

VIPR

Seminar „Visuelle Sprachen“

Luis Javier Antunez Congil
ljac@tzi.de

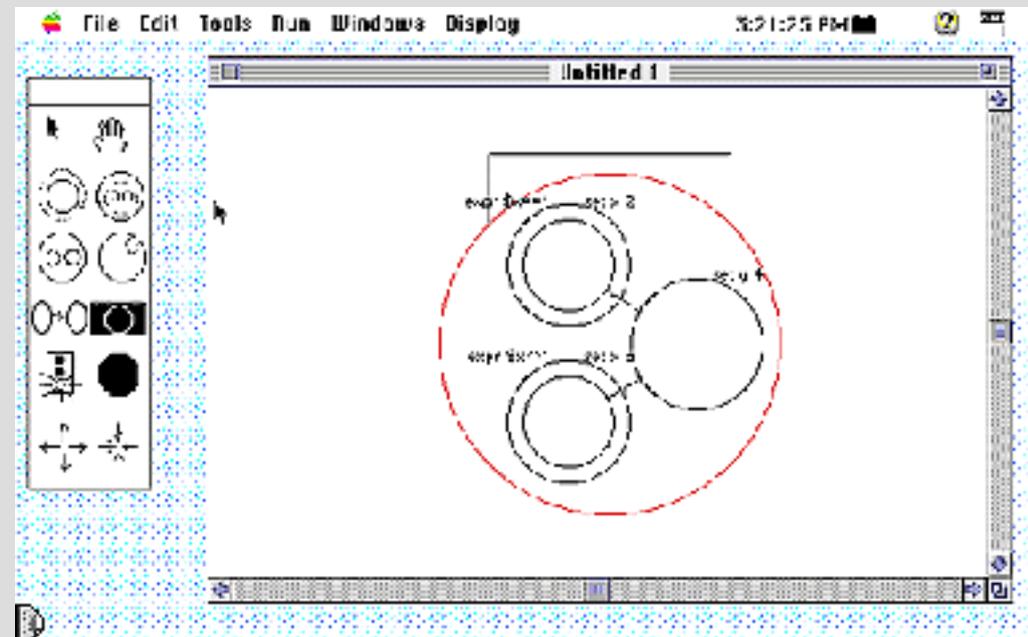
Übersicht

- Was ist VIPR?
- Grundelemente in VIPR
- Kontrollstrukturen
- Prozeduren
- Objektorientierung
- Bewertung & Zusammenfassung
- Quellen

Was ist VIPR?

Visual Imperative Programming (Language)
(University of Colorado) Anfang der 90er

- Semantik ähnlich c++
- Entwickler:
 - Wayne Citrin
 - Michael Doherty
 - Benjamin Zorn

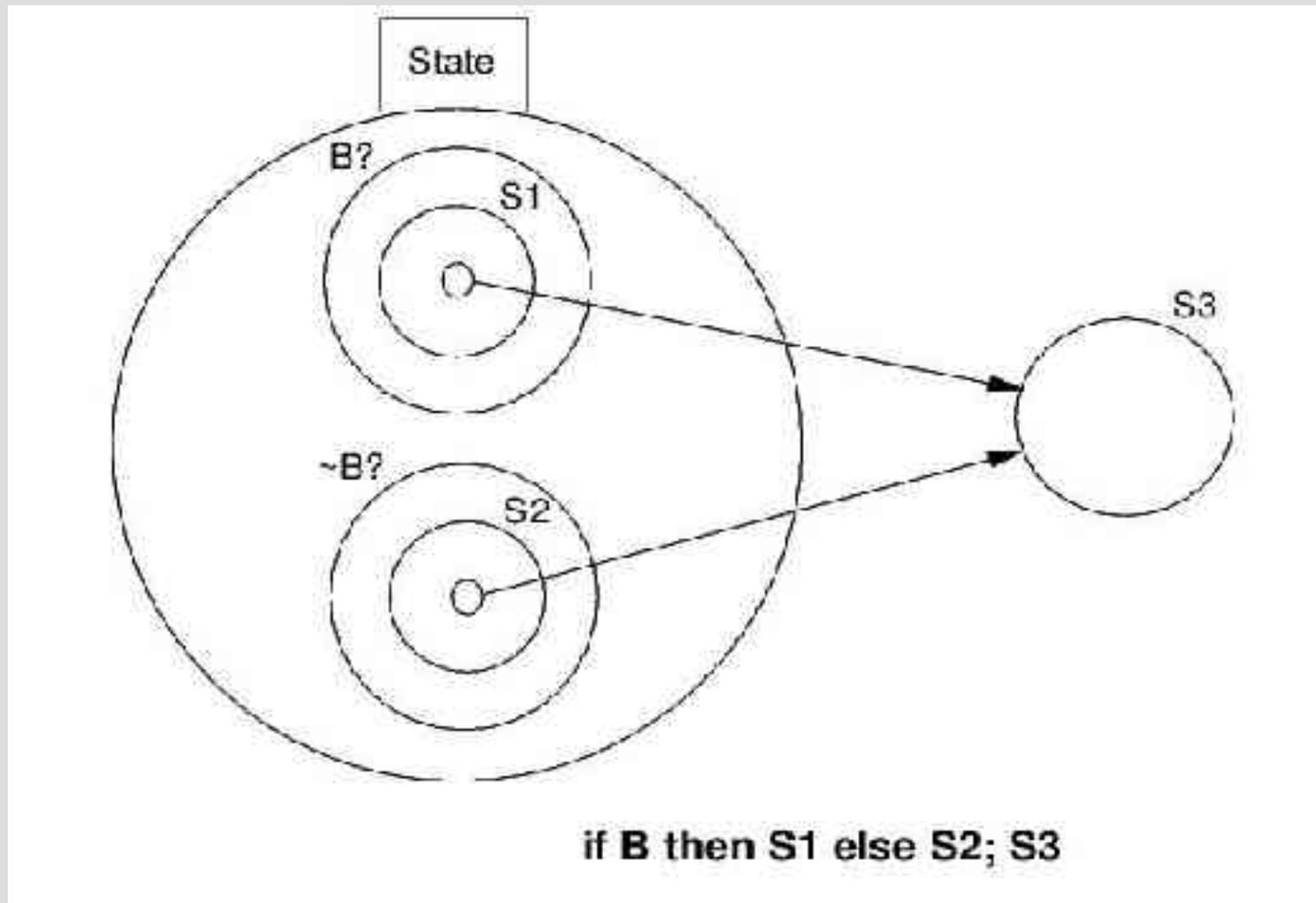


Was ist VIPR?

- Motivation:
 - Entwerfen eine objektorientierte Sprache
 - Einfach zu lernen und zu benutzen
- Keine Icons
- Keine Formulare
- Grundelemente:
 - Kreise
 - Pfeile
 - Text

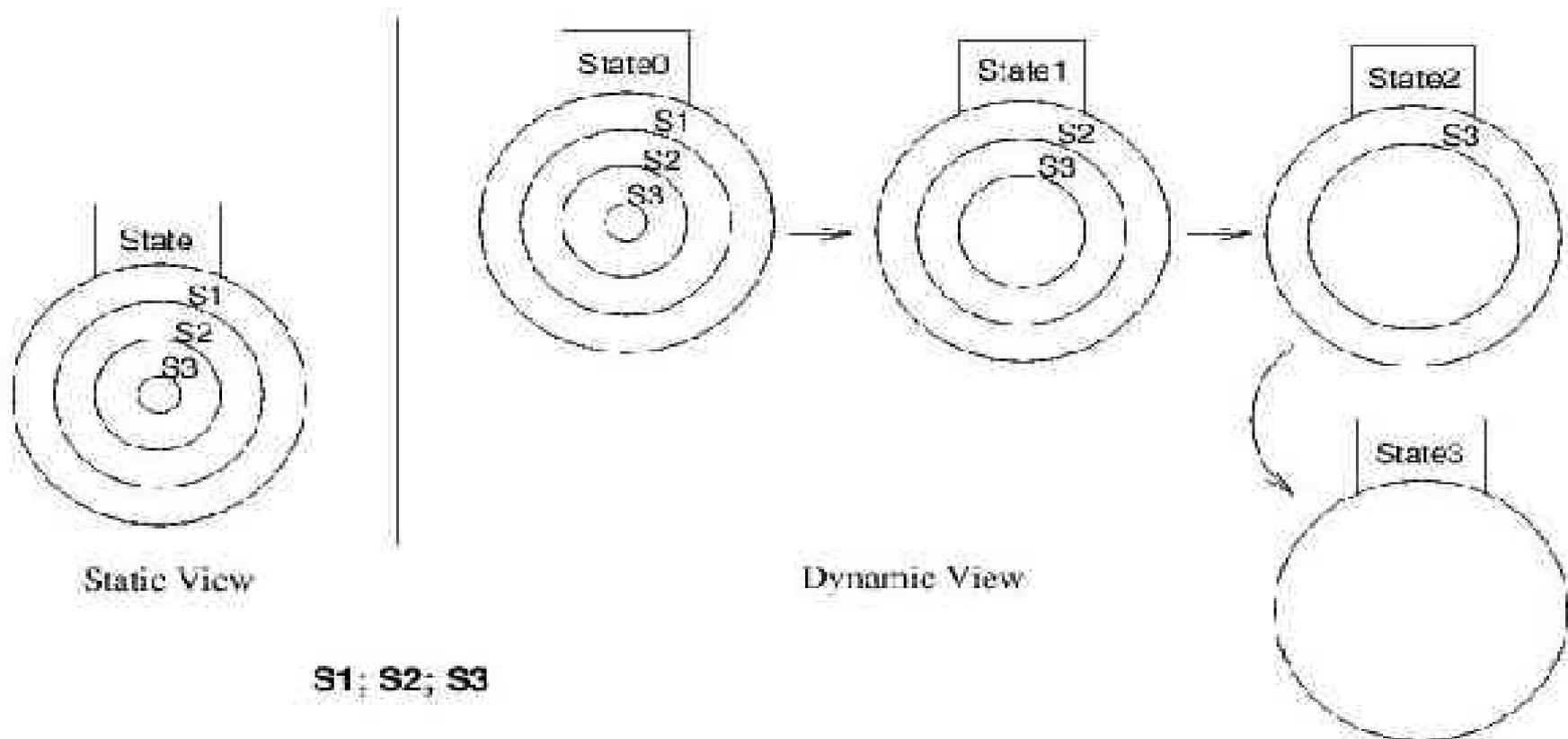
Was ist VIPR?

