

# Kaya

Declarative Reactive

Declarative Like SQL

Employee.Age : Name = "Bob" ; Age^

*The equivalent in SQL*

SELECT Age FROM Employees WHERE Name = 'Bob' ORDER BY Age

*SELECTION : CONDITION ; GROUP BY and ORDER BY >> EACH*

Closer to OQL : Object containment, not joins

*The name of every museum director in Paris whose museum holds a Picasso*

Museum.Director.Name : Location =  
#Paris\_France + Artwork.Artist = #Pablo\_Picasso

*The budget of all films where the director's spouse's parents live in New York*

Film.Budget : Director.Spouse.Parent.Residence  
= #New\_York\_City

No explicit composite types

$x := 10$

$x$  10

Unordered List

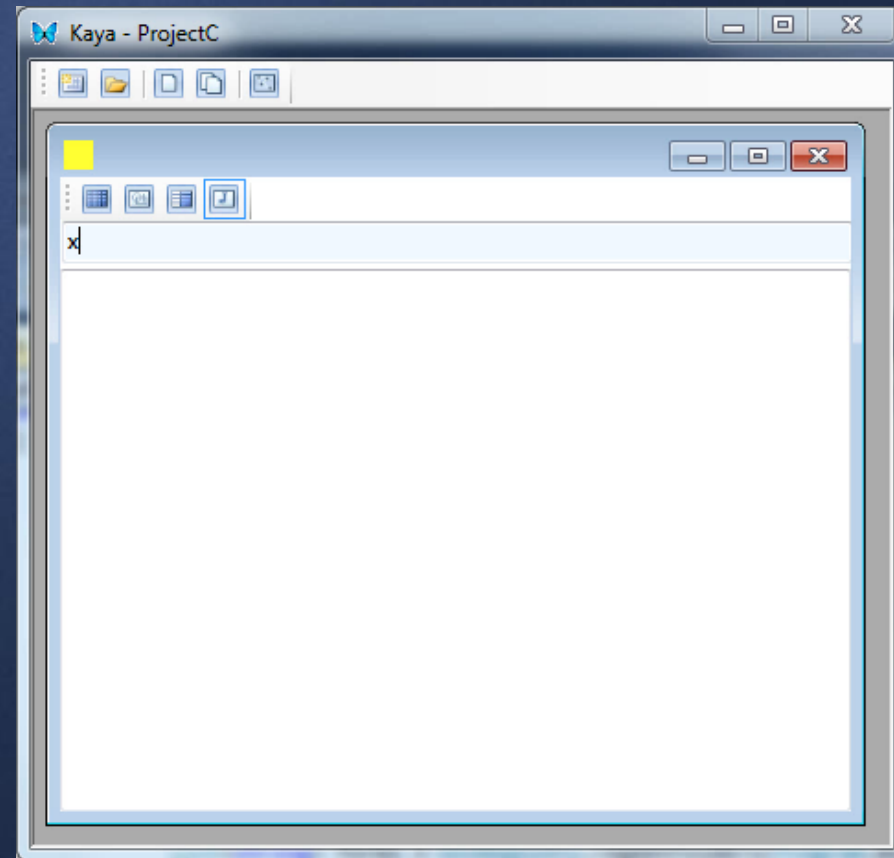
$x += 20$

$x$  10, 20

Replace contents

$x := 30$

$x$  30



Attributes / Structure

$x.y := 20$

`x.y` 20

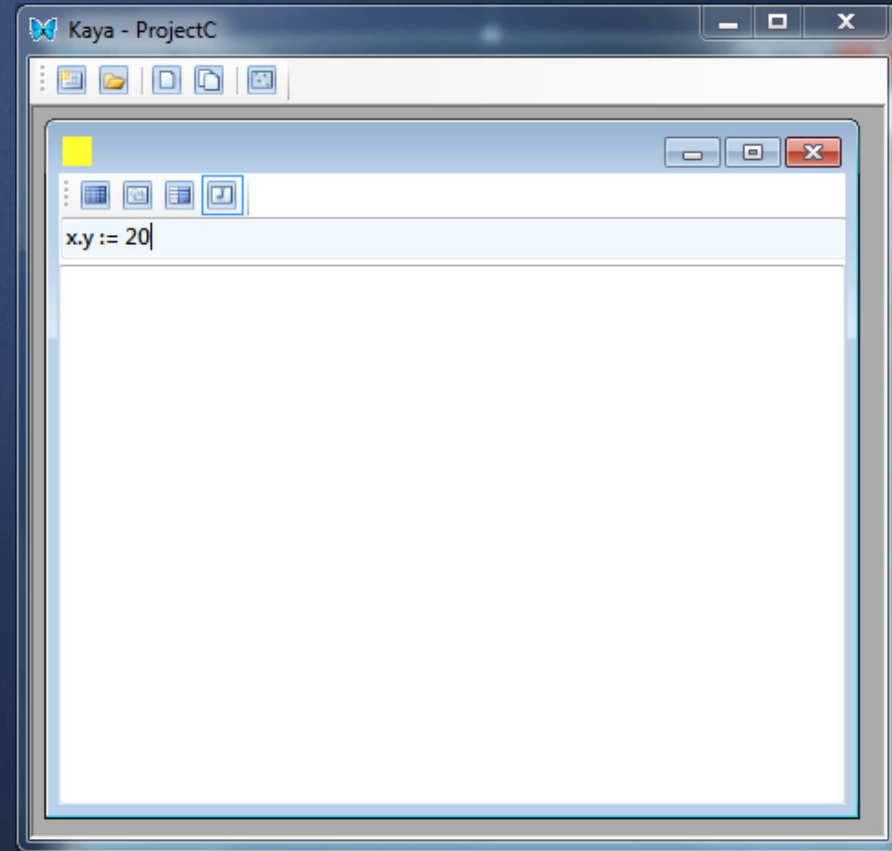
Indexed List

$c.'3 := 100$

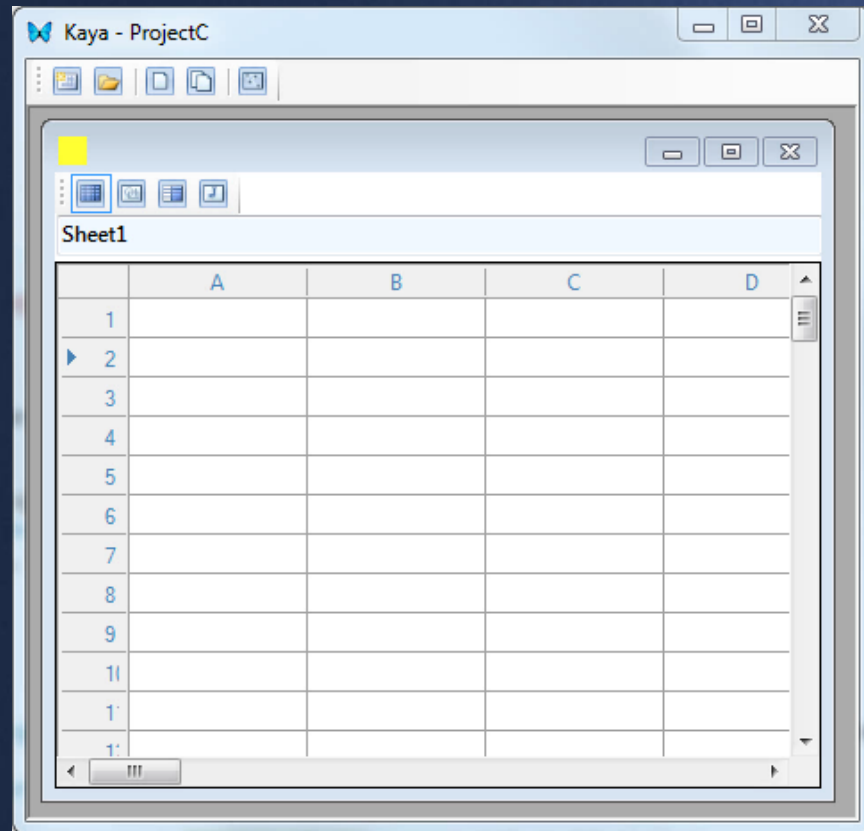
`c.'3` 100

Indexed list of structures

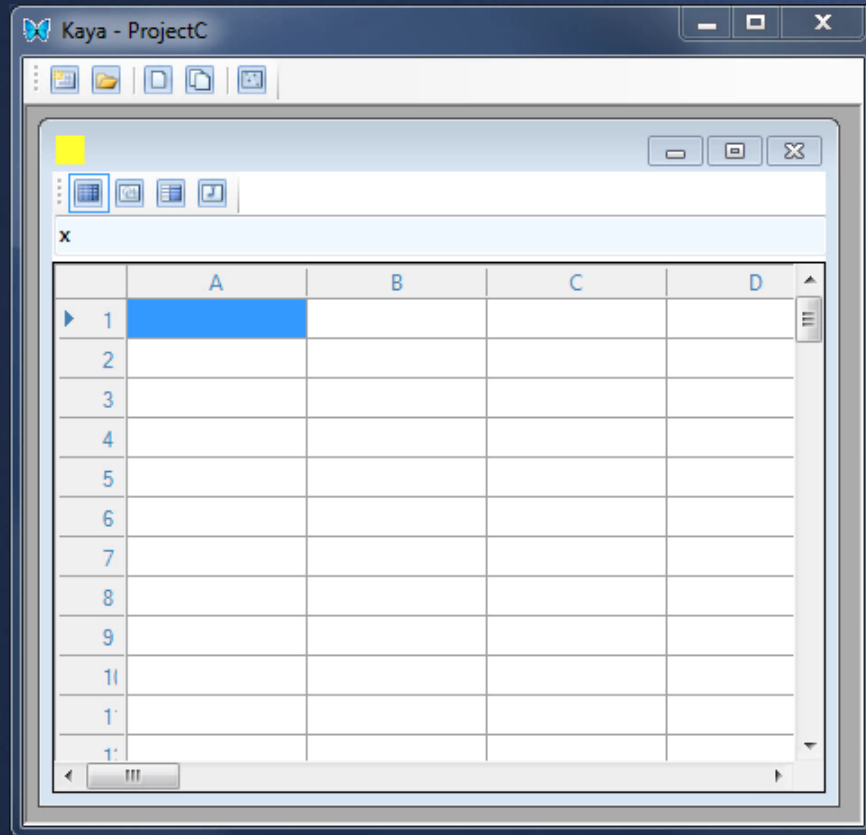
$\text{Employee.Name}'2 := \text{'Rob'}$



