



Halle, 04.. Juli 2006

## Programmiersprachen (SS 2006)

### Übungsserie 14

#### Aufgabe 1 (Vererbung in Eiffel)

In Eiffel gilt kovariante Vererbung.

- Welche Einschränkung gilt für die Redefinition eines Resultatstypen einer geerbten Funktion (ein Rückgabewert)?
- Wie darf die Klasseninvariante verändert werden?
- Welche Aussage über die Typsicherheit können Sie treffen?

#### Aufgabe 2 (Entwurf durch Vertrag)

- Welche Rechte und Pflichten haben die Funktion *fac* und der Benutzer dieser Funktion?

```
fac(n:INTEGER):INTEGER is
do
  if n=0 then Result:=1
  else Result:=n*fac(n-1)
  end
end
```

- Eiffel erlaubt Vertragsprüfung zur Laufzeit.  
Gegeben sei die Spezifikation eines Stacks in Eiffel:

```
class Stack
export
  number_of_elements, empty, full, push, pop, top
feature
  push(x:T) is
    require -- Vorbedingung
    ...    -- Angabe der Vorbedingung
do
```

```

        number_of_elements:=number_of_elements+1;
    ensure -- Nachbedingung
    ...    -- Angabe der Nachbedingung
end
...
    invariant -- Invariante
    ...     -- Angabe der Invariante
end --class Stack

```

Geben Sie Vor- und Nachbedingung(en) sowie die Invariante an.

### Aufgabe 3 (Typsystem)

ML war die erste bedeutende Programmiersprache, die ein polymorphes Typsystem, eine sogenannte *parametrische Polymorphie*, unterstützt.

Bestimmen Sie den Typ folgenden ML-Funktionen. Nehmen Sie an, `not` habe den Typ *Truth-Value*->*Truth-Value*, und die Komposition  $\circ$  den Typ  $(\beta \rightarrow \gamma) \times (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)$ .

- a. `fun negation(p) = not o p`
- b. `fun cond(b, f, g) =  
fn x => if b(x) then f(x) else g(x)`

### Aufgabe 4 (virtuelle Funktionen in C++)

Was sind die Werte der Ausdrücke `d.testF()` und `d.testG()`?

```

class B {
public:
    virtual char f()    { return 'B';}
    char g()           { return 'B';}
    char testF() { return f();}
    char testG() { return g();}
};

class D : public B {
public:
    char f()           { return 'D';}
    char g()           { return 'D';}
};

main(){
    D d;
    print d.testF(),d.testG();
}

```

**Aufgabe 5** (LISP-Ausdrücke)

Wenden Sie die folgende Funktion FA auf den S-Ausdruck ((SÄGE BOHRER) DREH-BANK) an:

```
(DEF FA
  (LAMBDA (X)
    (COND ((ATOM X) X)(T (FA (CAR X)))
    )
  )
)
```

Was macht die Funktion?