- Wie sieht es aus?



Was ist VIPR?

- Wie schreibt man ein Programm in VIPR?
- Wie sieht ein Programm in Ausführung aus?

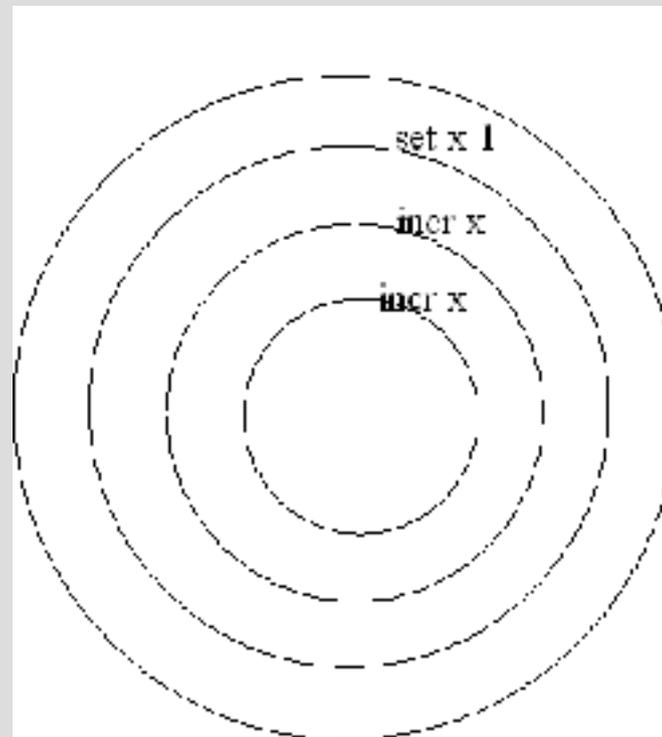


Kontrollstrukturen

- ◆ Sequentiell
- ◆ Bedingte Ausführung
- ◆ Schleifen

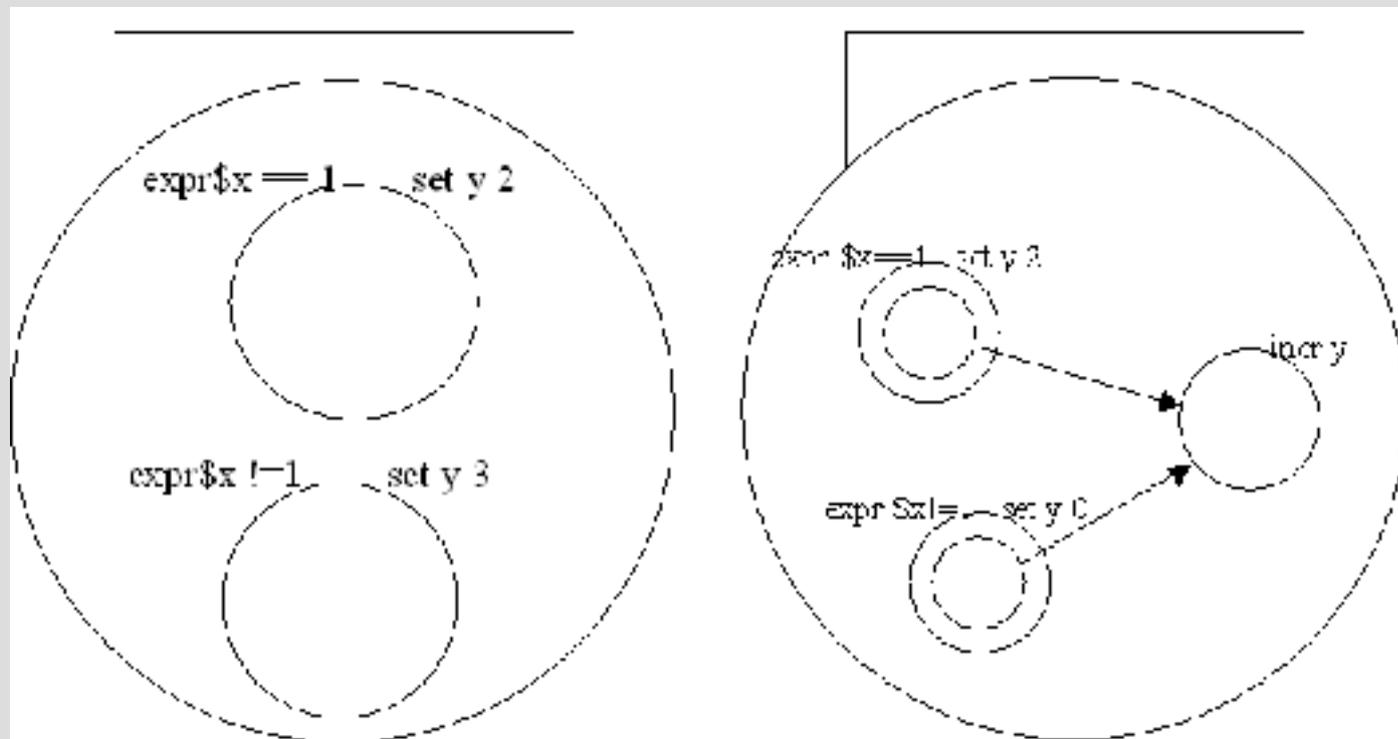
Kontrollstrukturen

- Sequentiell