# Spreadsheet Metaphor



# Spreadsheet Metaphor



Scalar

$x := \text{"Hi"}$

Object

$y.C := \text{"Hello"}$

Indexed List

$z.5 := \text{"Bye"}$

The screenshot shows a software application window titled "Kaya - ProjectC". Inside the window, there is a sub-window titled "Employee" which contains a table. The table has 6 columns: an index column, "Name", "Age", "Start Date", "Department", and "E". There are 11 rows of data. The table is scrollable, with a scrollbar on the right and a status bar at the bottom showing "11" and a scroll button.

|    | Name     | Age | Start Date | Department        | E |
|----|----------|-----|------------|-------------------|---|
| 1  | Jane     | 21  | 5/5/13     | Sales             |   |
| 2  | Bob      | 32  | 12/1/13    | Marketing         |   |
| 3  | Jack     | 55  | 1/7/12     | Accounting        |   |
| 4  | Sophie   | 24  | 8/14/10    | Shipping          |   |
| 5  | Harrison | 46  | 8/26/12    | Marketing         |   |
| 6  | Paul     | 38  | 11/11/12   | Human Resources   |   |
| 7  | Jacob    | 27  | 3/17/10    | Shipping          |   |
| 8  | Terrance | 40  | 6/3/12     | Payroll           |   |
| 9  | William  | 60  | 7/5/10     | PMO               |   |
| 10 | Krystal  | 28  | 9/12/11    | Office Management |   |
| 11 | Lawrence | 55  | 10/31/13   | Legal             |   |

Syntax of criteria

Employee : Age < 30



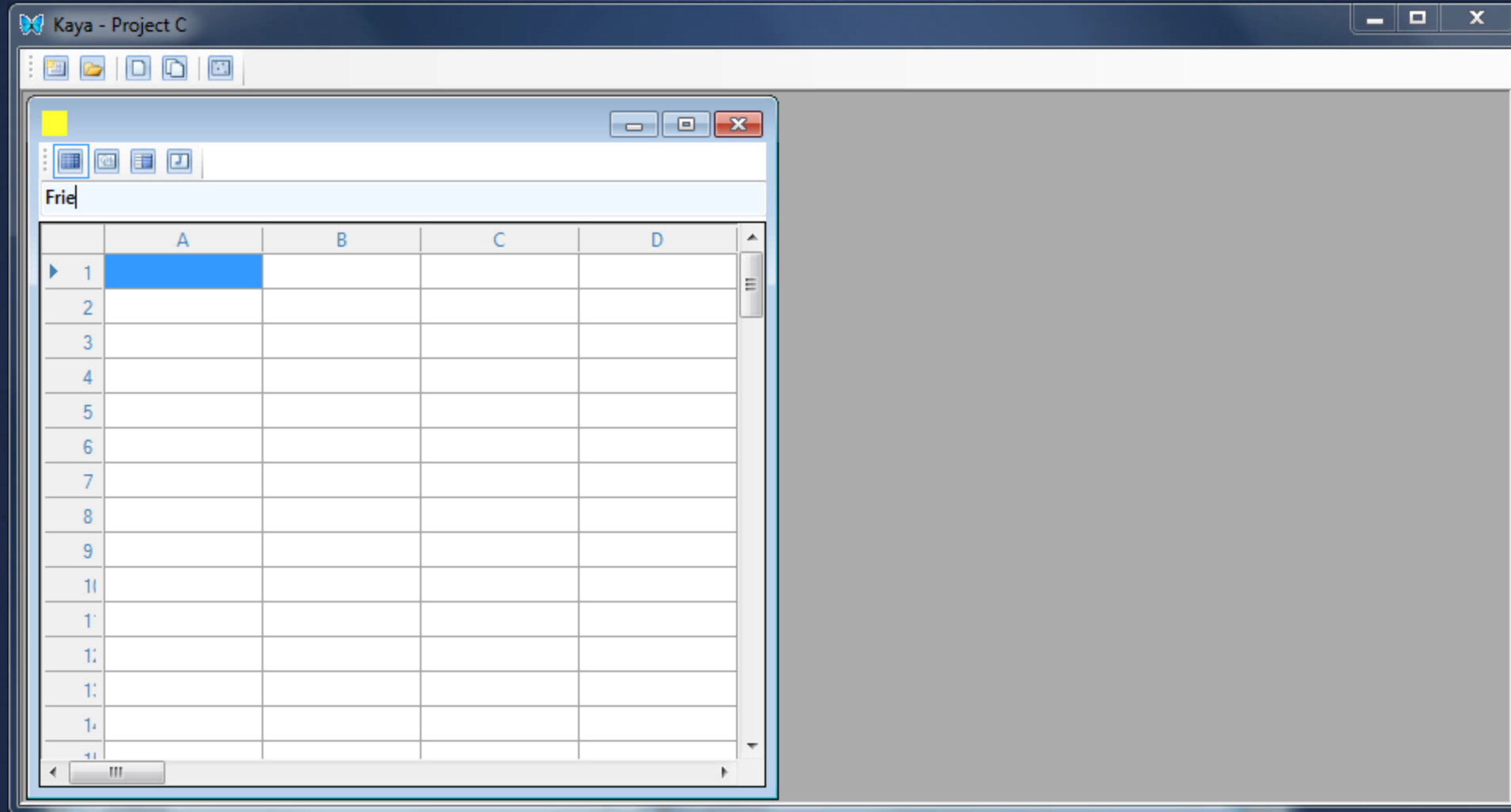
The screenshot shows a software application window titled "Kaya - ProjectC". Inside, there is a sub-window titled "Employee : Age < 30". This sub-window contains a table with the following data:

|    | Name    | Age | Start Date | Department        | E |
|----|---------|-----|------------|-------------------|---|
| 1  | Jane    | 21  | 5/5/13     | Sales             |   |
| 4  | Sophie  | 24  | 8/14/10    | Shipping          |   |
| 7  | Jacob   | 27  | 3/17/10    | Shipping          |   |
| 10 | Krystal | 28  | 9/12/11    | Office Management |   |
| 12 |         |     |            |                   |   |
| 13 |         |     |            |                   |   |
| 14 |         |     |            |                   |   |
| 15 |         |     |            |                   |   |
| 16 |         |     |            |                   |   |
| 17 |         |     |            |                   |   |
| 18 |         |     |            |                   |   |

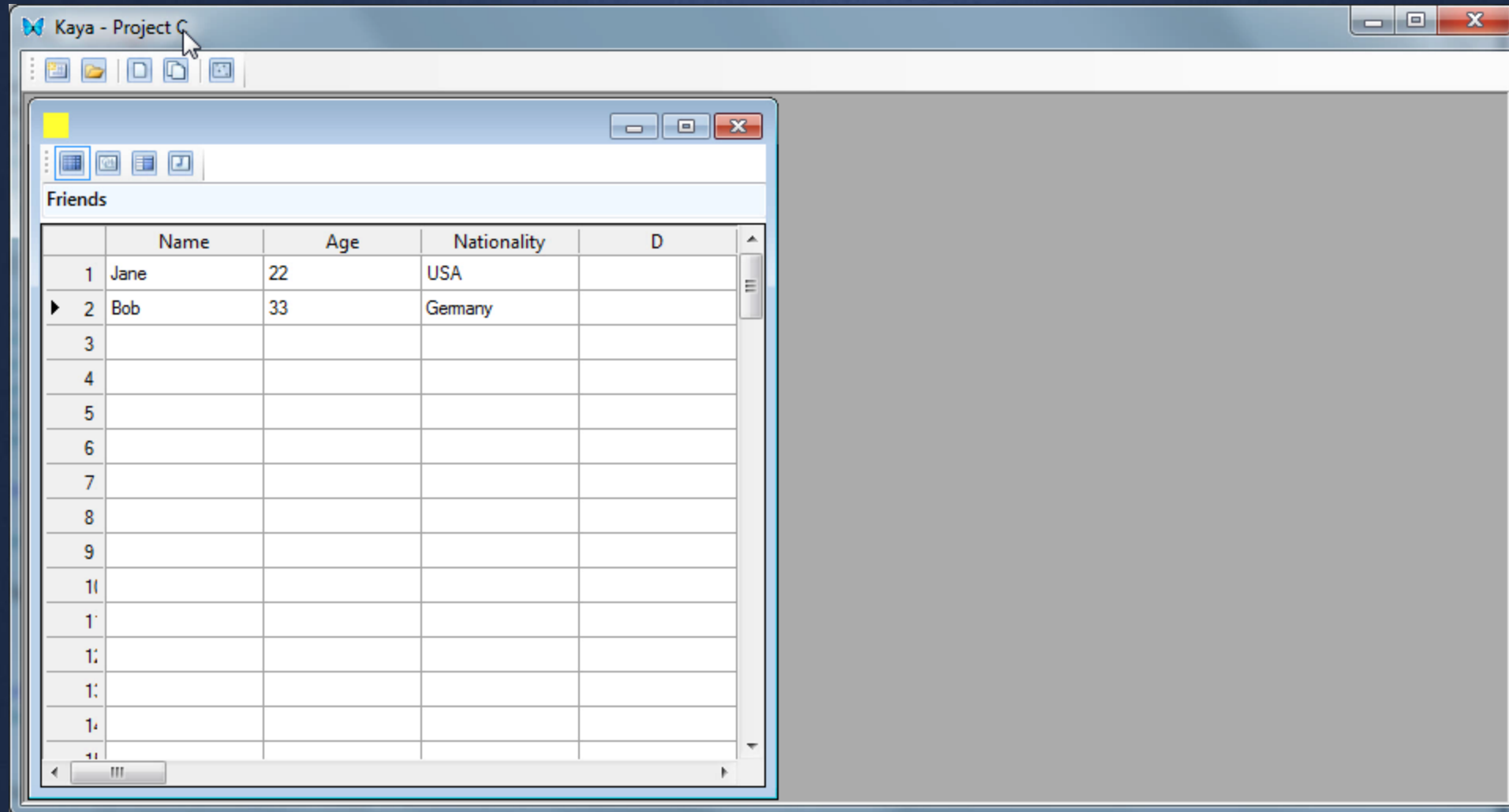
Syntax of criteria

Employee : Age < 30

# Linked Editor



# Linked Editor



# Linked Editor

The screenshot shows a software application window titled "Kaya - Project C". It contains two side-by-side panels, each with a table. The left panel is titled "Friends" and the right panel is titled ".Children'1". Both tables have a yellow icon in the top-left corner and a toolbar with icons for grid, list, and other views. The "Friends" table has columns: Name, Age, Nationality, and Children. The ".Children'1" table has columns: Name, Birthday, C, and D. The "Children" column in the "Friends" table is currently selected, and its value "Jim, Joe" is highlighted. The "Name" column in the ".Children'1" table is also selected, and its value "Jim" is highlighted. This visualizes a linked editor where changes in one table are reflected in another.

