Programmiersprachenkonzepte

Linda

ein Vortrag
von
Alexander Lehning

Was ist Linda?

- Eine Programmbibliothek zur Erweiterung sequentieler Programmiersprachen
- ermöglicht nebenläufigen Zugriff und Ausführung von Prozessen
- ausserdem ein sehr einfaches Konzept
  - 1 Konstrukt
  - 4 + 2 Befehle/Operationen

Was ist der Tuple-Space?

- ein reiner Datenspeicher
- eine Multimenge/Bag
- sie enthält n-Tupel,

  n aus N\{0}

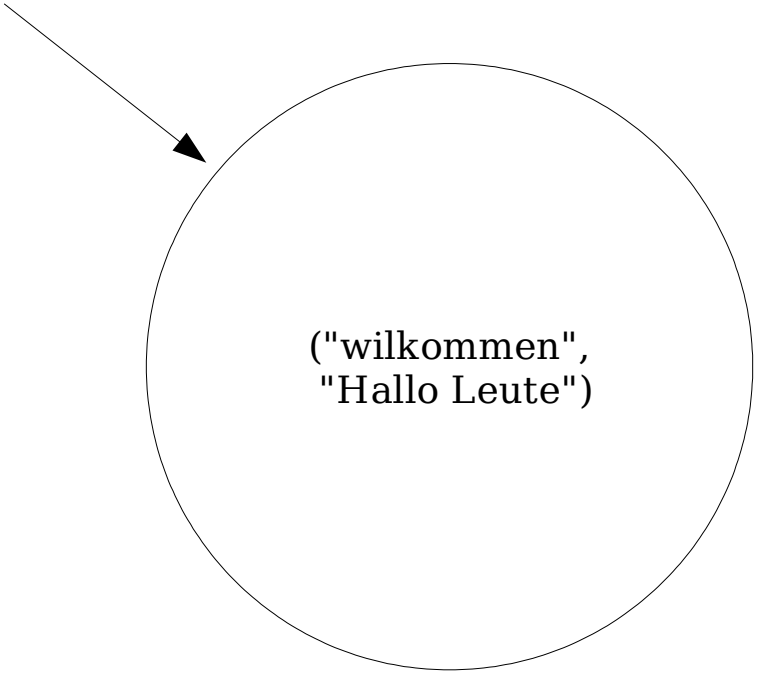
  – Tupel dürfen
    - unterschiedlich gross sein
    - verschiedene Datentypen enthalten

zur Erinnerung:
Multimenge ist eine Menge, die Elemente mehrfach enthalten kann.

Operation I: out

out(s) - erzeugt s im Tuple-Space

Beispiel: `out("wilkommen", "Hallo Leute");`



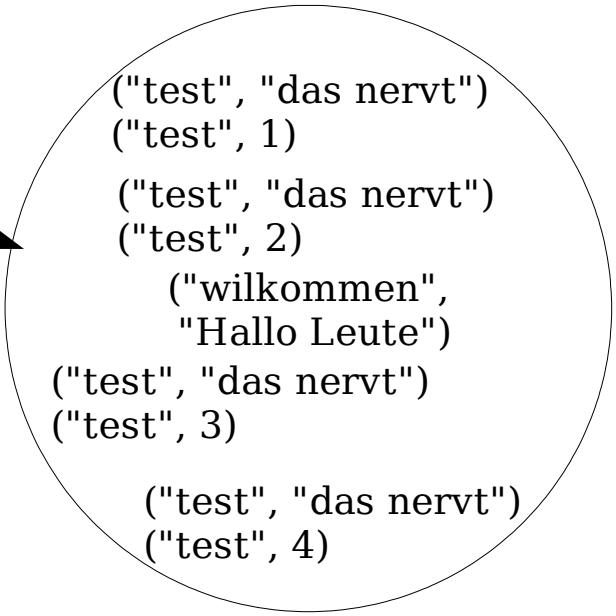
Tuple-Space

Operation I: out

out(s) - erzeugt s im Tuple-Space

Beispiel: `out("wilkommen", "Hallo Leute");`

```
for(i=1;i<5;i++){
   out("test", "das nervt");
   out("test", i);
   }
```