 - vom äußeren Ring bis hin zum innersten Ring



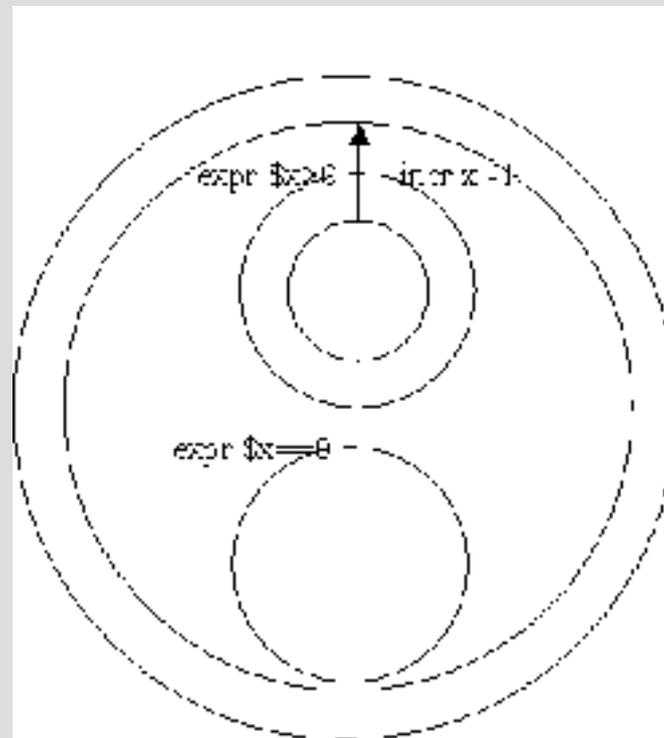
Kontrollstrukturen

- Bedingte Ausführung
 - Ringe in der gleichen Ebene
 - Keine spezielle Reihenfolge



Kontrollstrukturen

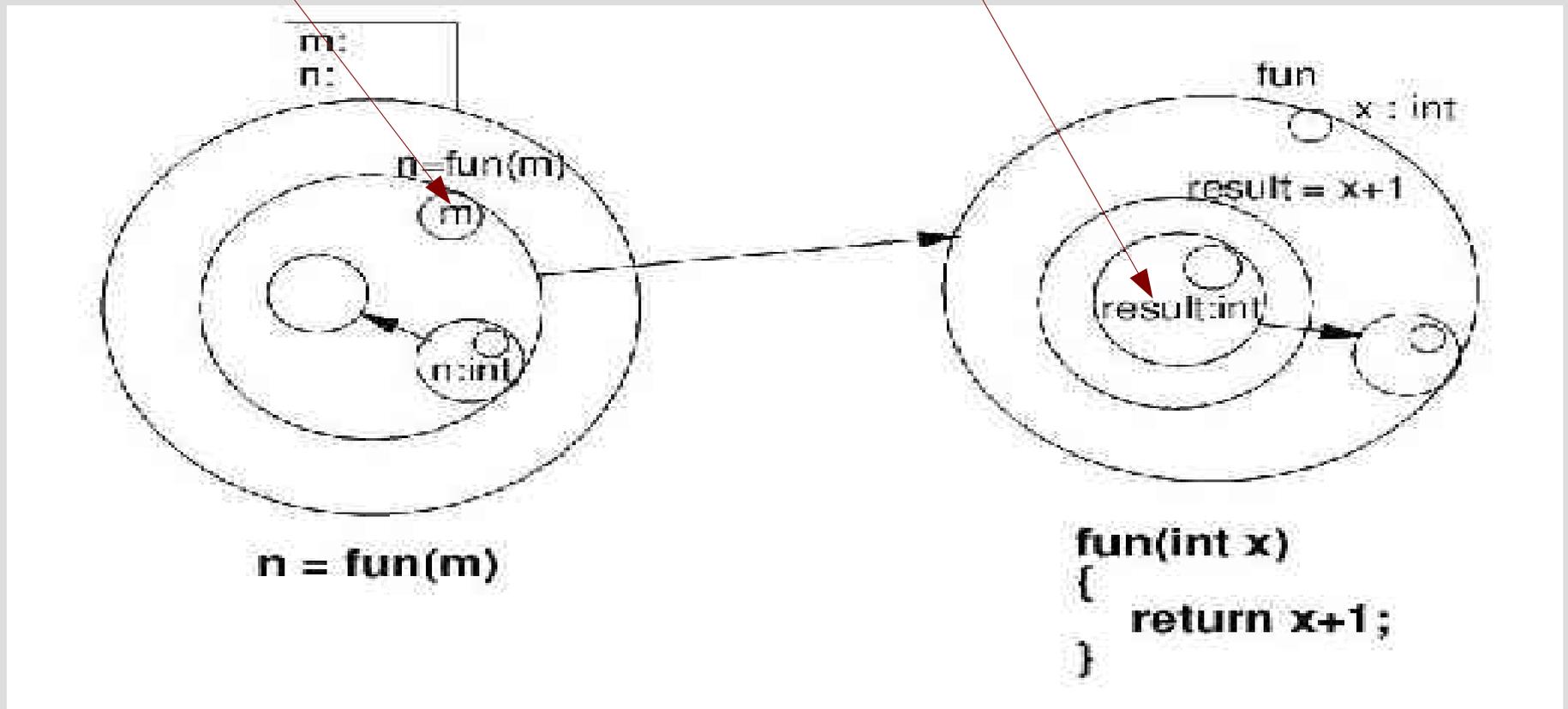
- Schleifen
 - Verwendung von Substitutionspfeilen
 - Rücksprung auf einen äußeren Ring



