fehleranfällig(sw) \Rightarrow (vonWW(sw) \wedge \neg fehleranfällig(sw))$
10. $\forall sw \in Software \bullet (\neg vonWW(sw) \vee \neg preiswert(sw)) \vee fehleranfällig(sw)$
11. $\forall sw \in Software \bullet preiswert(sw) \Rightarrow (vonWW(sw) \wedge fehleranfällig(sw))$
12. $\forall sw \in Software \bullet vonWW(sw) \Rightarrow (\neg preiswert(sw) \wedge fehleranfällig(sw))$

Geben sie für die folgenden Aussagen die Nummer(n) von **gleichwertigen** Formeln an, Mehrfachnennungen sind möglich, gibt es keine Formel tragen Sie 0 an die vorgesehene Stelle ein.

1. Es gibt WinzigWeich-Software, die ist preiswert und zuverlässig.
.....
 2. Software von WinzigWeich ist immer preiswert und zuverlässig.
.....
 3. Software von WinzigWeich ist weder preiswert noch zuverlässig.
.....
 4. Jede preiswerte Software ist von WinzigWeich und fehleranfällig.
.....
 5. Es gibt teure Software oder fehleranfällige WinzigWeich-Software.
.....
 6. Software von WinzigWeich ist immer preiswert und fehleranfällig.
.....
 7. Jede preiswerte Winzigweich-Software ist fehleranfällig.
.....
 8. Es gibt preiswerte, aber fehleranfällige WinzigWeich-Software.
.....
 9. Falsch.
.....
 10. Wahr.
.....
-

Aufgabe 3:

Gegeben seien die folgenden Variablen:

```
var f : array [0..n - 1] of Z;  
var b : B;  
var i : N0
```

Konstruieren Sie ein Programmstück, das folgendes Prädikat in der Variablen b berechnet.

$$\forall 0 < j < n \bullet f[j] \geq 0 \Leftrightarrow f[j - 1] < 0$$

Das Programmstück soll systematisch mit den Techniken aus der Vorlesung aus dem \forall Quantor abgeleitet werden, es soll so arbeiten, daß keine überflüssigen Berechnungen gemacht werden, nachdem das Resultat feststeht.

.....

.....

.....

.....

.....

.....

.....

.....

.....