Tuple-Space

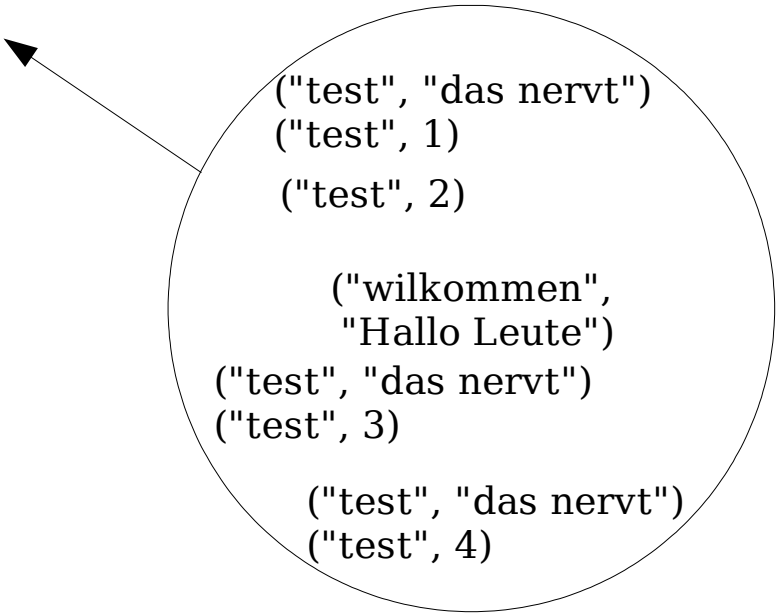
Operation II: in

in(s), entnimmt dem Tupelspace s

Beispiel:

```
in("test", ?text:String);

print(text); => das nervt
```



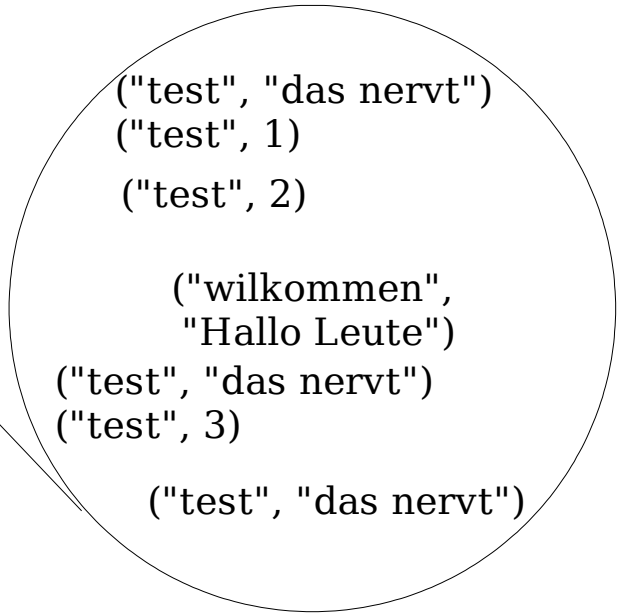
Tuple-Space

Operation II: in

in(s), entnimmt dem Tupelspace s

Beispiel:

```
in("test", ?text:String);
in("test", ?zahl:int);
```



Tuple-Space

Operation III: read

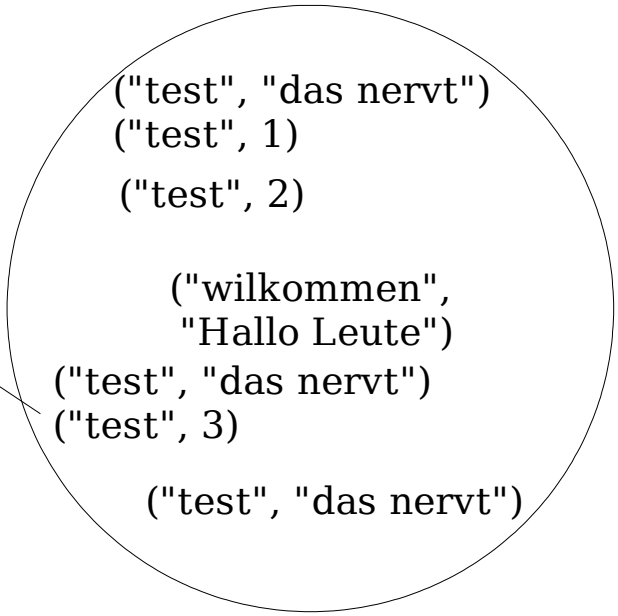
rd(s), dasselbe wie in(s) nur ohne entnahme