Prozeduren

Eingabeparameter

Rückgabewert

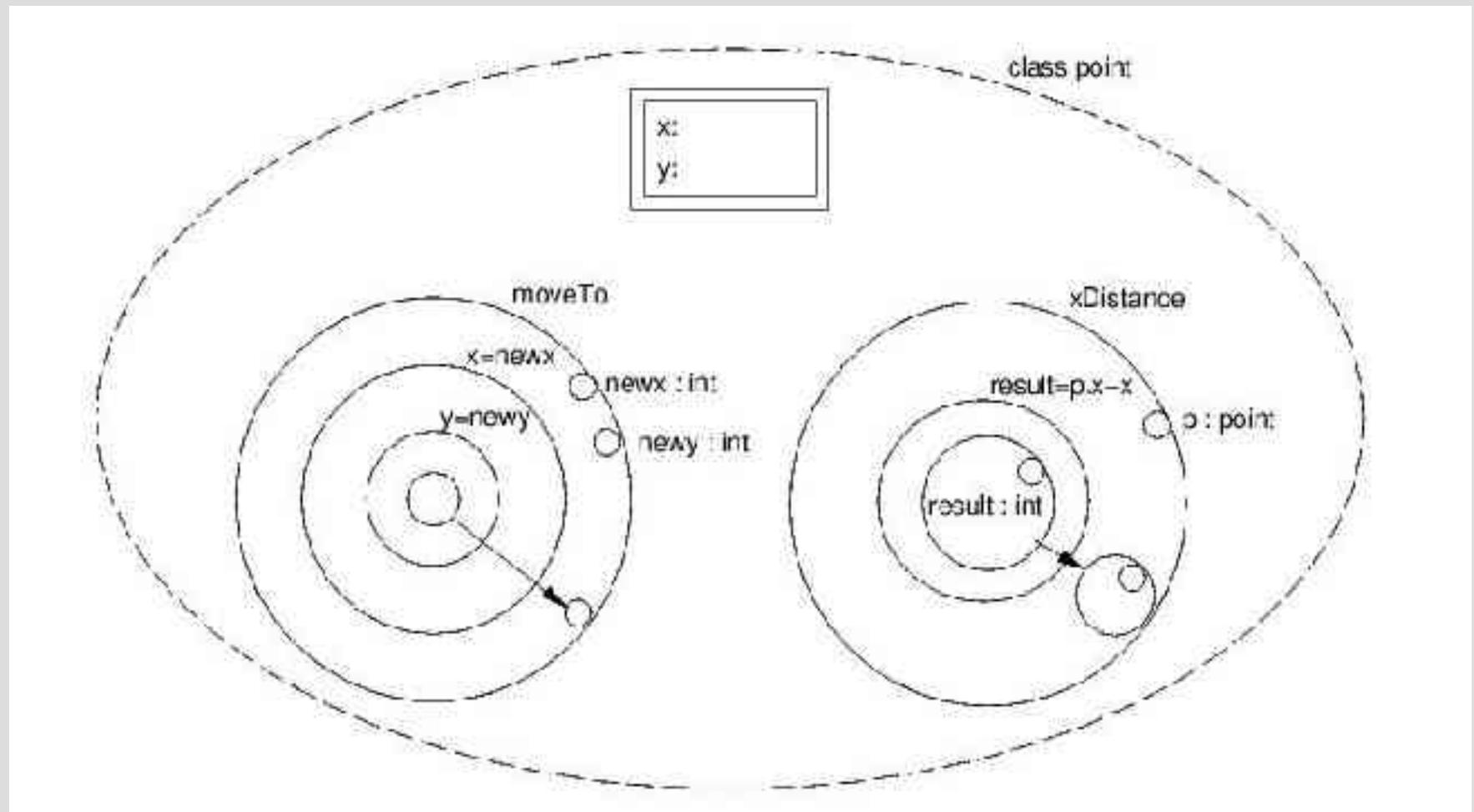


Objektorientierung

- Objekte und Klassen
- Vererbung, Polymorphismus

Objektorientierung

- Objekte und Klassen



Objektorientierung

- Objekte und Klassen

```
class point {
    int x, y;
public:
    void moveTo(int newX, int newY)
        { x = newX; y = newY; }

    int xDistance(point *p)
    {
        return p->x - x;
    }
};
```

