



Beispiel-Klausuraufgaben
zur Vorlesung **Programmiersprachen 1**
im Wintersemester 2014/2015

Matrikelnummer:

Name:

Diese Beispielklausur besteht aus einer Textaufgabe und zwei Multiple-Choice-Fragen. Jede Textaufgabe wird mit 4 Punkten bewertet; jede Multiple-Choice Frage wird mit einem Punkt bewertet.

Jede der zwei Fragen hat genau eine richtige Antwort. Für jede Frage können Sie keine Antwort, eine Antwort, oder mehrere Antworten geben indem sie die entsprechenden Felder ankreuzen. Das bedeutet:

- Wenn Sie die Antwort sicher wissen, kreuzen sie nur diese an.
- Wenn Sie sich nicht sicher sind aber einige Antworten ausschliessen können, kreuzen sie die Antworten an, die korrekt sein könnten.
- Wenn Sie überhaupt nicht wissen, was die Antwort sein könnte und nichts ausschliessen können, kreuzen Sie nichts an.

Die Antworten werden so bewertet, dass zufälliges Raten null Punkte ergibt, also eine falsche Antwort Minuspunkte gibt.

Wir benutzen die Konvention dass ☒ “Antwort ausgewählt” bedeutet, und sowohl ☐ als auch ☐ bedeuten “Antwort *nicht* gewählt”.

Textaufgabe A

Vervollständigen Sie den `Letcc` Fall im Interpreter aus der Vorlesung über first-class continuations:

```
def eval(e: Exp, env: Env, k: Value => Nothing) : Nothing = e match {  
  ...  
  case Letcc(param,body) =>
```

Frage 1

Was ist ein tail call?

Ein tail call ist ...

- ☐ *a* ... der letzte Funktionsaufruf im Quellcode eines Programmes.
- ☐ *b* ... der möglicherweise zuletzt ausgeführte Funktionsaufruf in einem Programm zur Laufzeit.
- ☐ *c* ... der letzte Funktionsaufruf in Quellcode einer Funktion.
- ☐ *d* ... der möglicherweise zuletzt ausgeführte Funktionsaufruf in einer Funktion zur Laufzeit.
- ☐ *e* ... der Aufruf einer Continuation.
- ☐ *f* ... der Aufruf einer Funktion, die eine Continuation als zusätzlichen Parameter akzeptiert.

Frage 2

Was könnte das Ergebnis des Aufrufs `eval(NewBox(42), Map(), Map())` im Interpreter für BFAE aus der Vorlesung sein?

Das Ergebnis könnte ...

- ☐ *a* ... `(BoxV(42), Map())` sein.
- ☐ *b* ... `(BoxV(1), Map(1 -> IntV(42)))` sein.
- ☐ *c* ... `(BoxV('x'), Map('x -> IntV(42)))` sein.
- ☐ *d* ... `(BoxV(1), Map(1 -> 42))` sein.
- ☐ *e* ... `(BoxV('x'), Map('x -> 42))` sein.