Beispiel:

```
rd("test", zahl2:int);

erg = zahl + zahl2;

==> erg = 7
```



Tuple-Space

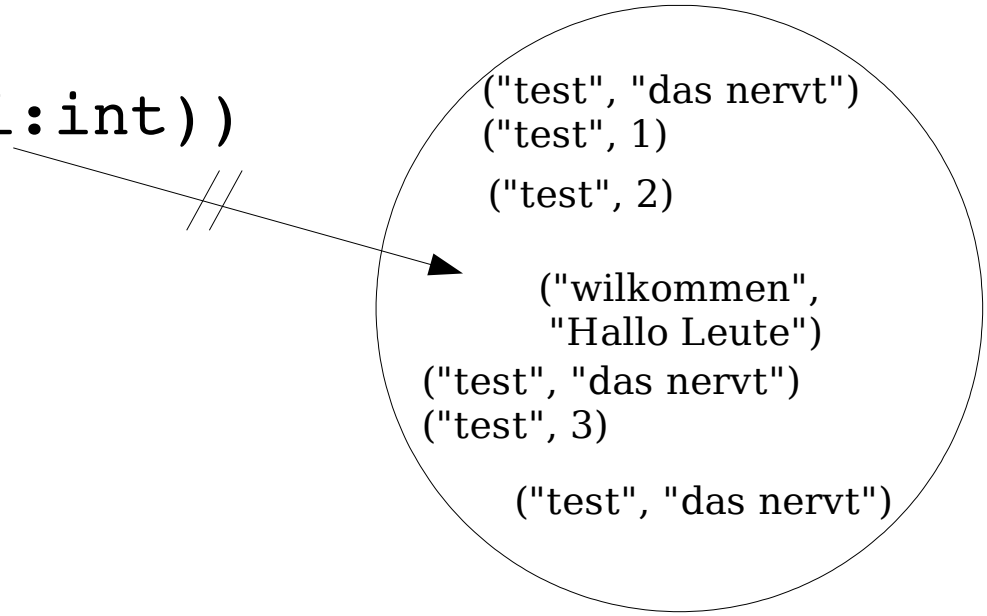
Operation IV + V: inp, rdp

inp(s) und rdp(s), Prädikate von in(s) und rd(s)

Beispiel:

```
if(rdp("wilkommen",i:int))

{restart}
```



Tuple-Space

Operation IV + V: inp, rdp

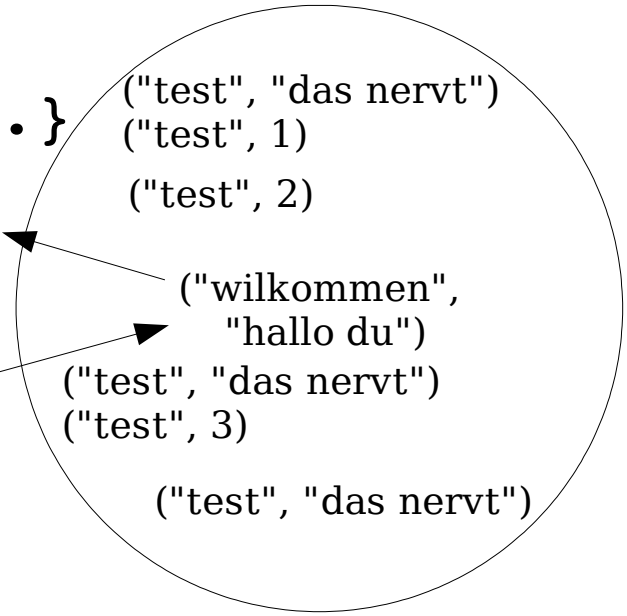
inp(s) und rdp(s), Prädikate von in(s) und rd(s)