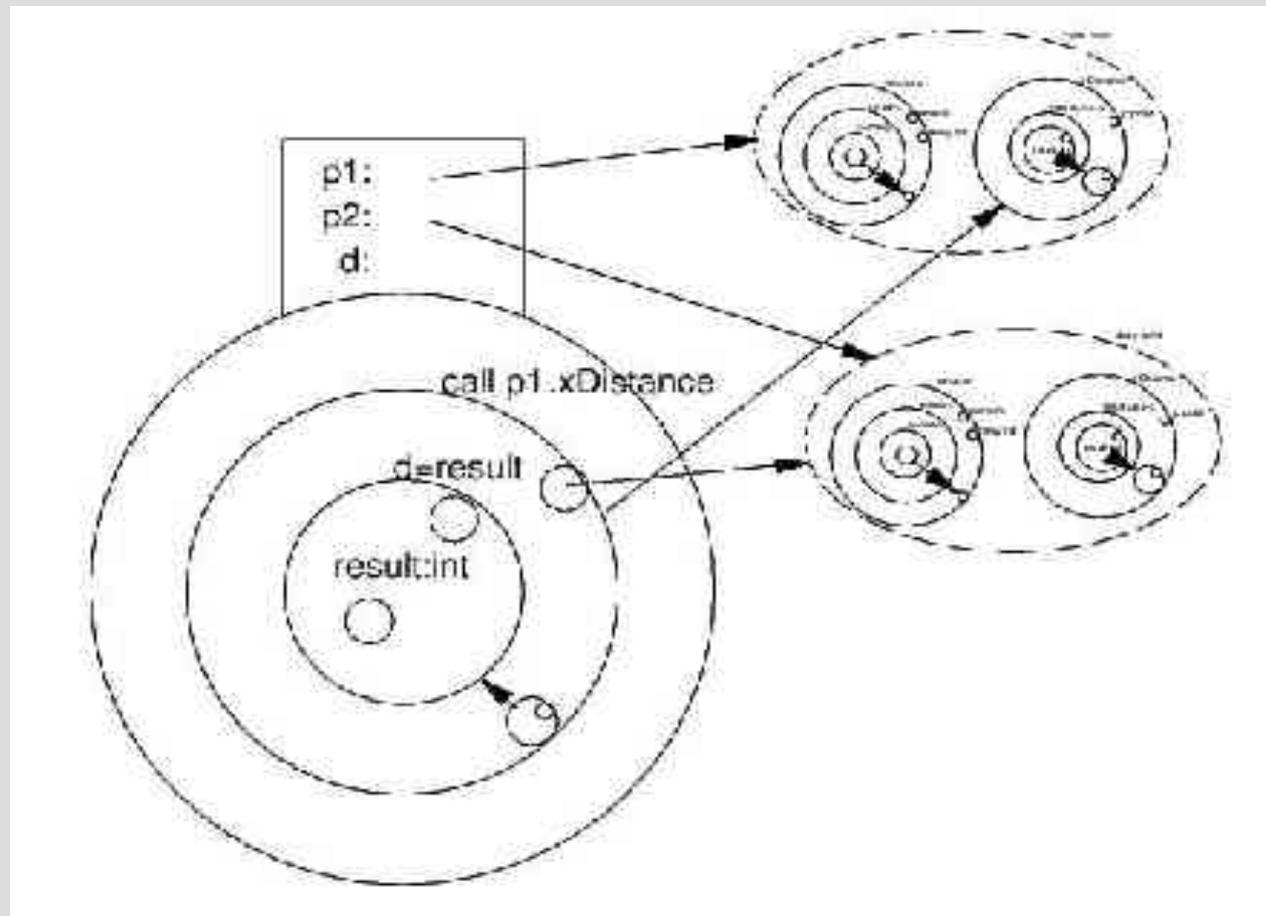
Objektorientierung

- Objekte und Klassen

```
main()
{
    point *p1, *p2;
    int d;
    p1 = new point;
    p2 = new point;
    p1->moveTo(3,4);
    p2->moveTo(0,0);
    d = p1->xDistance(p2);
}
```

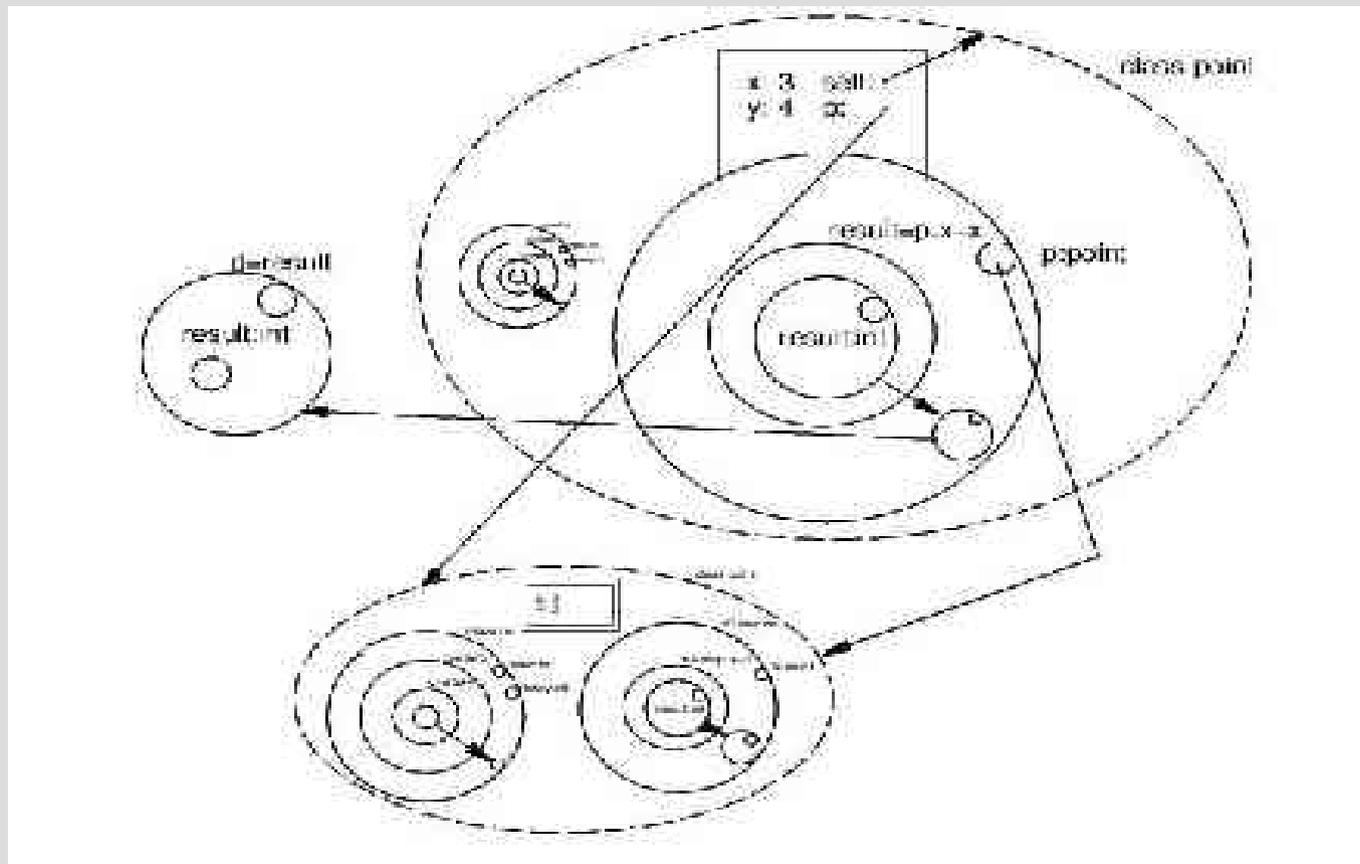

Objektorientierung

- Objekte und Klassen



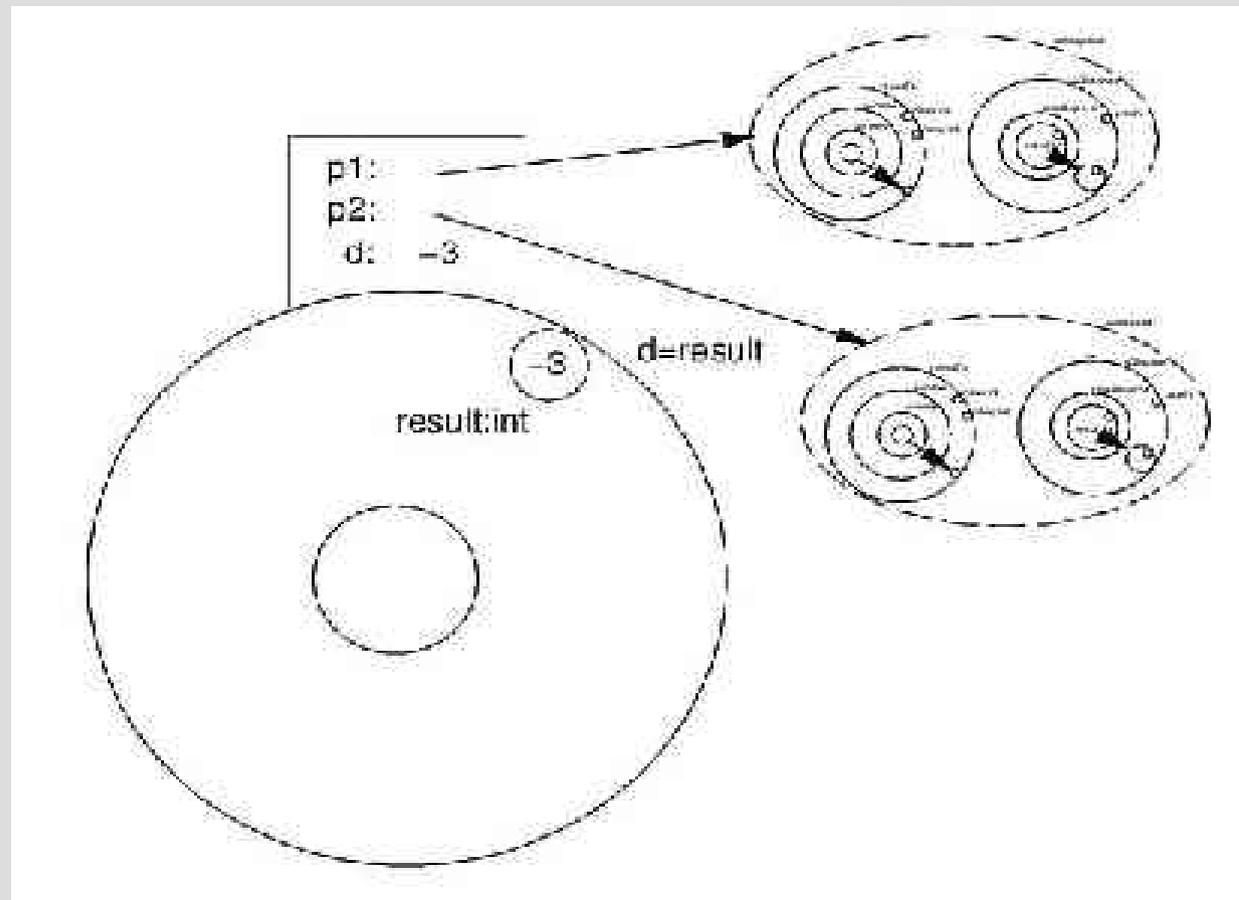
Objektorientierung

- Objekte und Klassen



Objektorientierung

- Objekte und Klassen



Objektorientierung

- Vererbung, Polymorphismus

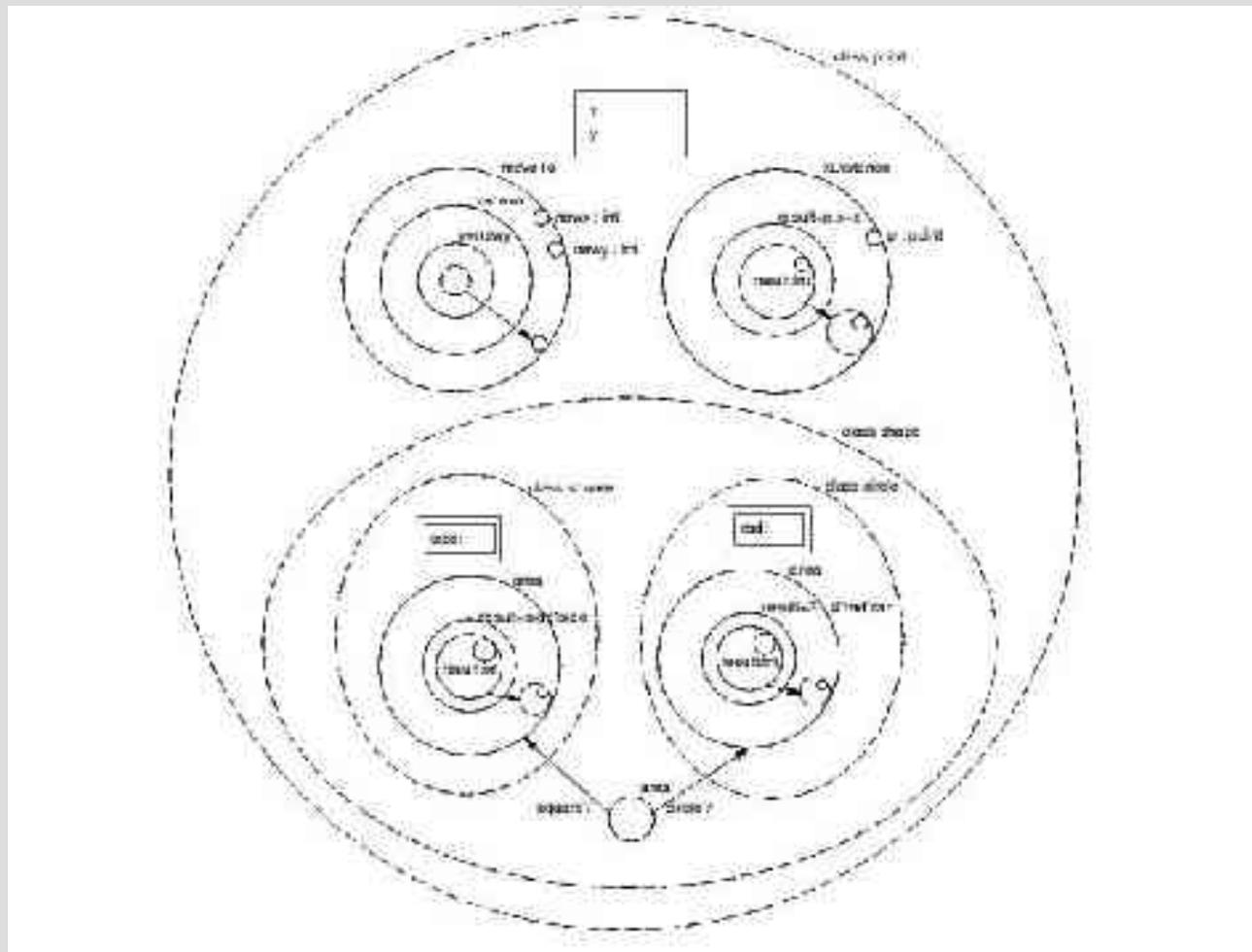
```
class shape : public point {  
public:  
    virtual double area() = 0;  
};
```

```
class square : public shape {  
    int side;  
public:  
    double area() { return side * side ;}  
};
```

```
class circle : public shape {  
    double rad;  
public:  
    double area() { return 3.14 * rad * rad ;}  
}
```

Objektorientierung

- Vererbung, Polymorphismus



Bewertung & Zusammenfassung

Negativ

- x Das Programm wird nicht mehr weiter entwickelt
- x Das Programm ist nicht mehr zu finden
- x Es ist nicht 100% visuell
- x Schnell unübersichtlich

Positiv

- ✓ Visueller als viele andere visuelle Sprachen
- ✓ Semantik ähnlich c++

Quellen

- 1. VIPR and the Visual Programming Challenge: A Potencial Solution using Scalability and Visual Semantics (1996), S. Ghiasi**
- 2. The Design of a Completely Visual Object-Oriented Programming Language (1994), W. Citrin, M. Doherty, B. Zorn**
- 3. Control Constructs in a Completely Visual Imperative Programming Language (1993), W. Citrin, M. Doherty, B. Zorn**