|    | Name | Age | Nationality     | Children      |
|----|------|-----|-----------------|---------------|
| 1  | Jane | 22  | USA             | Jim, Joe      |
| 2  | Bob  | 33  | Germany, France | Sophie, Alice |
| 3  |      |     |                 |               |
| 4  |      |     |                 |               |
| 5  |      |     |                 |               |
| 6  |      |     |                 |               |
| 7  |      |     |                 |               |
| 8  |      |     |                 |               |
| 9  |      |     |                 |               |
| 10 |      |     |                 |               |
| 11 |      |     |                 |               |
| 12 |      |     |                 |               |
| 13 |      |     |                 |               |
| 14 |      |     |                 |               |
| 15 |      |     |                 |               |

|    | Name | Birthday   | C | D |
|----|------|------------|---|---|
| 1  | Jim  | 12/12/2006 |   |   |
| 2  | Joe  | 10/30/2007 |   |   |
| 3  |      |            |   |   |
| 4  |      |            |   |   |
| 5  |      |            |   |   |
| 6  |      |            |   |   |
| 7  |      |            |   |   |
| 8  |      |            |   |   |
| 9  |      |            |   |   |
| 10 |      |            |   |   |
| 11 |      |            |   |   |
| 12 |      |            |   |   |
| 13 |      |            |   |   |
| 14 |      |            |   |   |
| 15 |      |            |   |   |

# Linked Editor

The screenshot shows the Kaya - Project C application with two linked editors. The left editor displays a table titled "Friends.Children" with columns Name, Birthday, C, and D. The right editor displays an empty table with columns A, B, C, and D. Both editors have a toolbar with icons for file operations and a scroll bar on the right.

|    | Name   | Birthday   | C | D |
|----|--------|------------|---|---|
| 1  | Jim    | 12/12/2006 |   |   |
| 2  | Joe    | 10/30/2007 |   |   |
| 3  | Sophie | 8/14/2005  |   |   |
| 4  | Alice  | 7/11/2007  |   |   |
| 5  |        |            |   |   |
| 6  |        |            |   |   |
| 7  |        |            |   |   |
| 8  |        |            |   |   |
| 9  |        |            |   |   |
| 10 |        |            |   |   |
| 11 |        |            |   |   |
| 12 |        |            |   |   |
| 13 |        |            |   |   |
| 14 |        |            |   |   |
| 15 |        |            |   |   |

|    | A | B | C | D |
|----|---|---|---|---|
| 1  |   |   |   |   |
| 2  |   |   |   |   |
| 3  |   |   |   |   |
| 4  |   |   |   |   |
| 5  |   |   |   |   |
| 6  |   |   |   |   |
| 7  |   |   |   |   |
| 8  |   |   |   |   |
| 9  |   |   |   |   |
| 10 |   |   |   |   |
| 11 |   |   |   |   |
| 12 |   |   |   |   |
| 13 |   |   |   |   |
| 14 |   |   |   |   |
| 15 |   |   |   |   |

The screenshot shows a software application window titled "Kaya - ProjectC". Inside the window, there is a sub-window titled "Employee" which contains a table. The table has 6 columns: an index column, "Name", "Age", "Start Date", "Department", and "E". There are 11 rows of data in the table.

|    | Name     | Age | Start Date | Department        | E |
|----|----------|-----|------------|-------------------|---|
| 1  | Jane     | 21  | 5/5/13     | Sales             |   |
| 2  | Bob      | 32  | 12/1/13    | Marketing         |   |
| 3  | Jack     | 55  | 1/7/12     | Accounting        |   |
| 4  | Sophie   | 24  | 8/14/10    | Shipping          |   |
| 5  | Harrison | 46  | 8/26/12    | Marketing         |   |
| 6  | Paul     | 38  | 11/11/12   | Human Resources   |   |
| 7  | Jacob    | 27  | 3/17/10    | Shipping          |   |
| 8  | Terrance | 40  | 6/3/12     | Payroll           |   |
| 9  | William  | 60  | 7/5/10     | PMO               |   |
| 10 | Krystal  | 28  | 9/12/11    | Office Management |   |
| 11 | Lawrence | 55  | 10/31/13   | Legal             |   |

Syntax of insert

Employee += @(Name = "John", Age = 32)

@ is Clone

$y := @x$

|          |   |
|----------|---|
| y.parent | x |
|----------|---|

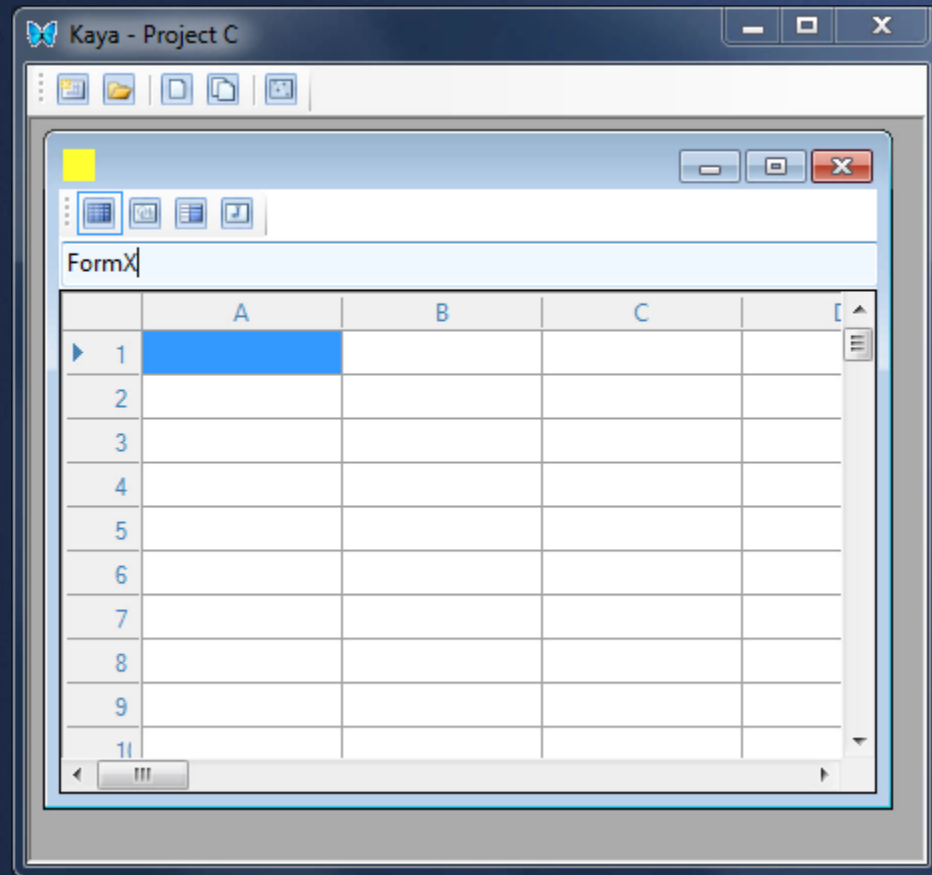
# is Identifier

#unique\_id

System objects

#graphicalobj

# Vector Editor

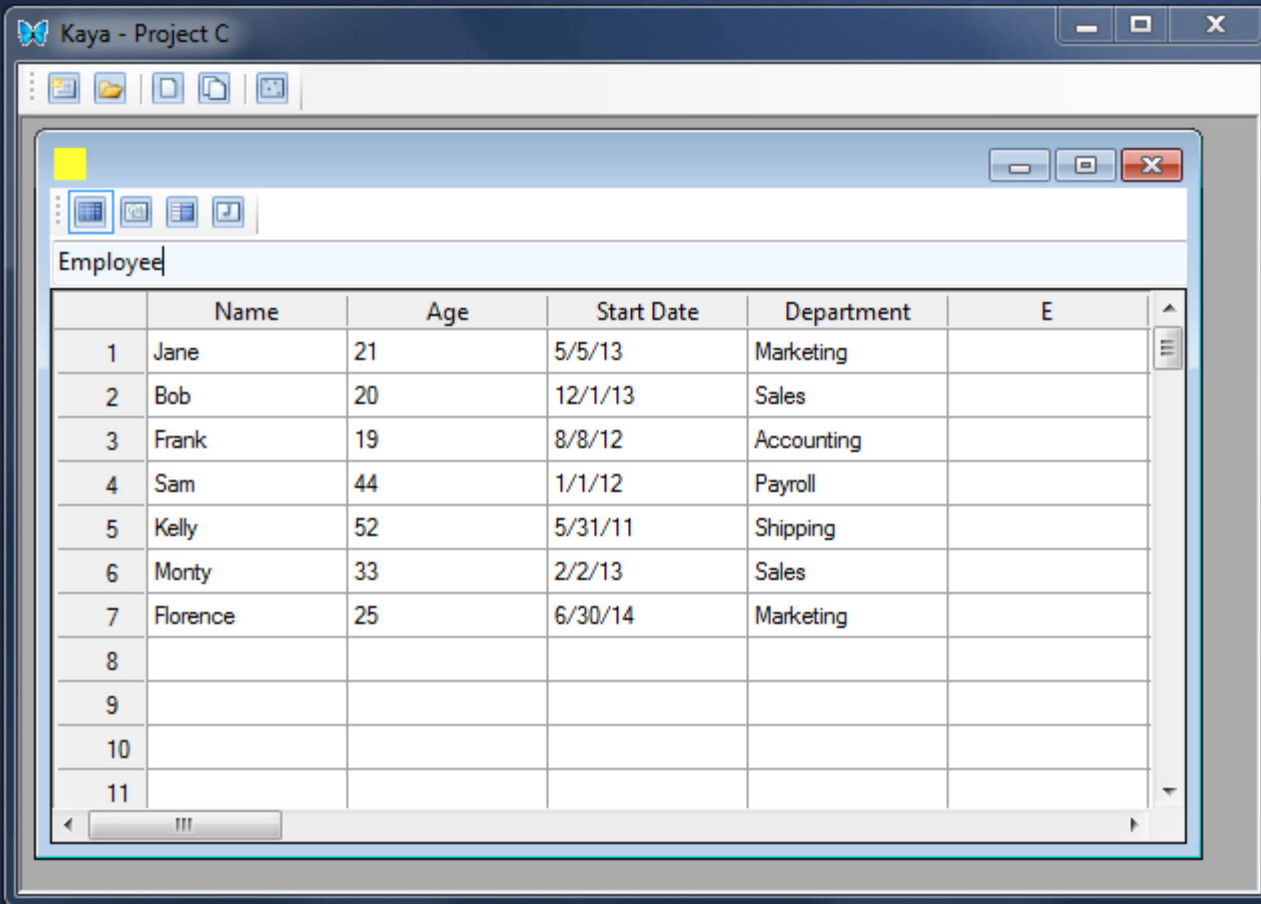


Add graphical object to form

```
FormX += @#graphicalobj(shape=#rectangle, x1=32, ...)
```



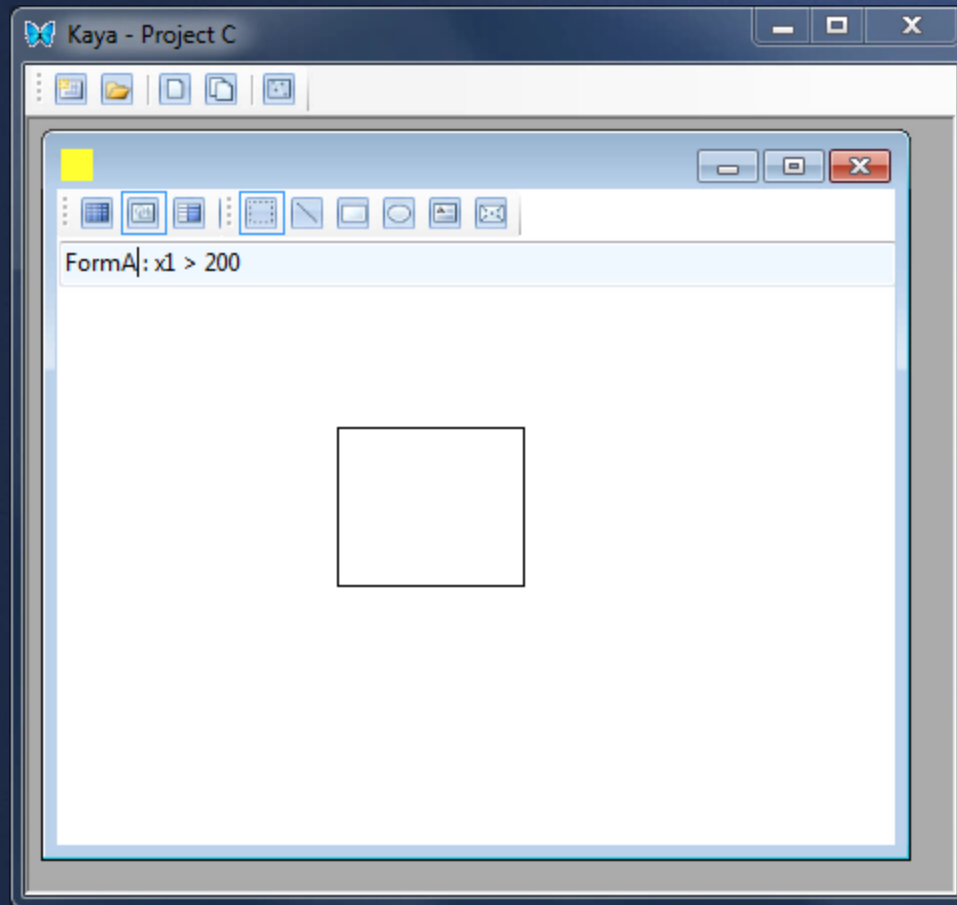
# Attribute Editor



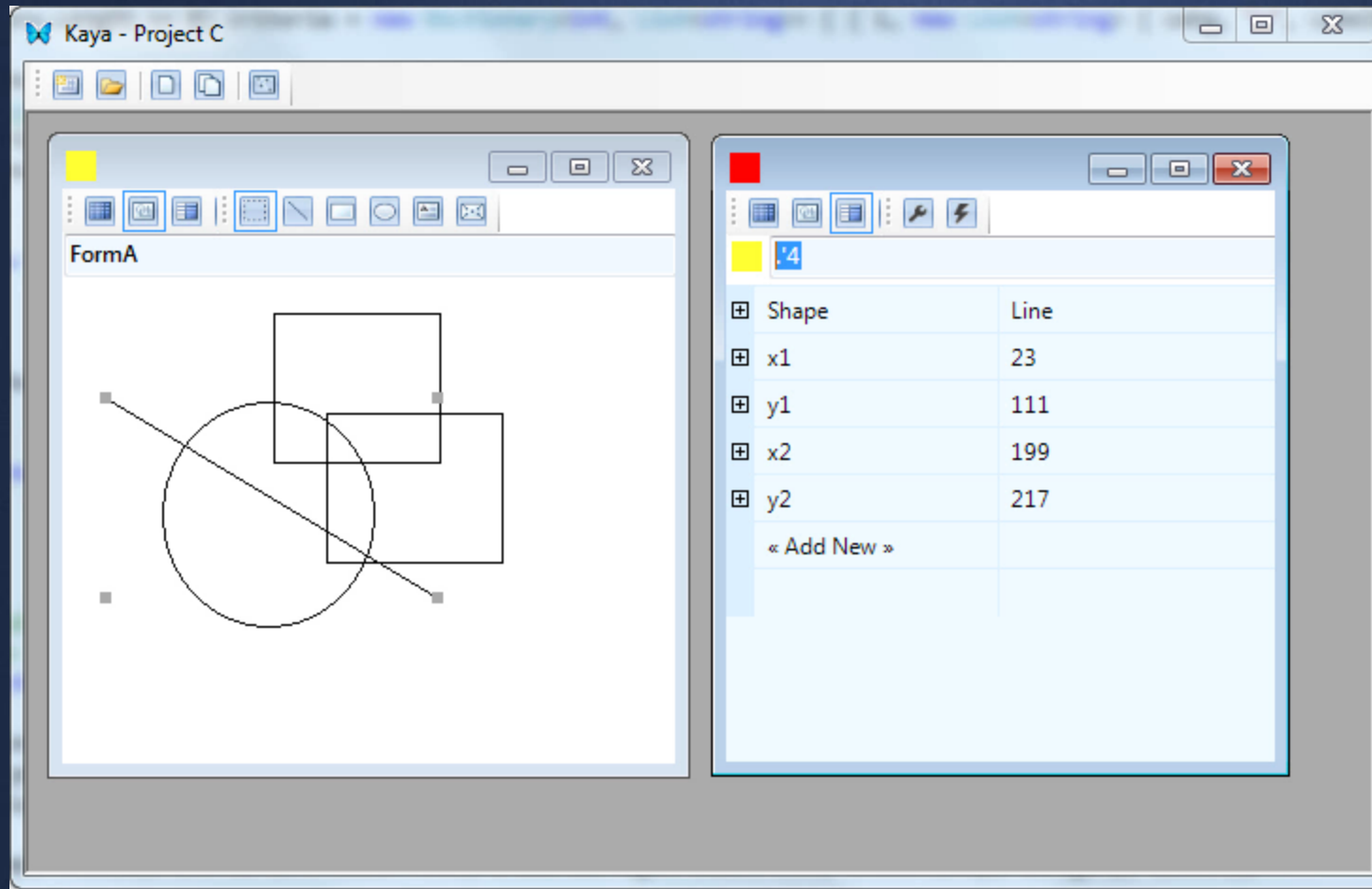
The screenshot shows a software window titled "Kaya - Project C". Inside, there is a sub-window titled "Employee" which contains a table. The table has 6 columns: an index column, "Name", "Age", "Start Date", "Department", and "E". There are 11 rows in total. The first 7 rows contain employee data, and the last 4 rows are empty.

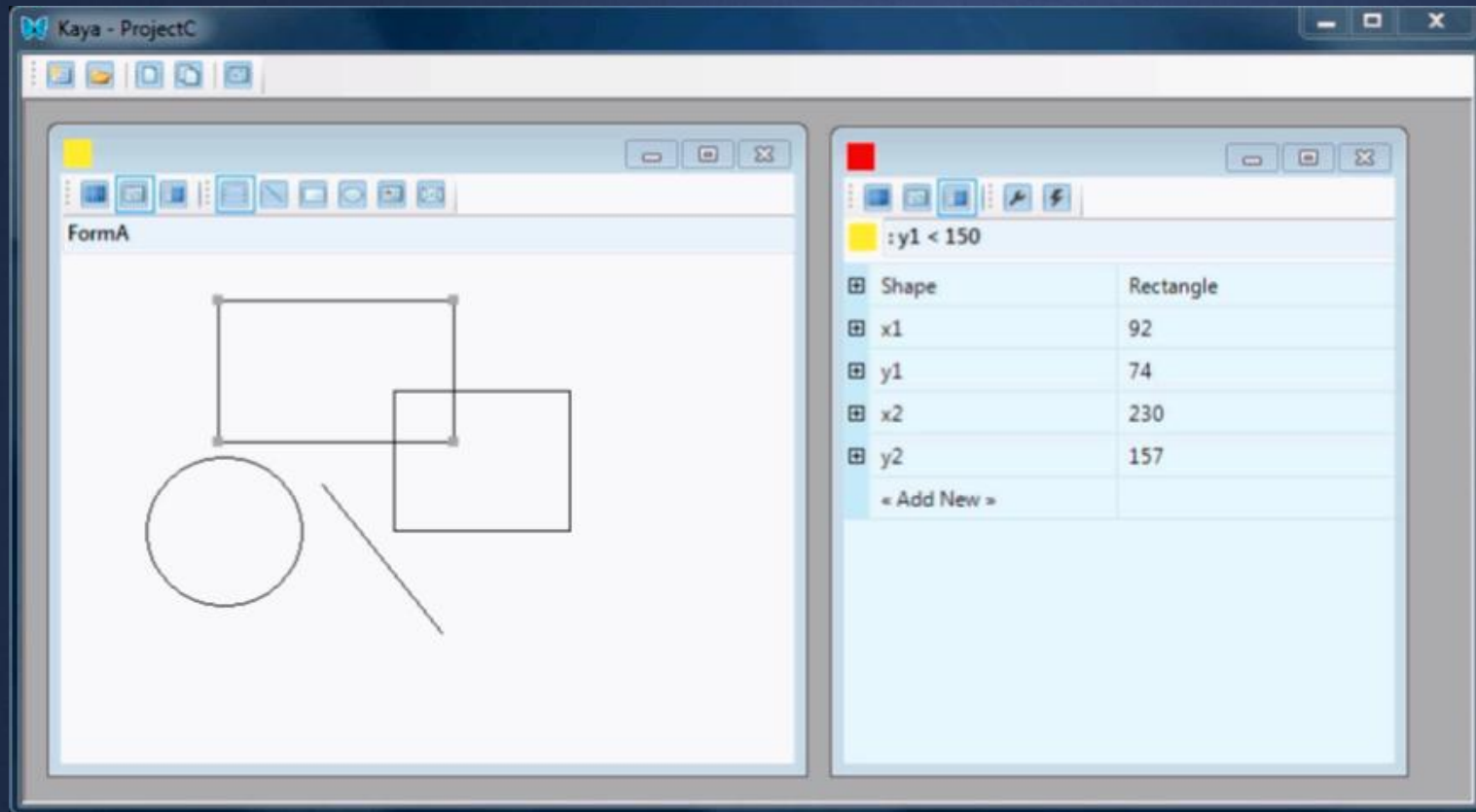
|    | Name     | Age | Start Date | Department | E |
|----|----------|-----|------------|------------|---|
| 1  | Jane     | 21  | 5/5/13     | Marketing  |   |
| 2  | Bob      | 20  | 12/1/13    | Sales      |   |
| 3  | Frank    | 19  | 8/8/12     | Accounting |   |
| 4  | Sam      | 44  | 1/1/12     | Payroll    |   |
| 5  | Kelly    | 52  | 5/31/11    | Shipping   |   |
| 6  | Monty    | 33  | 2/2/13     | Sales      |   |
| 7  | Florence | 25  | 6/30/14    | Marketing  |   |
| 8  |          |     |            |            |   |
| 9  |          |     |            |            |   |
| 10 |          |     |            |            |   |
| 11 |          |     |            |            |   |

# Attribute Editor



# Attribute Editor

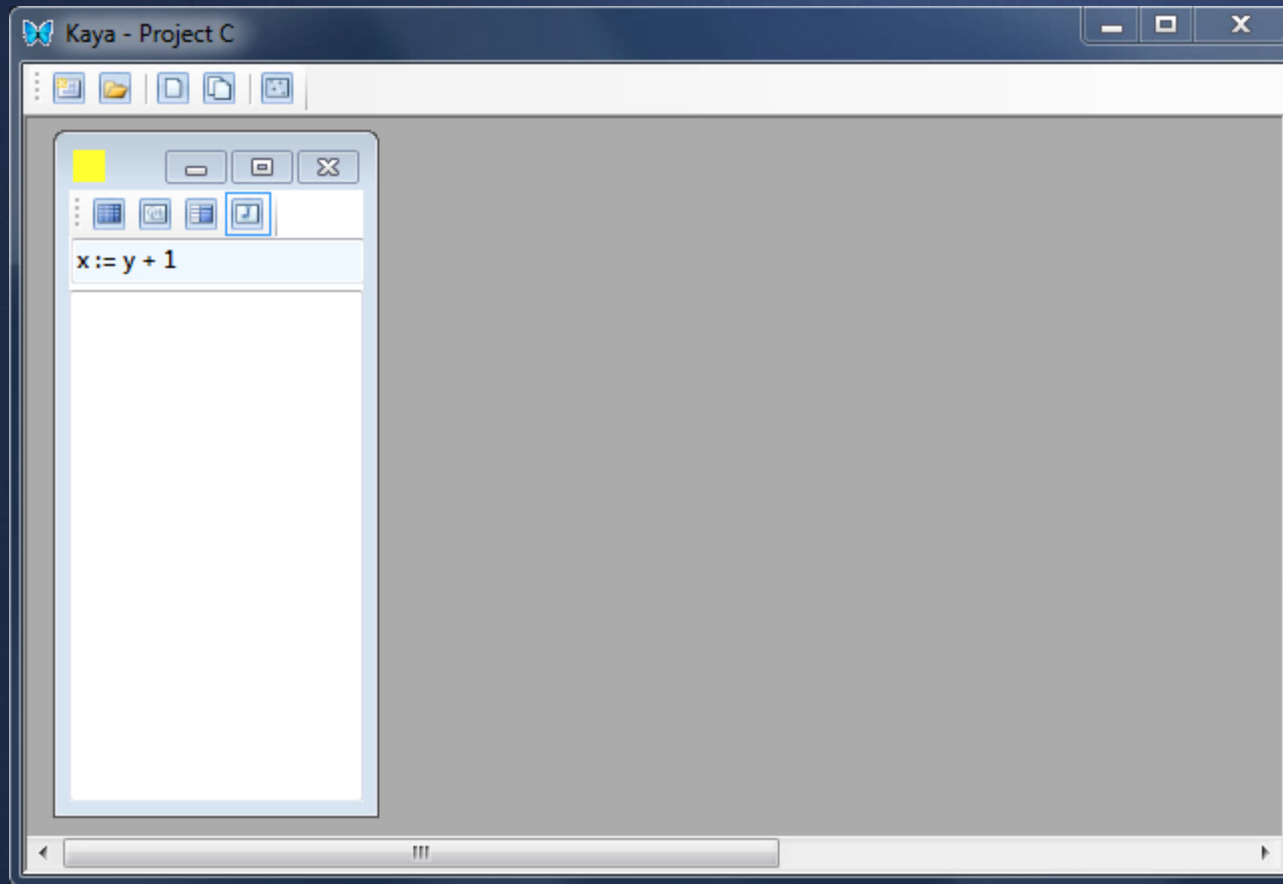




Continuous Binding, Attribute Chaining

(FormA : y1 < 150).This.That := "Some value"

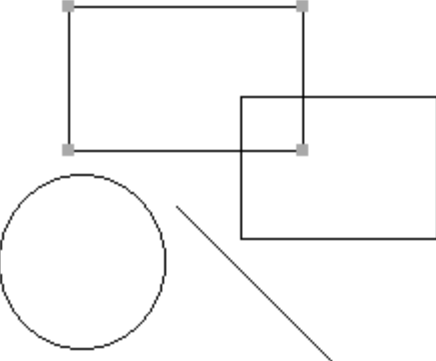
Reactive: Continually evented



# Reactive

Kaya - Project C

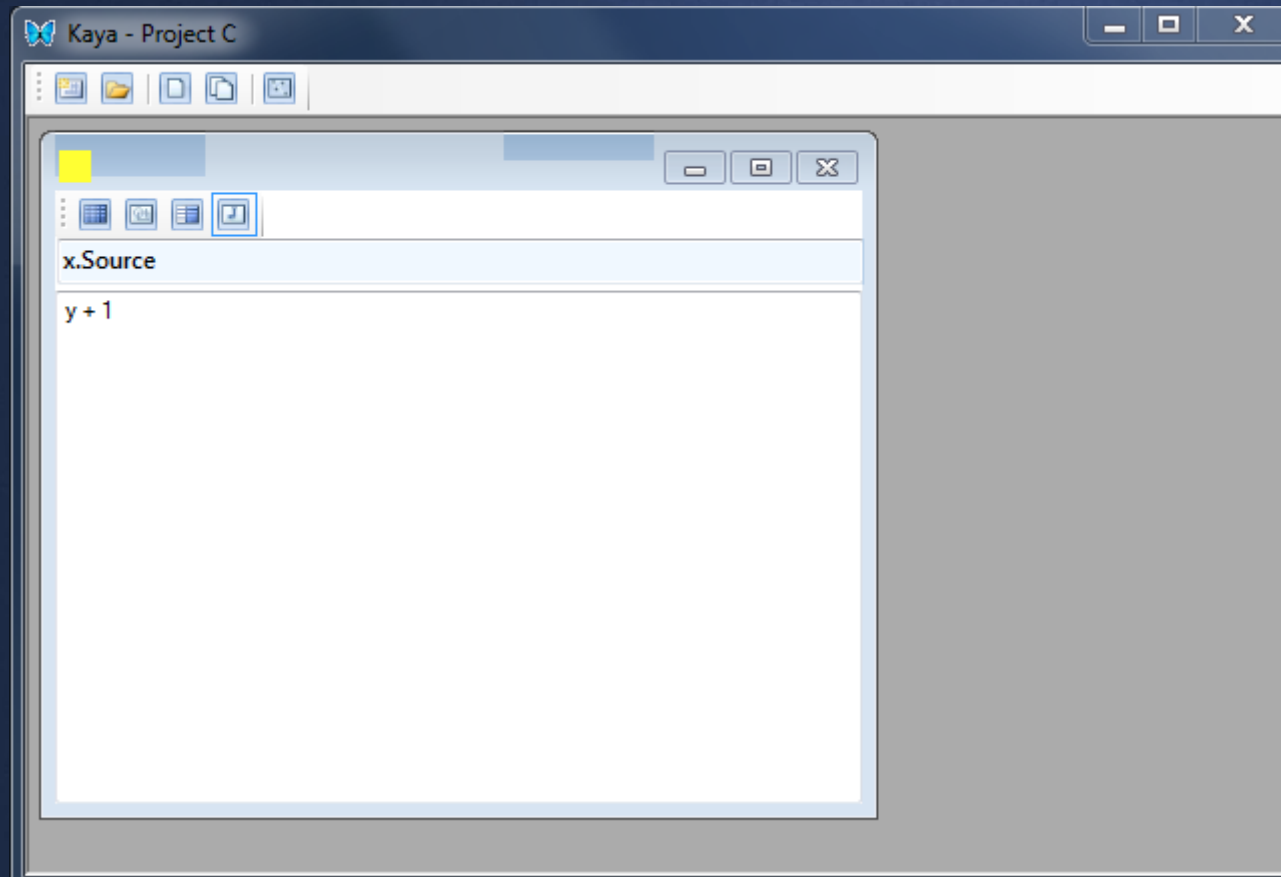
FormA



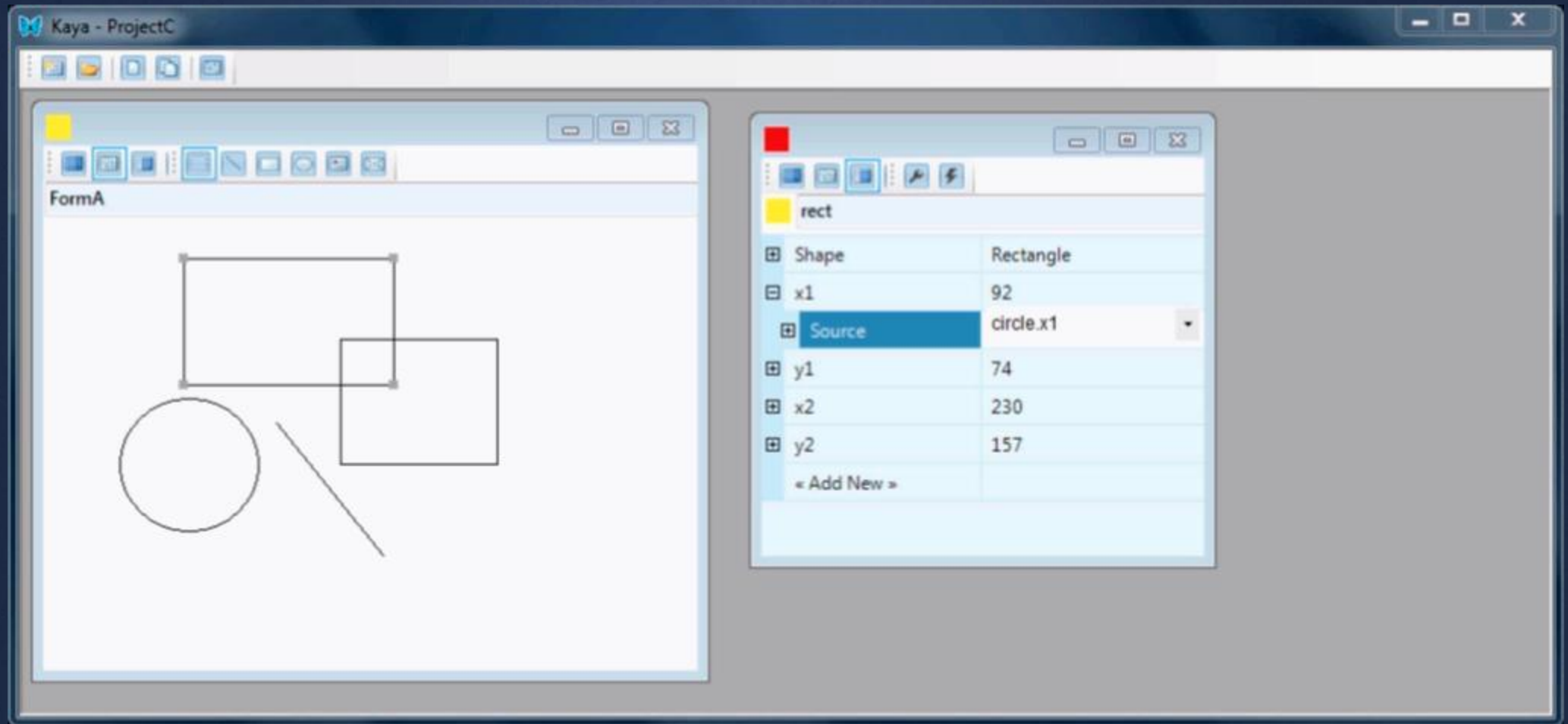
.1.Contains

|     | Shape | x1    | y1    | x2    | y2    |
|-----|-------|-------|-------|-------|-------|
| ▶ 1 | Line  | =..x1 | =..y1 | =..x2 | =..y1 |
| 2   | Line  | =1.x2 | =1.y1 | =.x1  | =..y2 |
| 3   | Line  | =1.x1 | =2.y2 | =1.x2 | =.y1  |
| 4   | Line  | =1.x1 | =1.y1 | =.x1  | =2.y2 |
| 5   |       |       |       |       |       |
| 6   |       |       |       |       |       |
| 7   |       |       |       |       |       |
| 8   |       |       |       |       |       |
| 9   |       |       |       |       |       |
| 10  |       |       |       |       |       |
| 11  |       |       |       |       |       |
| 12  |       |       |       |       |       |
| 13  |       |       |       |       |       |
| 14  |       |       |       |       |       |
| 15  |       |       |       |       |       |

## Standard library (.Source)



## Standard library (.Strict)





Standard library (.Parent)

$x := @WholeNumber$

$x := @WholeNumber : > 10 + < 100$

$z.Parent := ( Employees : Age < 20 )$

Positive and negative variance

Trainees := @Employee

Employee

|            |
|------------|
| Name       |
| Age        |
| Department |
| Start Date |



Trainees

|                  |
|------------------|
| Name             |
| Age              |
| Department       |
| Start Date       |
| Training Program |

The screenshot shows a Kaya IDE window titled "Kaya - Project C". Inside the window, there is a table named "Trainees". The table has five columns: an index column, "Name", "Age", "Department", "D", and "E". The first two rows are populated with data: Jane (19, Sales) and Bill (18, Marketing). The remaining rows are empty.

|   | Name | Age | Department | D | E |
|---|------|-----|------------|---|---|
| 1 | Jane | 19  | Sales      |   |   |
| 2 | Bill | 18  | Marketing  |   |   |
| 3 |      |     |            |   |   |
| 4 |      |     |            |   |   |
| 5 |      |     |            |   |   |
| 6 |      |     |            |   |   |
| 7 |      |     |            |   |   |
| 8 |      |     |            |   |   |

Static and dynamic sets

Trainees := @|(Employee : Age < 20)

The screenshot shows a Kaya IDE window titled "Kaya - Project C". Inside, there is a table named "Trainees" with 8 rows and 6 columns. The columns are labeled "Name", "Age", "Department", "D", and "E". The first three rows are populated with data: Jane (19, Sales), Bill (18, Marketing), and Sophie (24, Accounting). The remaining five rows are empty.

|   | Name   | Age | Department | D | E |
|---|--------|-----|------------|---|---|
| 1 | Jane   | 19  | Sales      |   |   |
| 2 | Bill   | 18  | Marketing  |   |   |
| 3 | Sophie | 24  | Accounting |   |   |
| 4 |        |     |            |   |   |
| 5 |        |     |            |   |   |
| 6 |        |     |            |   |   |
| 7 |        |     |            |   |   |
| 8 |        |     |            |   |   |

Static and dynamic sets

Trainees := @|(Employee : Age < 20)

Trainees += #m3bc54

Kaya - Project C

Trainees

|   | Name   | Age | Department | D | E |
|---|--------|-----|------------|---|---|
| 1 | Jane   | 19  | Sales      |   |   |
| 2 | Bill   | 18  | Marketing  |   |   |
| 3 | Sophie | 24  | Accounting |   |   |
| 4 | Bob    | 25  | Shipping   |   |   |
| 5 |        |     |            |   |   |
| 6 |        |     |            |   |   |
| 7 |        |     |            |   |   |
| 8 |        |     |            |   |   |

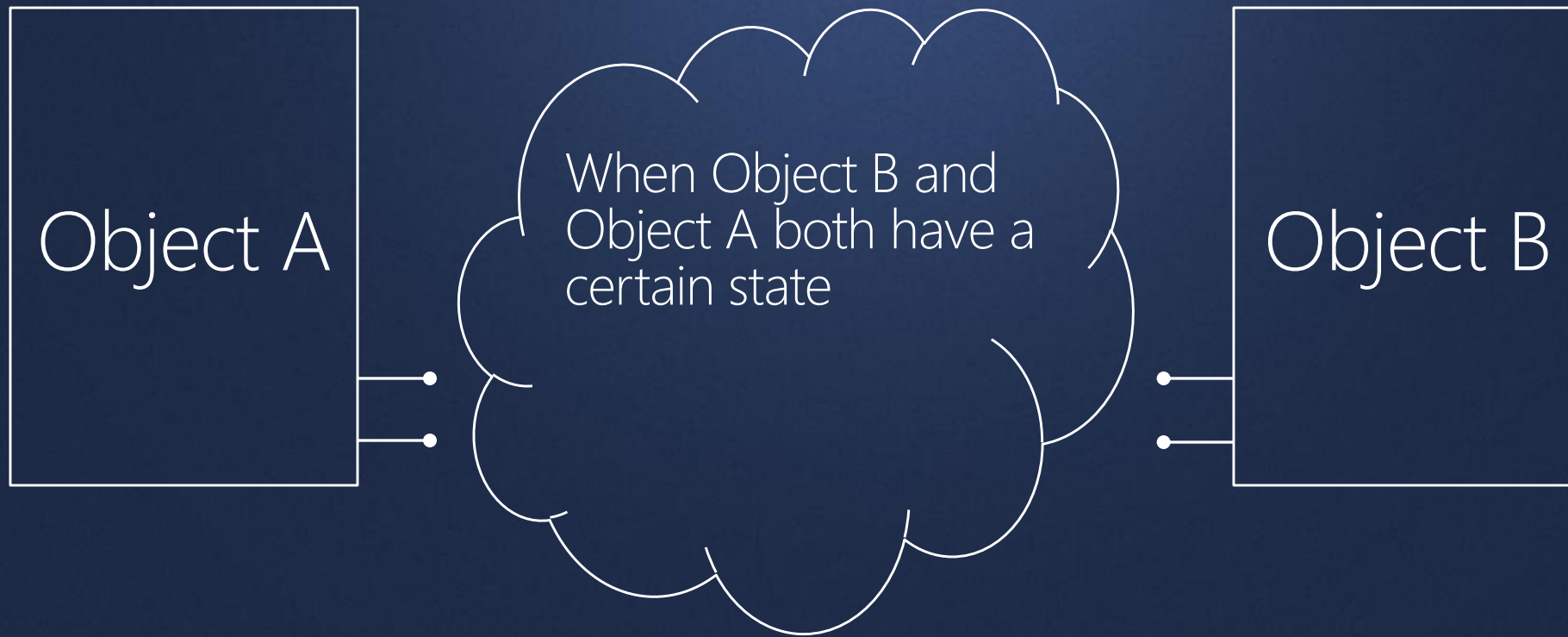
Static and dynamic sets

Trainees := @|(Employee : Age < 20)

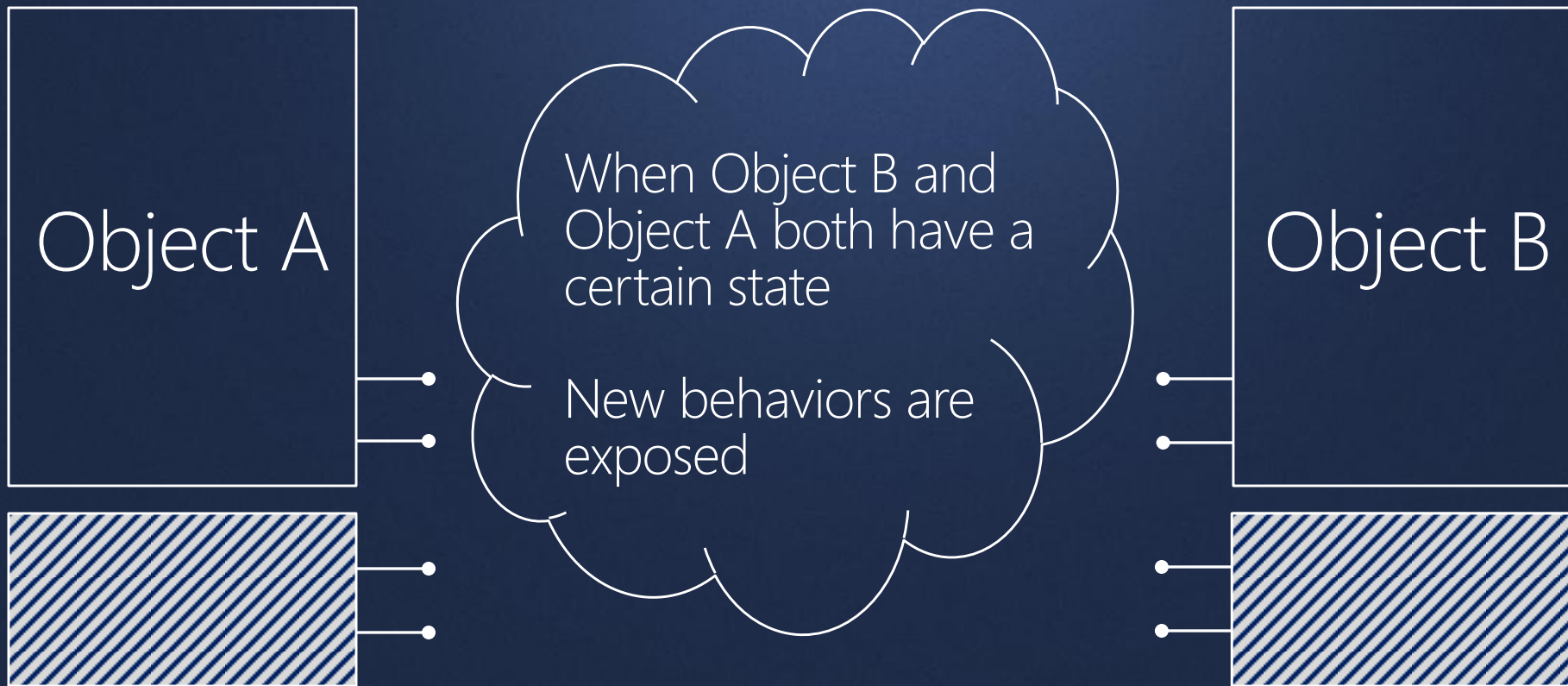
Trainees += #m3bc54

Trainees += (Name = "Bob", Age = 25)

Variance is autonomous



Variance is autonomous



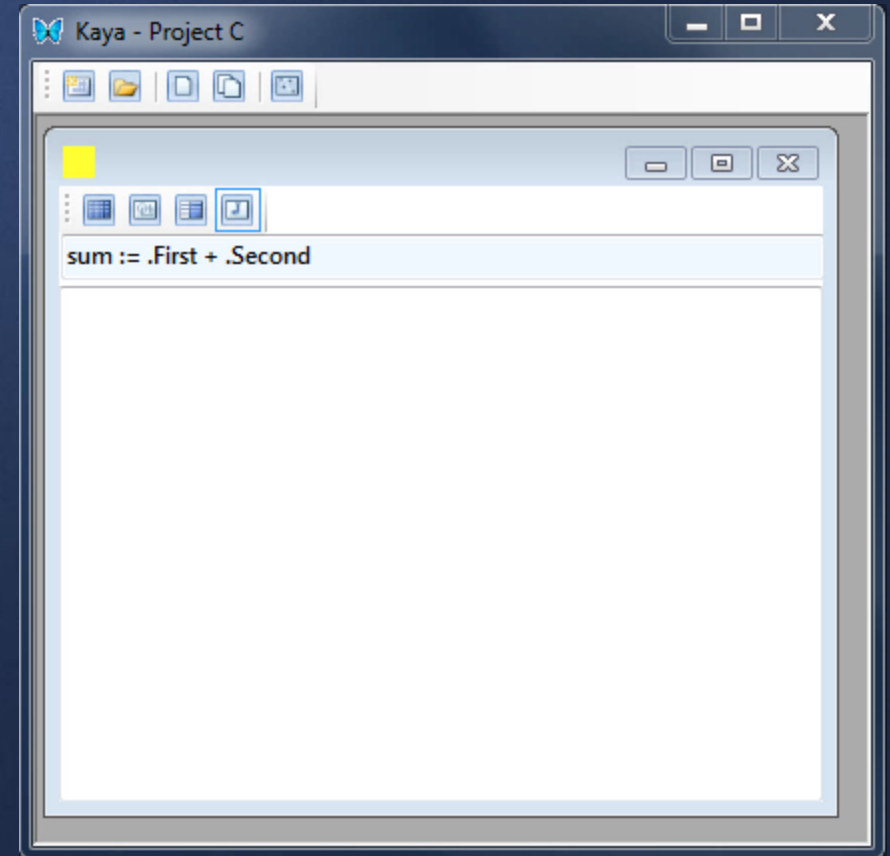
## Expressions

`sum := .First + .Second`

`sum(.First:=5, .Second:=6)`

`sum := .A + .B`

`sum(5, 6)`





Events

EventA := (x, y : x > 10 + x < 100, y < 20)

EventA += ( #click )

EventA.Action += ( xyz := 100 )

## Control Structures

### Conditionals

- Edge Events
- Exists attribute (gives context)

### Looping

- Duration Events
- Ranges >> each

Declarative Like SQL

Employee.Age : Name = "Bob" ; Age^

*The equivalent in SQL*

SELECT Age FROM Employees WHERE Name = 'Bob' ORDER BY Age

*SELECTION : CONDITION ; GROUP BY and ORDER BY >> EACH*

## Control Structures

### Conditionals

- Edge Events
- Exists attribute (gives context)

### Looping

- Duration Events
- Ranges >> each

# Virtual Machine



- Key / value datastore
- Insert-only, eventually-consistent

# Virtual Machine



# Virtual Machine

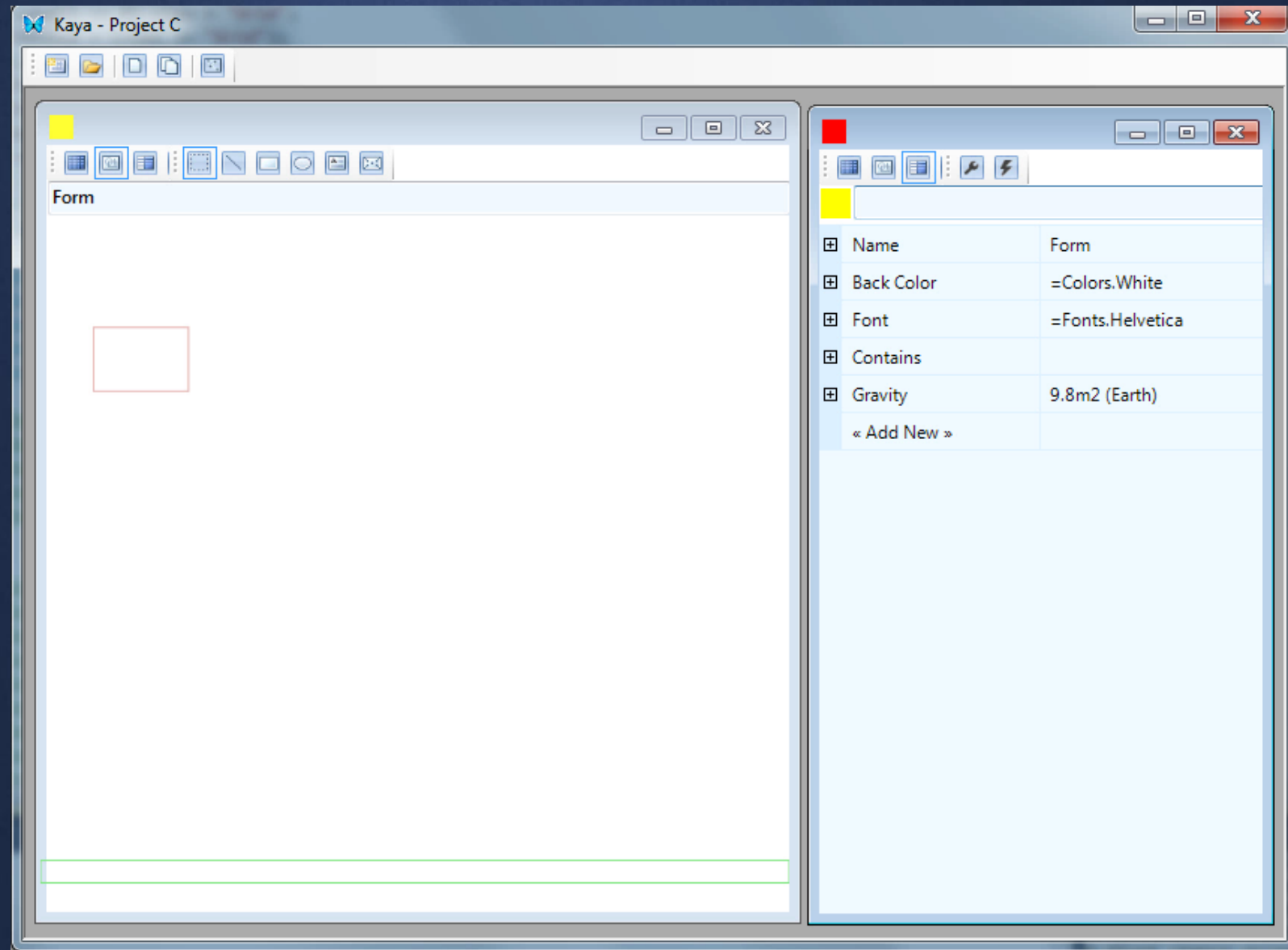


• Key / value datastore

• Insert-only, eventually-consistent

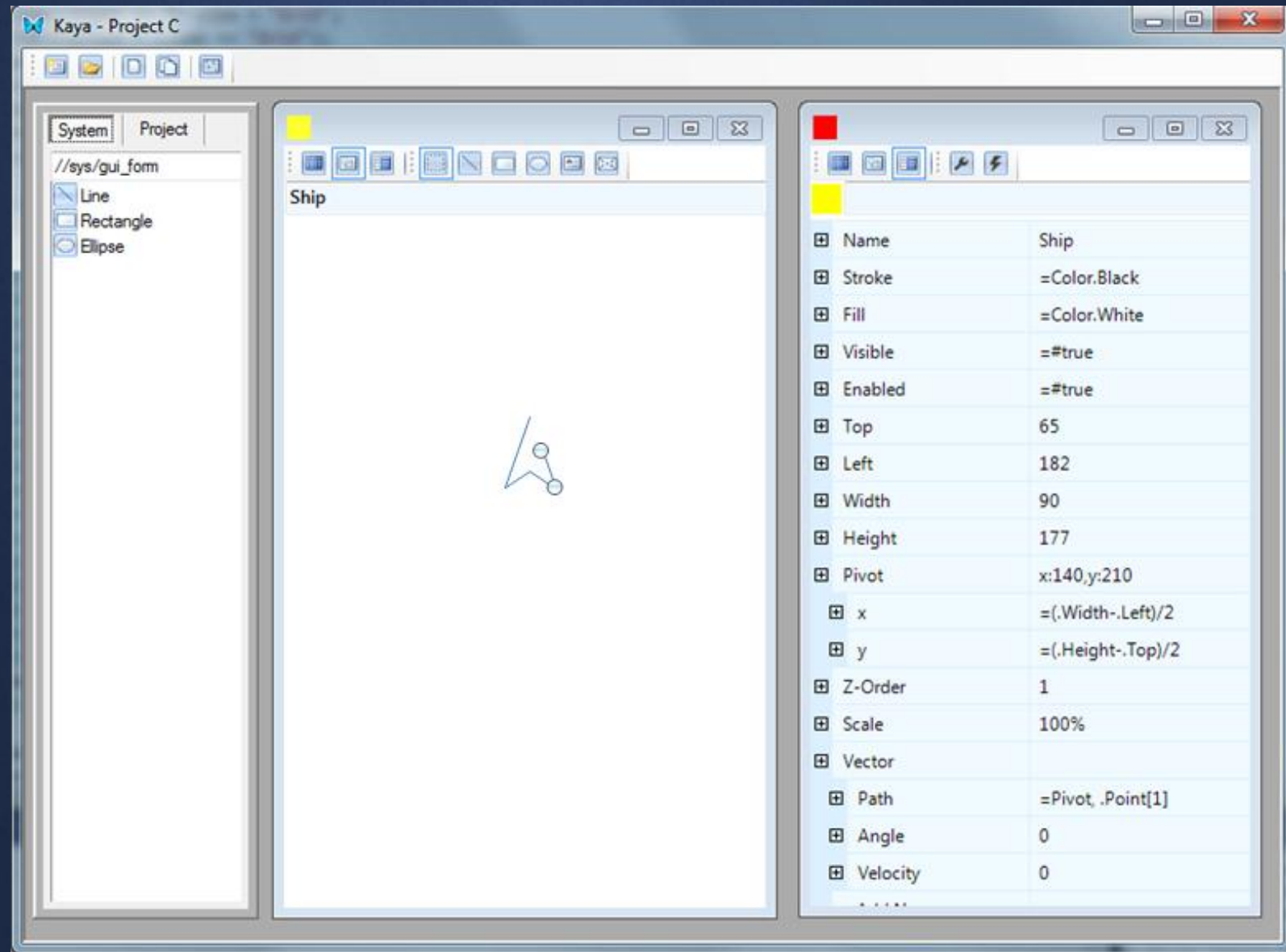


## Example

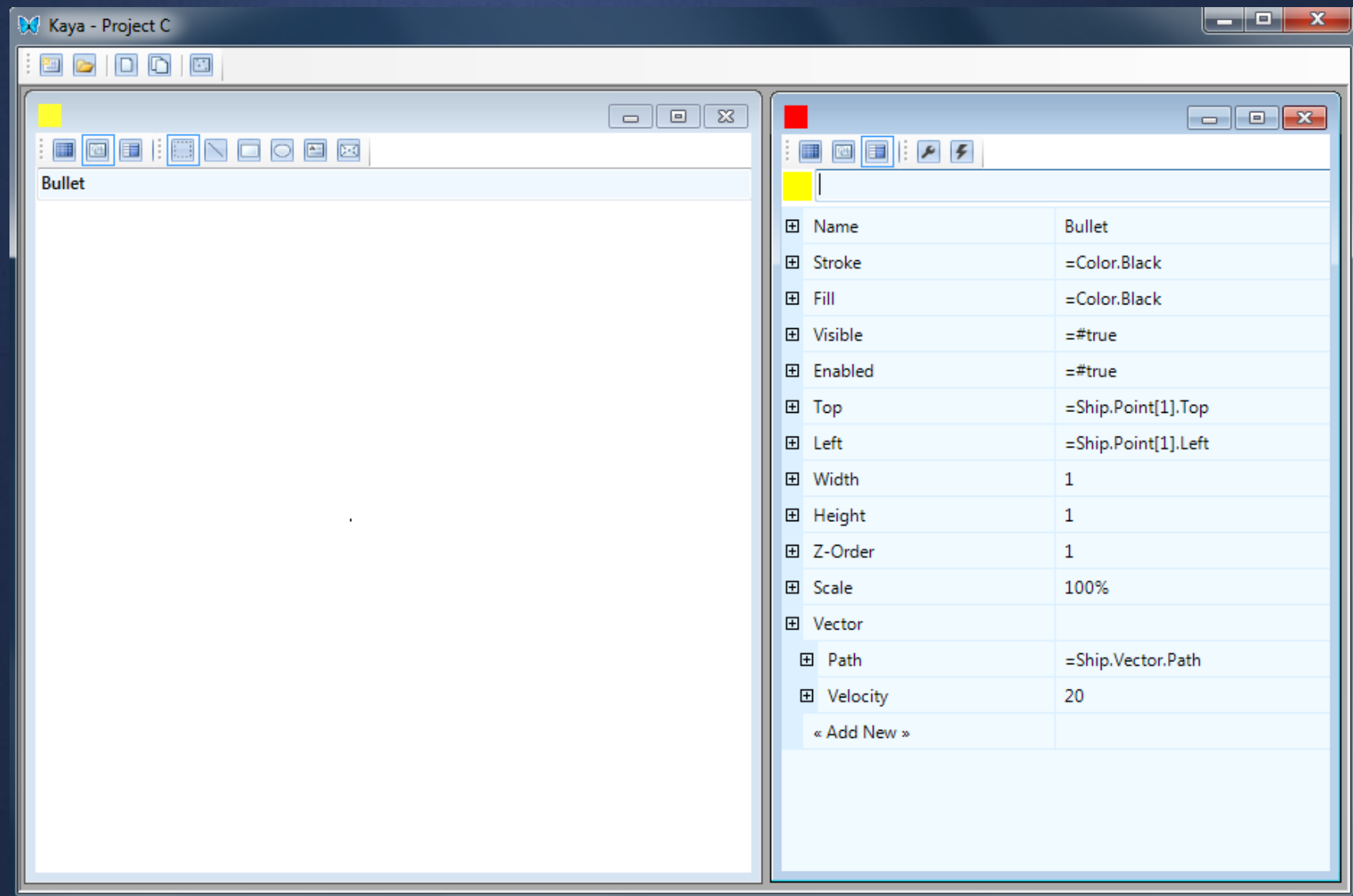




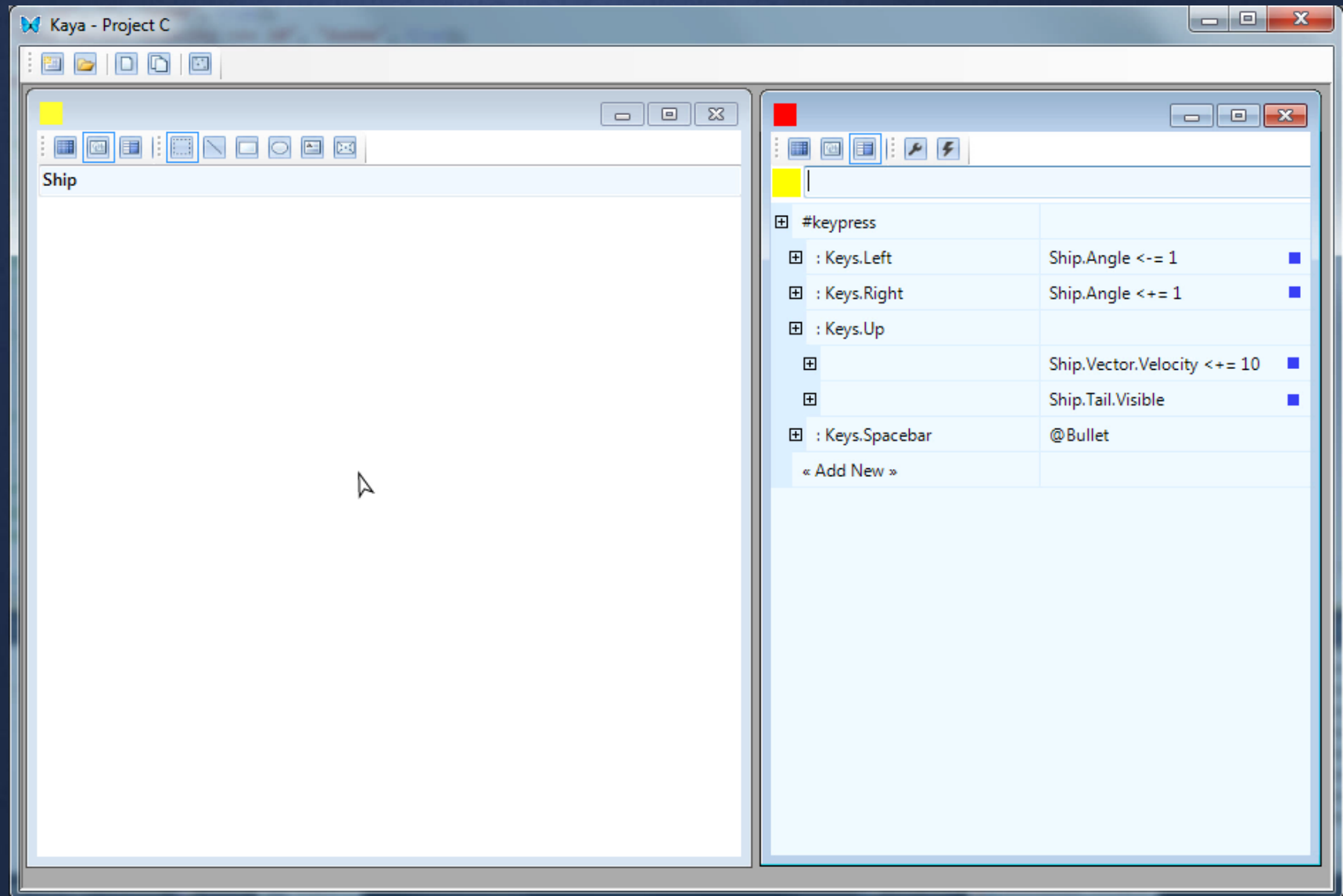
# Example



## Example



# Example



## Summary

- Declarative + Spreadsheet Metaphor  
= Usage dictates model
- Reactive, Immediately parsed + interpreted  
= Manage instructions the same as data  
= Syntax irrelevant, editor more important
- Homogenous universe of data/instructions  
= Simplicity of model

Thank You

[davidbroderick@yahoo.com](mailto:davidbroderick@yahoo.com)