Beispiel:

```
if(rdp("wilkommen",i:int)){...}

if(inp("wilkommen",s:String))

{out("wilkommen","hallo du")}
```



Tuple-Space

Operation IV + V: inp, rdp

inp(s) und rdp(s), Prädikate von in(s) und rd(s)

Beispiel:

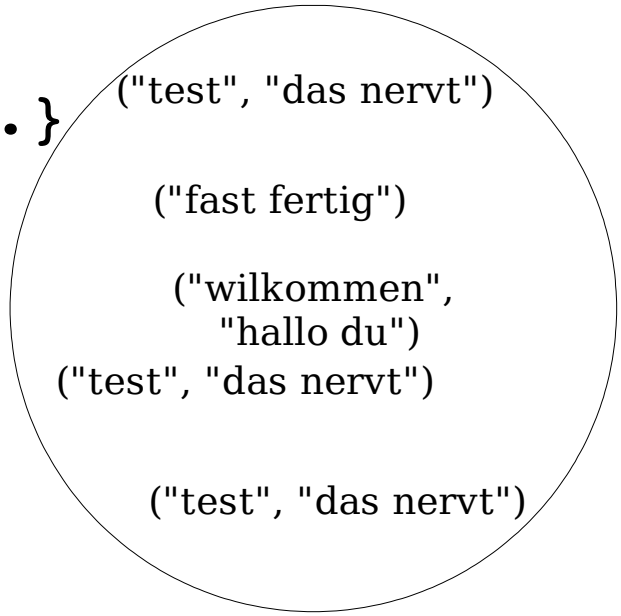
```
if(rdp("wilkommen",i:int)){...}

if(inp("wilkommen",s:String))

{out("wilkommen","hallo du")}

while(inp("test",i:int)){}

out("fast fertig")
```



Tuple-Space

Operation VI: eval

eval(s), entspricht einem out, nur dass vorher ein Wert berechnet wird

Beispiel:

```
eval("Wurzel",9,sqrt(9))
```

Tuple-Space

Merkwürdigkeiten

Doch was wenn:

```
out("zahl", 42, "toll");
x=rdp("zahl",i:int, "toll");
in("zahl",42,was:String);
```

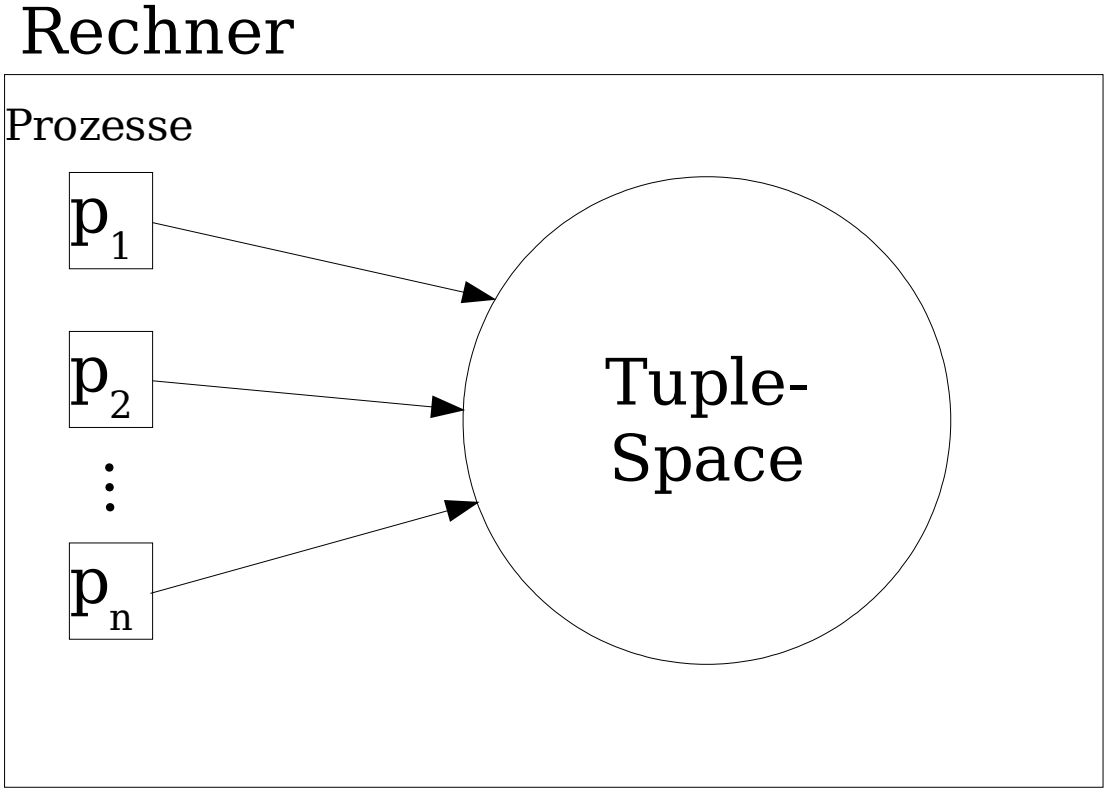
?

