

Nachbau des 1. PP-Test aus 2023

Aufgabe 1 (10 Punkte)

Welche Aussagen treffen auf die von uns betrachtete Variante des λ -Kalküls zu?

	JA	NEIN
Zu jedem λ -Ausdruck existiert ein dazu äquivalenter λ -Ausdruck in Normalform		
Ein λ -Ausdruck ist in Normalform, wenn darauf keine α - oder β -Konversion anwendbar ist.		
Alle Normalformen zu einem λ -Ausdruck sind bis auf α -Konversion gleich.		
Jedes Vorkommen von λ in einem λ -Ausdruck kennzeichnet einen rekursiven Funktionsaufruf		
$[e/v]f$ steht für: „e ersetzt alle freien Vorkommen von v in f“		
λ -Abstraktion ist ein anderer Name für eine Funktion im λ -Kalkül		
Reduktionsreihenfolgen aus einem λ -Ausdruck können unendlich lang sein		
Sind β - und η -Reduktion am gleichen Ausdruck anwendbar, haben sie das gleiche Ergebnis		
Konversionsregeln sind die gerichtete Variante der Reduktionsregeln		
Der λ -Kalkül ist gleich mächtig wie die Turing-Maschine		

Aufgabe 2 (3 Punkte)

Links stehen Regeln des λ -Kalküls, oben Namen solcher Regeln. Bitte Markieren Sie in jeder Zeile das Auswahlfeld, für das die links stehende Regel den darüber stehenden Namen hat.

	α -Konversion	β -Konversion	η -Konversion
$(\lambda v.f)e \equiv [e/v]f$			
$\lambda u.e \equiv \lambda v.[v/u] (e \notin fv(e))$			
$\lambda v.(e v) \equiv e \quad (e \notin fv(e))$			

Aufgabe 3 (7 Punkte)

Bitte markieren Sie jedes Auswahlfeld, bei dem die links stehende Eigenschaft auf die darüber stehende Modularisierungseinheit zutrifft

	Objekt	Klasse	Modul	Komponente	Namensraum
Kann erst zur Laufzeit erzeugt werden					
Erlaubt die Verwaltung globaler Namen					
Ist statisch					
Erlaubt zyklische Abhängigkeiten					
Fungiert als Übersetzungseinheit					
Wird durch „Deployment“ eingebunden					
Bildet eine Vorlage für die Objekterzeugung					

Aufgabe 4 (30 Punkte)

Bitte markieren Sie in jeder Zeile das Auswahlfeld, bei dem die links stehenden Eigenschaften für das darüber stehende Paradigma am besten zutrifft.

	Prozedural	Objektorientiert	Funktional	Parallel	Nebenläufig
Gut für Programmierung im Groben					
Gut für hardwarenahe Programme					
Gut für komplexe Algorithmen					
Gut für große Projekte					
Programmierung einfach erlernbar					
Ziel ist gute Kontrolle					
Ziel ist Programmiereffizienz					
Ziel ist langfristige Wartung					
Ziel ist kurze Programmlaufzeit					
Bewältigt viele Handlungsstränge					
Verspricht referentielle Transparenz					
Interprozesskommunikation nötig					
Gleichheit / Identität unterschieden					
Synchronisation (z.B. Monitor) nötig					
Rekursion statt Schleifen					
Programmfluss entspricht Kontrollfluss					
Nur nominale Abstraktion					
Unterstützt algebraische Datentypen					
Statisch typisiert mit Typinferenz					
Verlangt hohen Hardwareeinsatz					
Datenbereiche unabhängig bearbeitbar					
Verschiedenartige Daten als Einheit					
Daten global, nur ein Zugriffsweg					
Abstraktes Verständnis erzwungen					
Viele Möglichkeiten zur Faktorisierung					
Stark, aber nicht statisch typisiert					
Prozeduren/Objekte nicht als Daten					
Aliase stören nicht					
Effiziente Reaktion auf Ereignisse					
Die meisten Programme sind so					

Aufgabe 5 (6Punkte)

Bitte markieren Sie in jeder Zeile das Auswahlfeld, bei dem der links beschriebene Typ bei strukturellen Untertypbeziehungen die darüber stehende Varianz hat.

