

FLOGOL Integrated  
Development Environment  
FLOGOL Integrated  
Development Environment

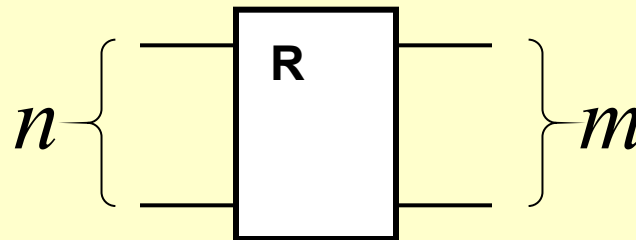
Система  
функционально-логического  
программирования  
на языке S-FLOGOL

# Язык S-FLOGOL

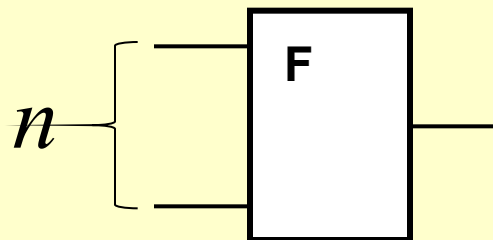
- Основан на теории направленных отношений (Фальк В.Н., Кутепов В.П.)
- Имеет развитые средства схемного описания направленных отношений
- Допускает использование индексированных имен отношений
- Поддерживает объектно-ориентированный стиль программирования
- Позволяет строить многомодульные программы
- Обладает средствами ограничения области видимости определяемых отношений (Private-домены)

# Семантические объекты

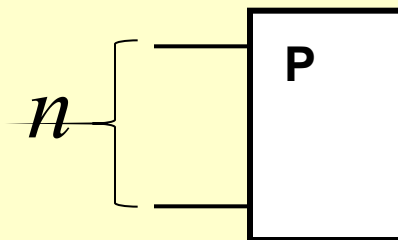
- Отношение  $R^{(n,m)}$



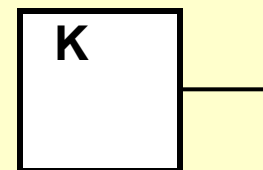
- Функция  $F^{(n,1)}$



- Предикат  $P^{(n,0)}$

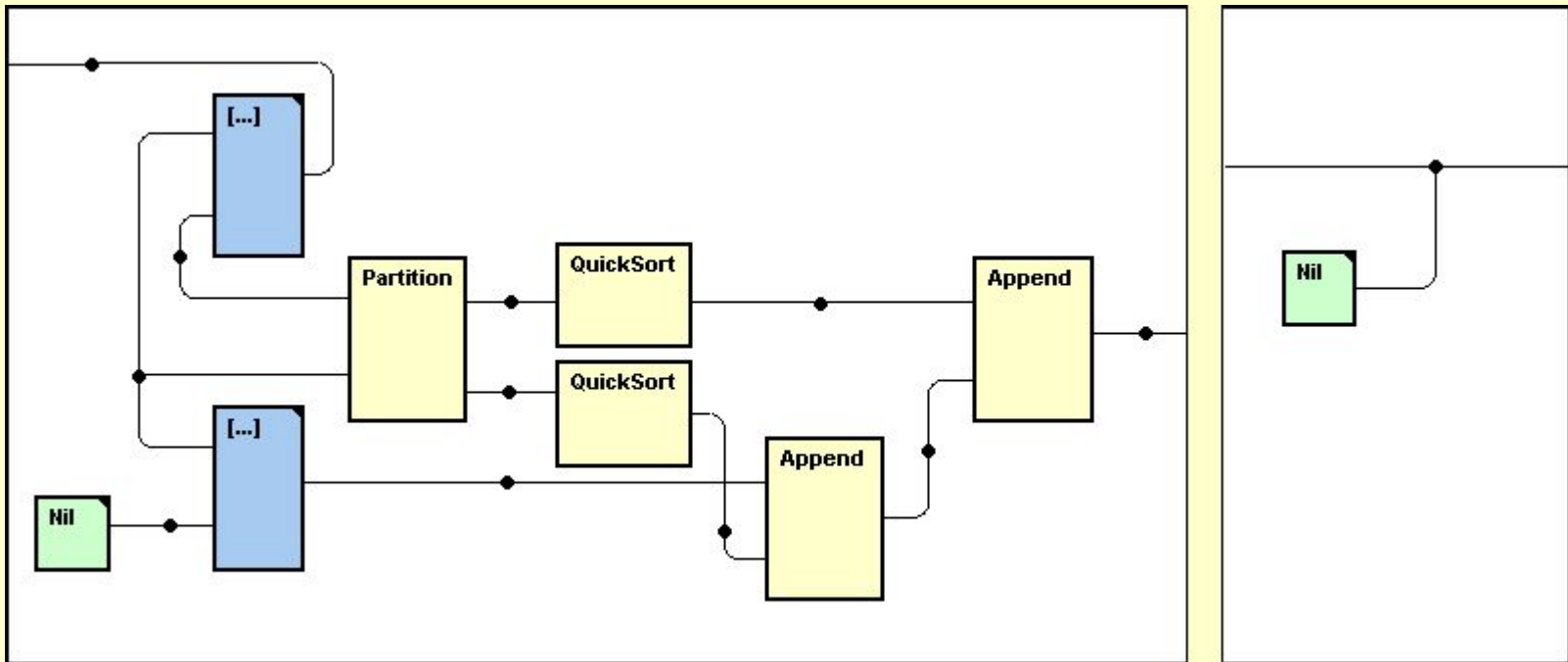


- Константа  $K^{(0,1)}$



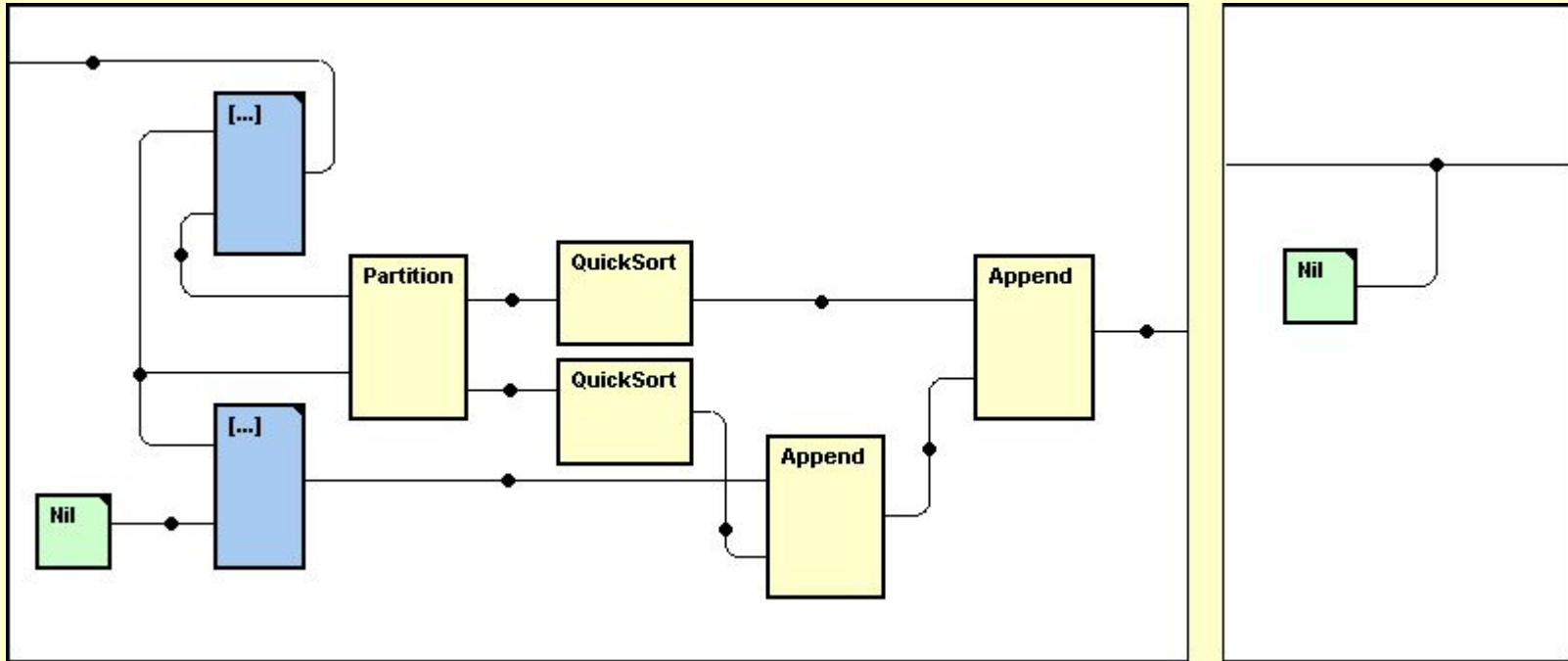
# Сетевое представление

Пример: быстрая сортировка  $QuickSort^{(1,1)}$



# Текстовое представление

Пример: быстрая сортировка  $QuickSort^{(1,1)}$



...

QuickSort=

```
{LCons(x,xs):Append(@ls,Append(LCons(x,Nil),@bs))
  ?Partition(xs,x)=ls,bs};
```

```
QuickSort={Nil:Nil};
```

# Композиции и свертки

Пример: определение натурального числа 3

- В форме графика:

$$\text{Nat3} = \{:\text{Succ}(\text{Succ}(\text{Succ}(\text{Null}))))\}$$

- В форме композиции:

$$\text{Nat3} = \text{Null} \cdot \text{Succ} \cdot \text{Succ} \cdot \text{Succ}$$

- При помощи свертки по последовательной композиции:

$$\text{Nat3} = \text{Null} \cdot (\cdot \text{I}=1..3)\text{Succ}$$

# Индексированные имена

Пример: множество натуральных чисел

```
[0]Nat= Null;  
(· I=1..)[I]Nat= Null·(· J=1..I)Succ
```

Эквивалентное множество определений:

```
[0]Nat= Null;  
[1]Nat= Null·Succ;  
[2]Nat= Null·Succ·Succ;  
[3]Nat= Null·Succ·Succ·Succ;  
...
```

# Среда программирования

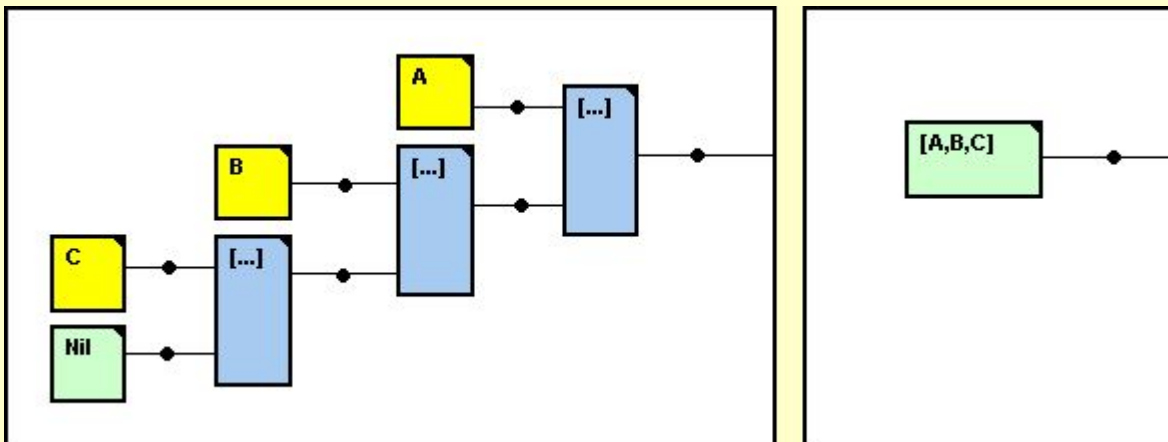
- Проектный подход к организации работы с комплексами программ
- Оригинальные технологии графического и структурированного текстового построения программ
- Интегрированные средства разработки и отладки
- Собственное сетевое ядро вычислений и компилятор запросов на языке S-FLOGOL
- Возможность ограниченного импорта программ, написанных на языке Пролог
- Реализация в среде C++ Builder 5.0.



# Системные типы данных

- Натуральные числа (0,1,2,...).
- Списки основных термов ([ ], [A(B),[D]]).
- Строки ('Some text').

Пример записи списка [A,B,C] в сетевой форме при помощи конструктора списка и системного генератора списков:



# Общий вид системы

The screenshot displays the FLIDE software interface, which is used for developing logic-based systems. It is divided into several main sections:

- Top Panel:** Contains the menu bar (Файл, Правка, Вид, Действия, Справка) and the title bar for the active window, "FLIDE - [Arithm.FLT]".
- Left Panel:** A tree view showing the project structure. It includes folders for "Конструкторы" (Constructors), "Правила" (Rules), and "Системные" (System). Under "Правила", there are sub-folders for "Add", "Mult", and "Power", each containing numbered items (e.g., Add.1, Add.2).
- Center Panel:** A text editor displaying the code for the "Arithm" module. The code defines various operations and their relationships:

```
MODULE Arithm=  
  (0:1)Null;  
  (1:1)Succ;  
  Nat3={:Succ(Succ(Succ(Null)))};  
  Nat2={:Succ(Succ(Null))};  
  Nat0={:Null};  
  Add={Null,x:x};  
  Add={Succ(x),y:Succ(@ (x,y))};  
  Mult={Null,x:Null};  
  Mult={Succ(x),y:Add(@ (x,y),y)};  
  Power={x,Null:Succ(Null)};  
  Power={x,Succ(y):Mult(x,@ (x,y))};  
  QAdd={:Add(Nat2,Nat3)};  
  Mult={:Mult(Nat2,Nat3)}.
```
- Right Panel:** A project browser showing a hierarchy of modules. It includes "Сетевые модули" (Network modules) like Arithm, ColorMap, Hanoi, Lists, Queens, and Triggers; "Дедуктивные модули" (Deductive modules) like Chang&Yong, Simple, and SteamRoller; and "Текстовые модули" (Textual modules) like Arithm, ColorMap, Hanoi, Lists, Queens, RelComp, and Triggers.
- Bottom Panel:** A diagram editor showing a flow graph. The graph consists of several nodes connected by lines. The nodes are labeled "Succ" (green), "Mult" (purple), and "Add" (yellow). The flow starts from a source node, goes through "Succ", then "Mult", and finally "Add".

# Графический редактор

The screenshot displays the FLIDE graphical editor interface. The title bar reads "FLIDE - [Hanoi.FLN]". The toolbar contains various icons for file operations, editing, and execution. The left sidebar shows a project tree with the following structure:

- Конструкторы
  - [...] (+2+:+1)
  - A (+0+:+1)
  - B (+0+:+1)
  - C (+0+:+1)
  - Nil (+0+:+1)
  - Null (+0+:+1)
  - Succ (+1+:+1)
  - to (+2+:+1)
- Константы
- Системные
- Правила
  - append.1
  - append.2
  - HANOI.1
  - HANOI.2**
- Запросы
  - Ханой(11)
  - Ханой(2)
- Результат
- Отладка

The main workspace shows a flowchart on a grid background. The flowchart consists of the following elements:

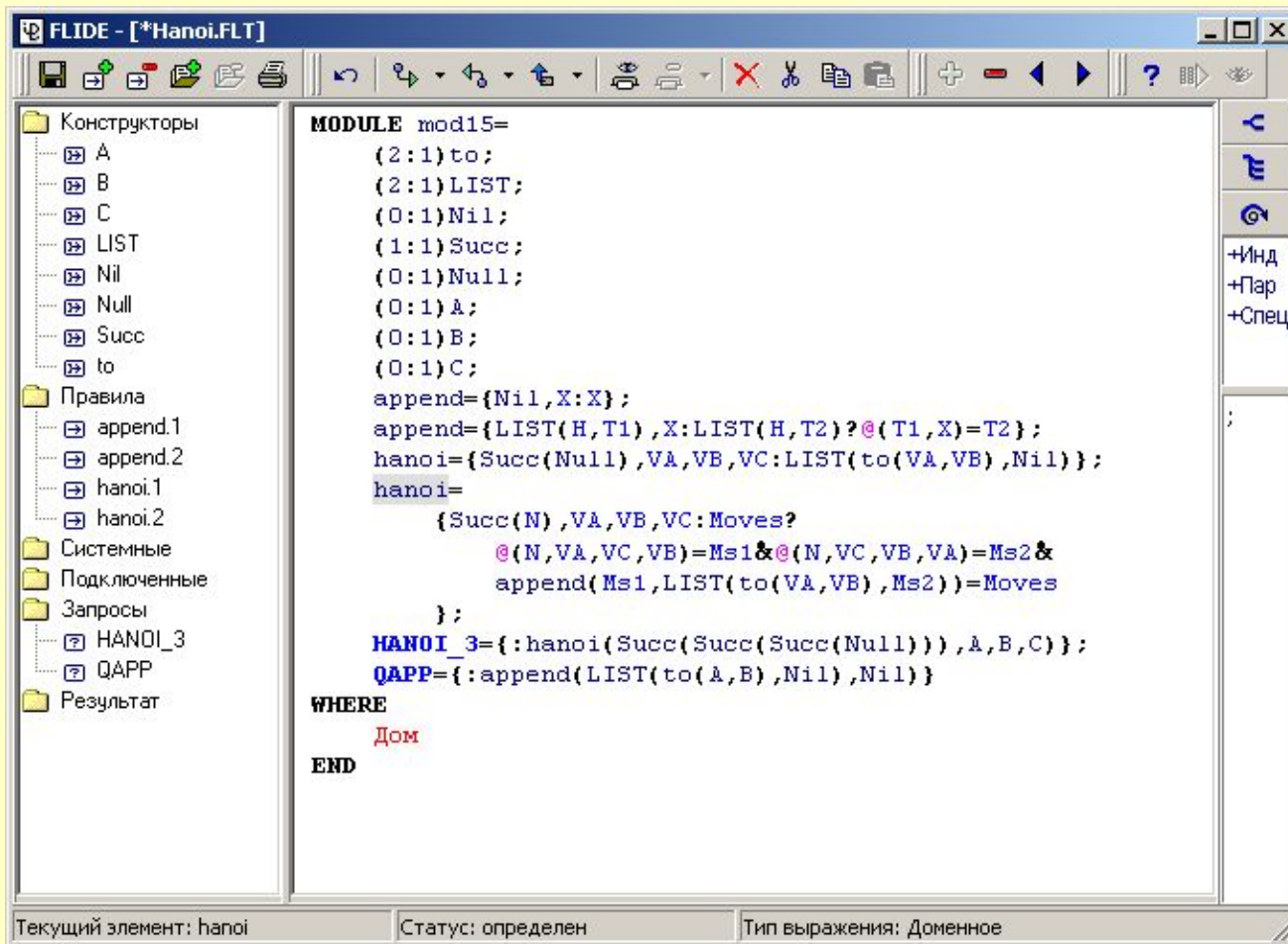
- A green box labeled "Succ" at the top left.
- Two pink boxes labeled "HANOI" with sub-labels "1" and "2" in the center.
- A blue box labeled "to" to the left of the "HANOI 1" box.
- A blue box labeled "[...]" to the right of the "HANOI 2" box.
- A yellow box labeled "append" with sub-label "3" on the right side.

Connections between boxes are made using black lines with circular nodes at the connection points. The flow starts from the left, goes through the "Succ" box, then splits to connect to both "HANOI 1" and "HANOI 2". From "HANOI 1", the flow goes to the "to" box. From "HANOI 2", the flow goes to the "[...]" box. Both the "to" and "[...]" boxes then connect to the "append" box. Finally, the "append" box connects to the right edge of the workspace.

The bottom status bar displays the following text:

Начало вычисления...  
Результат 1:  
Выход 1: [to(A,C),to(A,B),to(C,B)]  
Вычисление завершено за 0.005 секунд (9 шагов)

# Текстовый редактор



# Инструменты ввода

The screenshot displays the FLIDE software interface with the following components:

- Left Panel (Project Explorer):** A tree view showing a project structure with folders like "Конструкторы", "Правила", "Системные", "Подключенные", "Запросы", and "Результат".
- Center Panel (Code Editor):** Contains a code snippet for a module named "Arithm". The code defines various natural number types and operations like "Add" and "Mult". A context menu is open over the word "Терм" in the code, listing options such as "Переменная терма", "Анонимная переменная", "Вызов отношения", "Скрыть", "Свернуть", "Очистить", "Конструкция IF", "Конструкция CASE", and "Светка".
- Right Panel (Tool Palette):** A vertical toolbar with icons for "Списки", "IF", "Свертка", "Опции", and "Операции".
- Bottom Panel (Status Bar):** Shows "Текущий элемент: Терм", "Статус: не определен", and "Тип выражения: Терм".

Annotations with arrows point from text boxes to specific tools and code elements:

- "Списки" points to the list icon in the toolbar.
- "IF" points to the IF icon in the toolbar.
- "Свертка" points to the fold icon in the toolbar.
- "Опции" points to the options icon in the toolbar.
- "Операции" points to the operations icon in the toolbar.
- "Выбор альтернативы" points to the "Вызов отношения" option in the context menu.

# Ввод выбором альтернатив

Текст программы:

```
MODULE Common=  
  Дом  
END
```

Выбор альтернативы:

**Определение:**

Спец [СпИнд]ИдОтн[СпПар]=Рел

**Конструктор:**

Спец [СпИнд]ИдОтн

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]=Рел  
END
```

**Вызов отношения:**

ИмяОтн

**График:**

{Терм:Терм?Формула}

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]={Терм:Терм?Формула}  
END
```

# Ввод идентификаторов

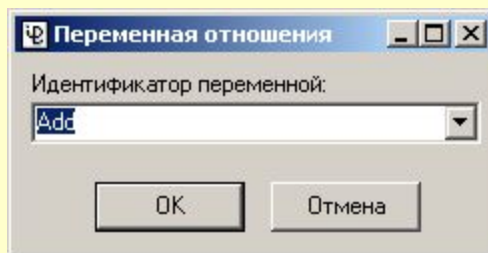
Текст программы:

```
MODULE Common=  
  Спец [СпИнд]ИдОтн[СпПар]=Рел  
END
```

Действие:

*Раскрыть:*  
Текст

Форма ввода:

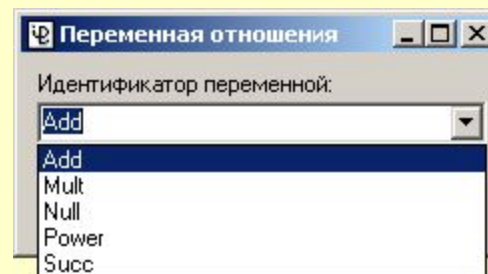


Переменная отношения

Идентификатор переменной:

Add

OK Отмена



Переменная отношения

Идентификатор переменной:

Add

Mult

Null

Power

Succ

```
MODULE Common=  
  Спец [СпИнд]Add[СпПар]=Рел  
END
```

# Ввод выражений: операции

Текст программы:

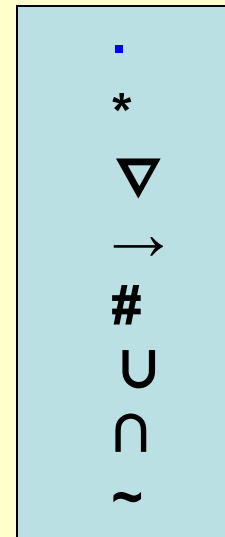
```
MODULE Common=  
  (0:1)[1]Nat=Рел  
END
```

```
MODULE Common=  
  (0:1)[1]Nat=Рел·Рел  
END
```

...

```
MODULE Common=  
  (0:1)[1]Nat=Null·Succ  
END
```

Операция:





# Ввод выражений: IF, Свертка.

Текст программы:

```
MODULE Common=  
  (0:1)R=Succ  
END
```

Действие:

```
MODULE Common=  
  (0:1)R=IF Лог THEN Рел ELSE Succ  
END
```

Условная  
конструкция IF

Кнопка:



```
MODULE Common=  
  (0:1)R=(ИдСв=СпЗнач)Succ  
END
```

Свертка

Кнопка:



# Автоструктурирование текста