Mal wieder die Philosophen

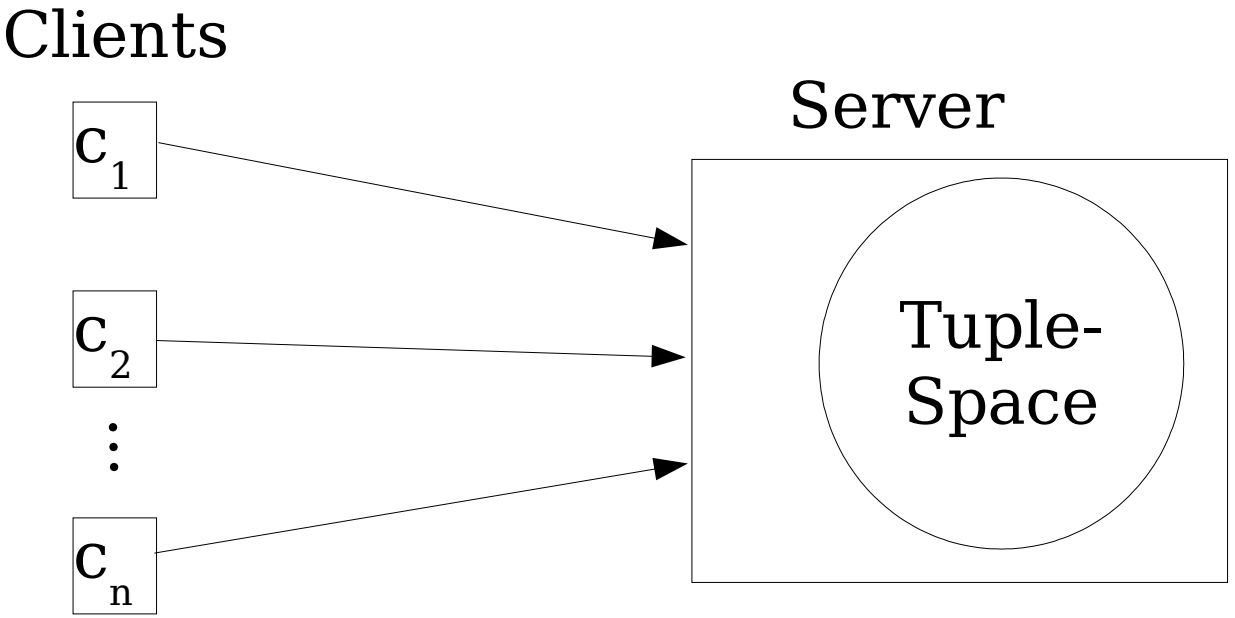
```
phil(i)
    int i;
    {while(1){think();
        in("table ticket");
        in("chopstick",i);
        in("chopstick",(i+1)%5); // % = modulo
        eat();
        out("chopstick",i);
        out("chopstick",(i+1)%5);
        out("table ticket");}}

initialize()
{   int i;
    for(i=0;i<5:i++){
        out("chopstick",i);
        if(i<4)
            out("table ticket");
        eval(phil(i));}}
```

Einsatzmöglichkeit von Linda I



Einsatzmöglichkeit von Linda II



Einsatzmöglichkeit von Linda III