	Kovariant	Kontravariant	Invariant
Typ eines Eingangsparameter			
Typ eines Durchgangsparameters			
Typ eines Ausgangsparameters			
Typ des Rückgabewertes einer Methode			
Typ einer Objektvariable			
Typ einer Konstante			

Jeder ist eingeladen diesen Test zu erweitern oder zu verbessern. Ich mache das hier gerne für meine Kollegen im Studium. Wenn jemand die Arbeit schätzen möchte, kann er mich ja auf ein Bier einladen.

Aufgabe 6 (6 Punkte)

Bitte markieren Sie in jeder Zeile das Auswahlfeld, das angibt, ob die links stehende Art von Zusicherung im Untertyp auch stärker oder schwächer (wenn nicht gleich) oder nur gleich sein kann wie im Obertyp.

	Stärker	Schwächer	Nur gleich
Nachbedingung, die eine Zustandsänderung beschreibt			
Nachbedingung, die einen Ergebniswert beschreibt			
Vorbedingung, die sich auf den Objektzustand bezieht			
Vorbedingung, die einen Parameterwert beschreibt			
Invariante wenn Objektvariablen nicht von außen geschrieben werden			
Invariante wenn Objektvariablen von außen geschrieben werden können			

Aufgabe 7 (20 Punkte)

Folgende Klasse ist gegeben (Variablen public, um Arten möglicher Zusicherung nicht einzuschränken):

```
public class Stack{
    public int top = 0, count = 0, elems[] = new int[10];
    public void push(int i) {count++; elems[top++] = i;}
    public int pull() {count++; return elems[--top];}
}
```

Bitte markieren Sie in jeder Zeile das Auswahlfeld, bei dem der links stehende Text eine Zusicherung der oben stehenden Art bezogen auf diese Klasse sein kann (jede Zeile für sich).

	Vorbedingung	Nachbedingung	Invariante	Serverkontrollerter History-Constraint	Clientkontrollierter History-Constraint
Stack holds at most 10 elements					
Push adds an element to the stack					
Pull must be called after 10 calls of push					
Pull return the top element					
Push throws exception when stack is full					
$i > 0$					
Count can only increase					
Elems.length is always 10					
$Top \leq count$					
The result of pull is larger than 0					
Every method changes the value of top					
Pull removes the top element					
No method assigns a new value to elem					
Push must be called first in a new stack					
$Count > 0$ if $top > 0$					
$i > 0$ if $count > 0$					
$Elems[top-1] == i$ after executing $push(i)$					
Pull can be called only if $top > 0$					
Push increases count					
Count is increased when top is changed					

Jeder ist eingeladen diesen Test zu erweitern oder zu verbessern. Ich mache das hier gerne für meine Kollegen im Studium. Wenn jemand die Arbeit schätzen möchte, kann er mich ja auf ein Bier einladen.

Aufgabe 8 (18 Punkte)

Jede Referenzvariable in Java hat gleichzeitig drei (möglicherweise voneinander verschiedene Typen). Bitte benennen Sie sie:

Nennen Sie drei Sprachkonzepte oder Techniken zur Parametrisierung statischer Modularisierungseinheiten:

Nennen Sie bitte drei Sprachkonzepte oder Techniken, mit denen Werte in Objektvariablen neuer Objekte gebracht werden können:

Nennen Sie bitte drei Arten von Liveness-Problemen, die bei der Synchronisation auftreten können:

Nennen Sie bitte drei Sprachkonzepte oder Techniken für die Interprozesskommunikation:

Nennen Sie bitte drei Arten von Beziehungen zwischen abstrakten Datentypen bzw. Klassen: