Программный модуль интерпретатора сценариев для системы моделирования

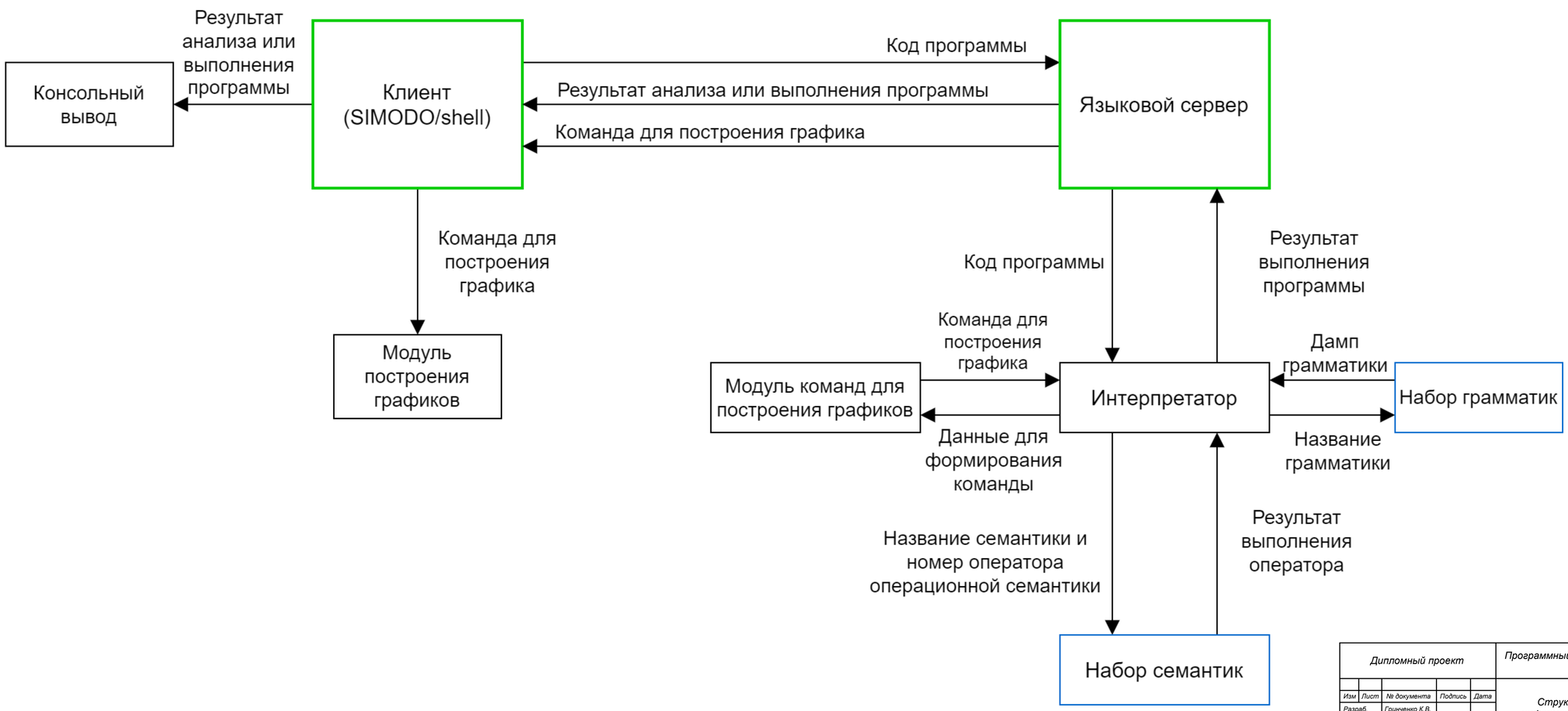
- **В адаптивной системе для имитационного моделирования комплексных систем (SIMODO)** реализована методика формирования предметно-ориентированных языков (ПОЯ)

- **Адаптивность** – приспособление под конкретную предметную область с помощью ПОЯ

- **Цель**: в качестве **демонстрации методики** выбрана цель реализовать подмножество языка описания и верификации аппаратуры **SystemVerilog**

- Для определния подмножества языка использовались задачи, которые даются в домашней работе по дисциплине **"Основы проектирования устройств ЭВМ"**

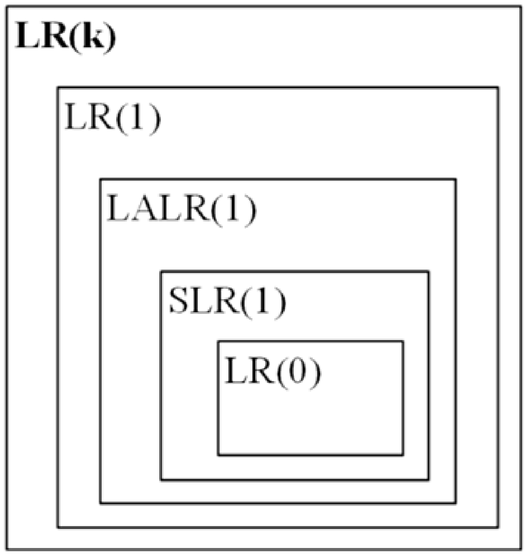
Структурная схема информационной системы



Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования		
						Литер.	Масса	Масштаб
Изм	Лист	№ документа	Подпись	Дата	Структурная схема информационной системы			
Разраб.		Гринченко К.В.				Лист 1	Листов 9	
Руковод.		Фомин М.М.						
Н. контр.		Ерёмин О.Ю.			МГТУ им. Н.Э. Баумана Группа ИУ6-82Б			

Анализ алгоритмов синтаксического разбора формальных грамматик

- Иерархия подклассов контекстно-свободных грамматик



- Правила для исследования

```
A -> A0B|B|a
B -> B1C|C|b|{A}
C -> C2D|D|c|[B]
D -> D3E|E|d|(C)
E -> E4F|F|f|{D}
F -> F5G|G|g|[E]
G -> G6H|H|h|(F)
H -> H7I|I|i|{G}
I -> I8J|J|j|[H]
...
```

- Зависимость роста количества состояний автомата для синтаксического анализатора от количества взятых строк правил

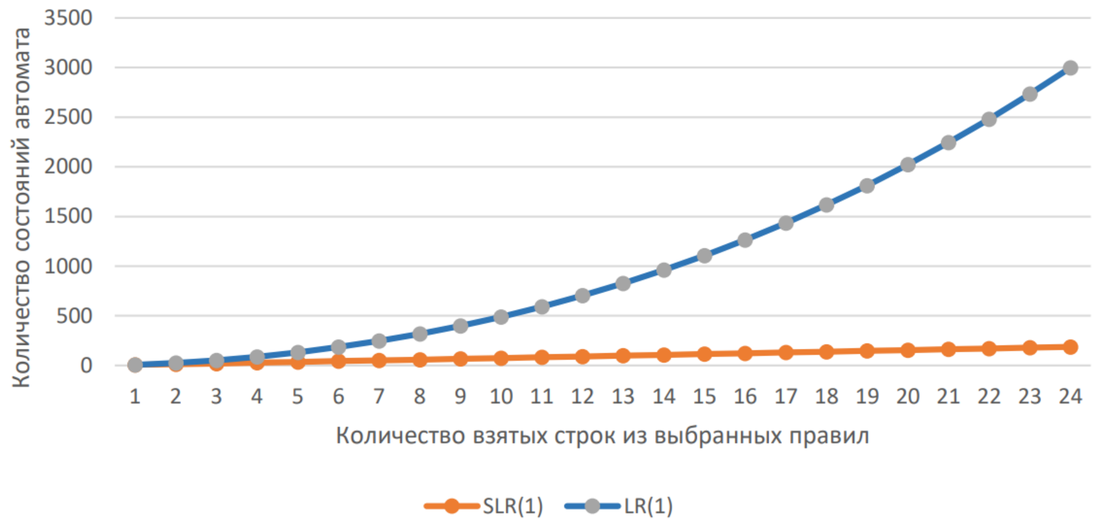
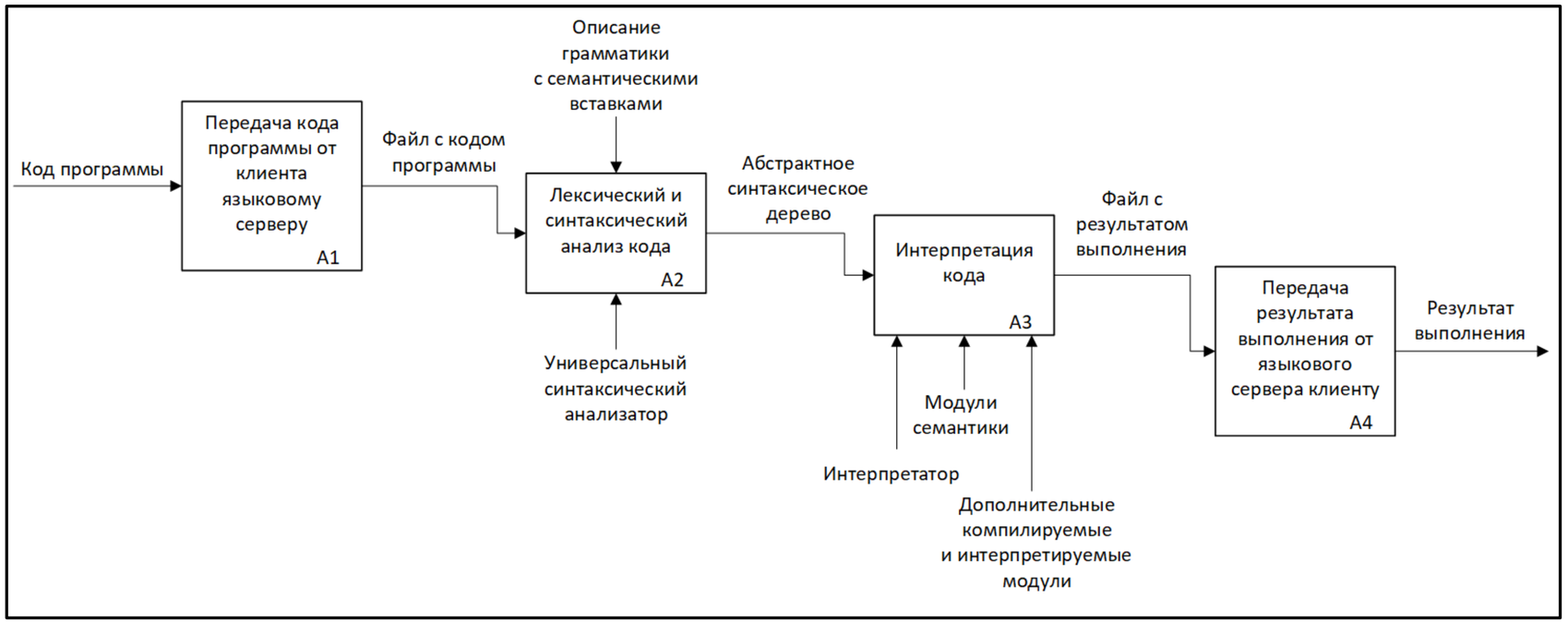
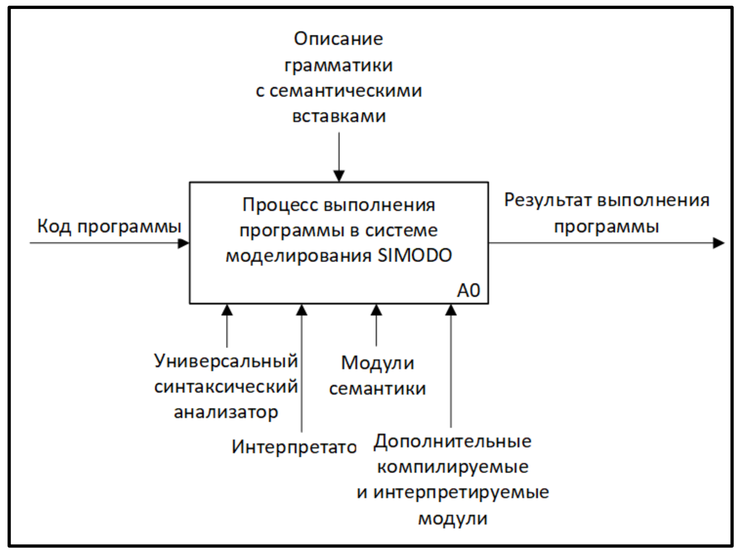


Диаграмма декомпозиции контекстной диаграммы



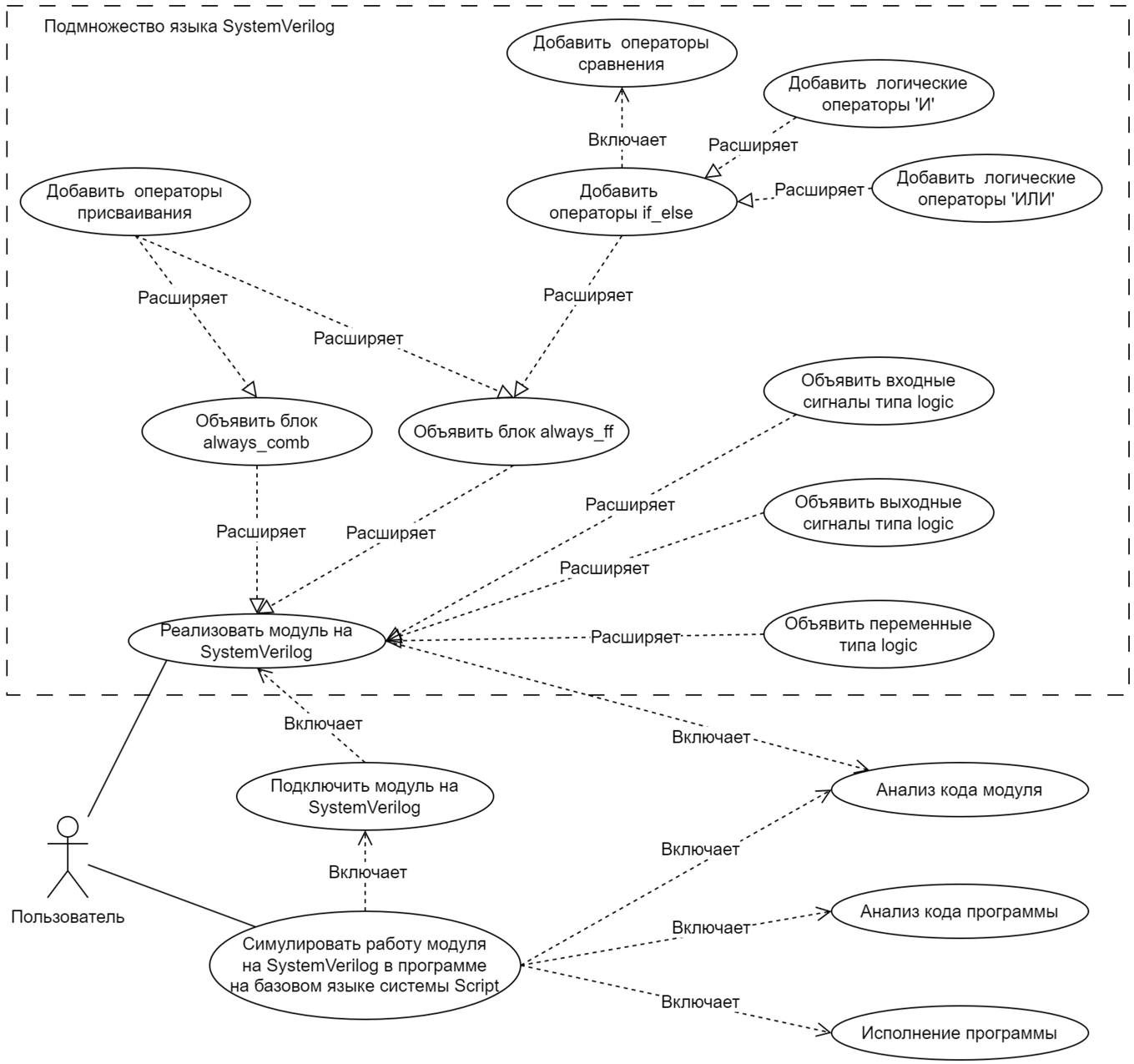
Контекстная диаграмма



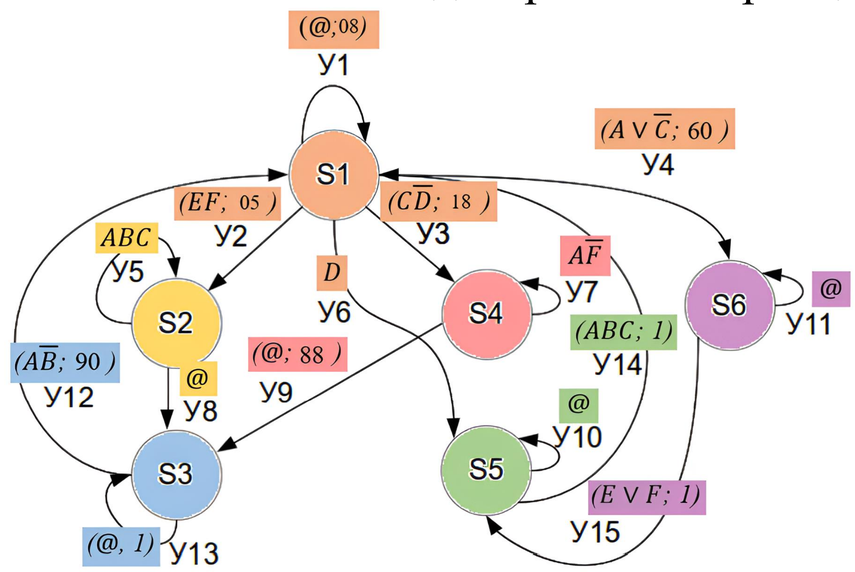
Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования	
Изм	Лист	№ документа	Подпись	Дата		
Разраб.		Гринченко К.В.			Функциональная диаграмма	Литер. / Масса / Масштаб
Руковод.		Фомин М.М.				Лист 2 / Листов 9
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б

Подмножество языка SystemVerilog

Диаграмма вариантов использования



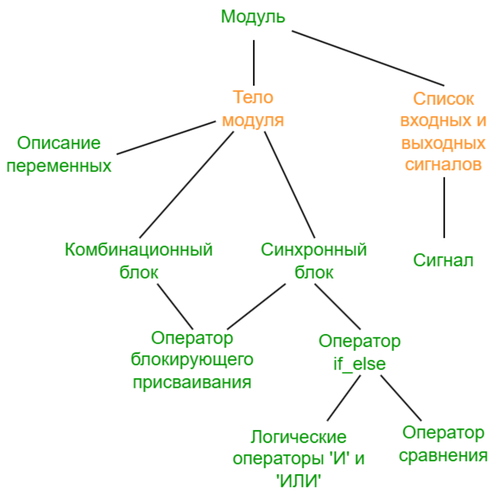
- Разработать и протестировать описание управляющего автомата в соответствии с диаграммой переходов



- Часть кода описания автомата

```
module fsm(
  input logic clk,
  input logic rst,
  input logic [5:0] c,
  output logic [2:0] st,
  output logic [7:0] num
);
  logic [2:0] State;
  always_comb
    st = State;
  always_ff @ (posedge clk, negedge rst)
  begin
    if(rst == 0) begin
      num = 0;
      State = 1;
    end
    else if(State == 1) begin
      if(c[4] == 1 && c[5] == 1) begin
        num = 'b00000101;
        State = 2;
      end
      else if(c[2] == 1 && c[3] == 0) begin
        num = 'b00011000;
```

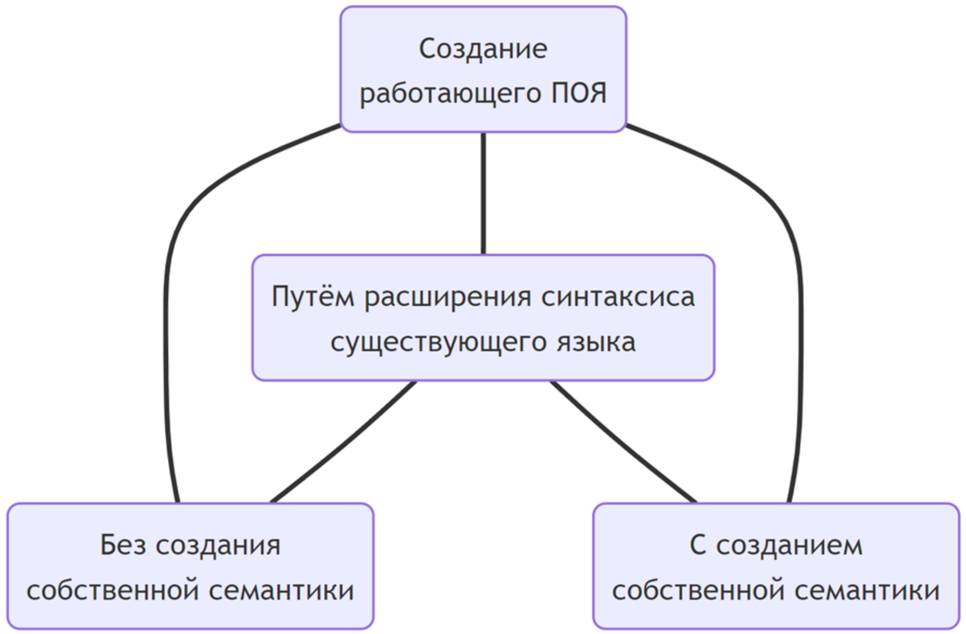
- Выбранное подмножество языка SystemVerilog



Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования			
Изм	Лист	№ документа	Подпись	Дата	Диаграмма вариантов использования	Литер.	Масса	Масштаб
Разраб.		Гринченко К.В.						
Руковод.		Фомин М.М.				Лист 3	Листов 9	
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б		

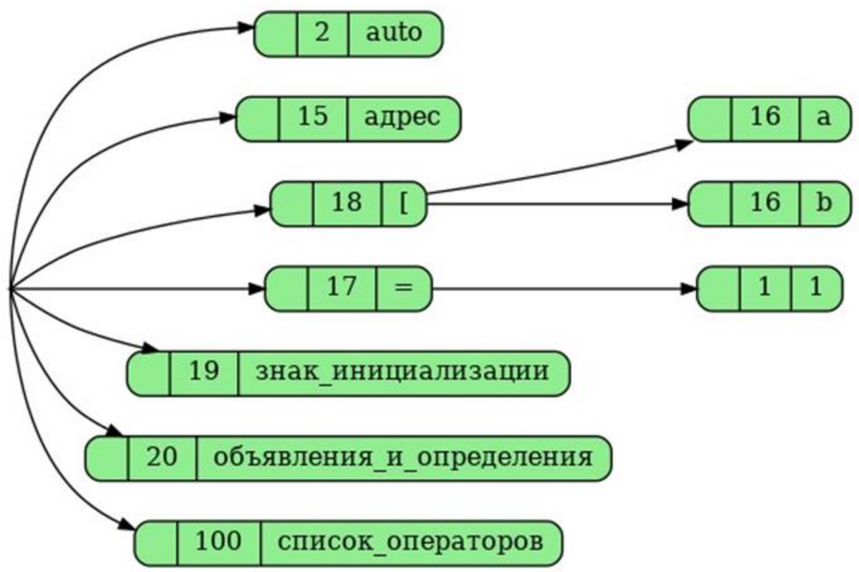
Этапы создания ПОЯ

• Схема вариантов создания работающего ПОЯ



• Пример программы на Script и построенного по ней абстрактного дерева операционной семантики (АДОС)

```
def auto : [ a, b ] = 1
```



Выбран вариант **с созданием собственной семантики**. Этот вариант предполагает следующие шаги:

- описание **лексики**;
- описание **синтаксиса**;
- настройка подсветки лексики;
- добавление настроек для данного ПОЯ в список языков оболочки SIMODO.
- добавление **семантики**:
  - добавление в грамматику вставок операционной семантики, содержащих короткое название семантики и символьное обозначение оператора;
  - реализация на языке программирования C++ своего семантического модуля, в котором реализованы ООС;
  - при реализации этого модуля важно максимально использовать существующие функции интерпретатора;

Структурная схема модуля ПОЯ

Дипломный проект						Программный модуль интерпретатора сценариев для системы моделирования	
							Литер. / Масса / Масштаб
Изм	Лист	№ документа	Подпись	Дата		Структурная схема	
Разраб.		Гринченко К.В.					Лист 4 / Листов 9
Руковод.		Фомин М.М.					
Н. контр.		Ерёмин О.Ю.					МГТУ им. Н.Э. Баумана Группа ИУ6-82Б

Описание грамматики разрабатываемого ПОЯ

• Часть описания синтаксиса

```
42  направление_сигнала
43      = "input"
44      = "output"
45      ;
46
47  тип_сигнала
48      = "logic"                   {
49                                      ast.addNode(ast.sbl.host, ast.sbl.op.PushVariable, "array");
50                                      ast.addNode(ast.sbl.host, ast.sbl.op.ObjectElement, "int1");
51                                      ast.addNode(ast.sbl.host, ast.sbl.op.FunctionCall, 0);
52                                      ast.addNode(ast.sbl.host, ast.sbl.op.Announcement, 0);
53                                  }
54      = "logic" начало_упаковки NUMBER ":" NUMBER "]"
55                                  {
56                                      ast.addNode(ast.sbl.host, ast.sbl.op.PushConstant, 2);
57                                      ast.goParent();
58                                      ast.addNode(ast.sbl.host, ast.sbl.op.Announcement, 0);
59                                  }
60      ;
61
62  начало_упаковки
63      = "["                       {
64                                      ast.addNode(ast.sbl.host, ast.sbl.op.PushVariable, "array");
65                                      ast.addNode(ast.sbl.host, ast.sbl.op.ObjectElement, "logic");
66                                      ast.addNode_StepInto(ast.sbl.host, ast.sbl.op.FunctionCall, 0);
67                                  }
68      ;
69
```

• Лексические вставки

```
14  lex {
15      lex.addMarkup("/*", "*/", "", lex.LexemeType.Comment);
16      lex.addMarkup("//", "", "", lex.LexemeType.Comment);
17      lex.addMarkup("\"", "\"", "\\", lex.LexemeType.Annotation);
18
19      lex.clearMasks();
20      lex.addMask("'bN", lex.LexemeType.Number, 2);
21      lex.addMask("'oN", lex.LexemeType.Number, 8);
22      lex.addMask("'hN", lex.LexemeType.Number, 16);
23      lex.addMask("@BUILDING_NUMBER_MASK", lex.LexemeType.Number, 10);
24  }
```

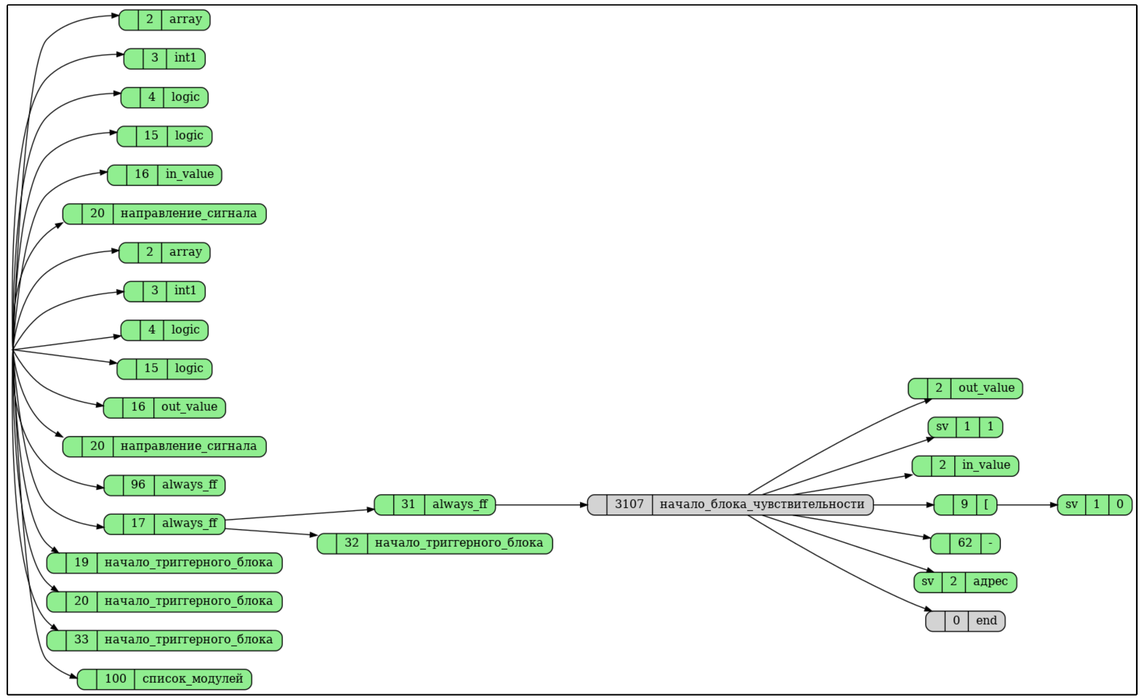
• Пример модуля

```
1  module ex(
2    input logic in_value,
3    output logic out_value
4  );
5    always_ff @ (posedge in_value)
6    begin
7      out_value = 1 - in_value[0];
8    end
9  endmodule
```

• Данные для подсветки лексики

```
1  {
2      "description" : "Структура LexicalParameters для SIMODO fuze",
3      "markups" : [
4          {
5              "start"       : "/*",
6              "end"         : "*/",
7              "ignore_sign" : "",
8              "type"        : "Comment"
9          },
10         {
11             "start"       : "//",
12             "end"         : "",
13             "ignore_sign" : "",
14             "type"        : "Comment"
15         },
16         {
17             "start"       : "\"",
18             "end"         : "\"",
19             "ignore_sign" : "\\",
20             "type"        : "Annotation"
21         }
22     ],
23     "masks" : [
24         {"chars" : "'bN", "type" : "Number", "system" : 2},
25         {"chars" : "'oN", "type" : "Number", "system" : 8},
26         {"chars" : "'N", "type" : "Number", "system" : 16},
27         {"chars" : "@BUILDING_NUMBER_MASK", "type" : "Number", "system" : 10}
28     ],
29     "punctuation_chars" : "+-,;(){}[].><=*/&#!|:?~%",
30     "punctuation_words" : ["if", "else", "for", "module", "begin", "end",
31                            "always_comb", "always_ff", "endmodule", "assign"],
32     "digits"            : "0123456789ABCDEF",
33     "latin_alphabet"    : "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ",
34     "national_alphabet" : "абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ",
35     "id_extra_symbols"  : "_@",
36     "may_national_letters_use" : false,
37     "may_national_letters_mix" : false,
38     "is_case_sensitive" : true
39 }
```

• Пример АДОС построенного по грамматике SystemVerilog



Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования		
Изм	Лист	№ документа	Подпись	Дата	Описание грамматики	Литер.	Масса / Масштаб
Разраб.		Гриченко К.В.					
Руковод.		Фомин М.М.				Лист 5	Листов 9
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б	

Семантический модуль

Диаграмма классов уровня спецификации

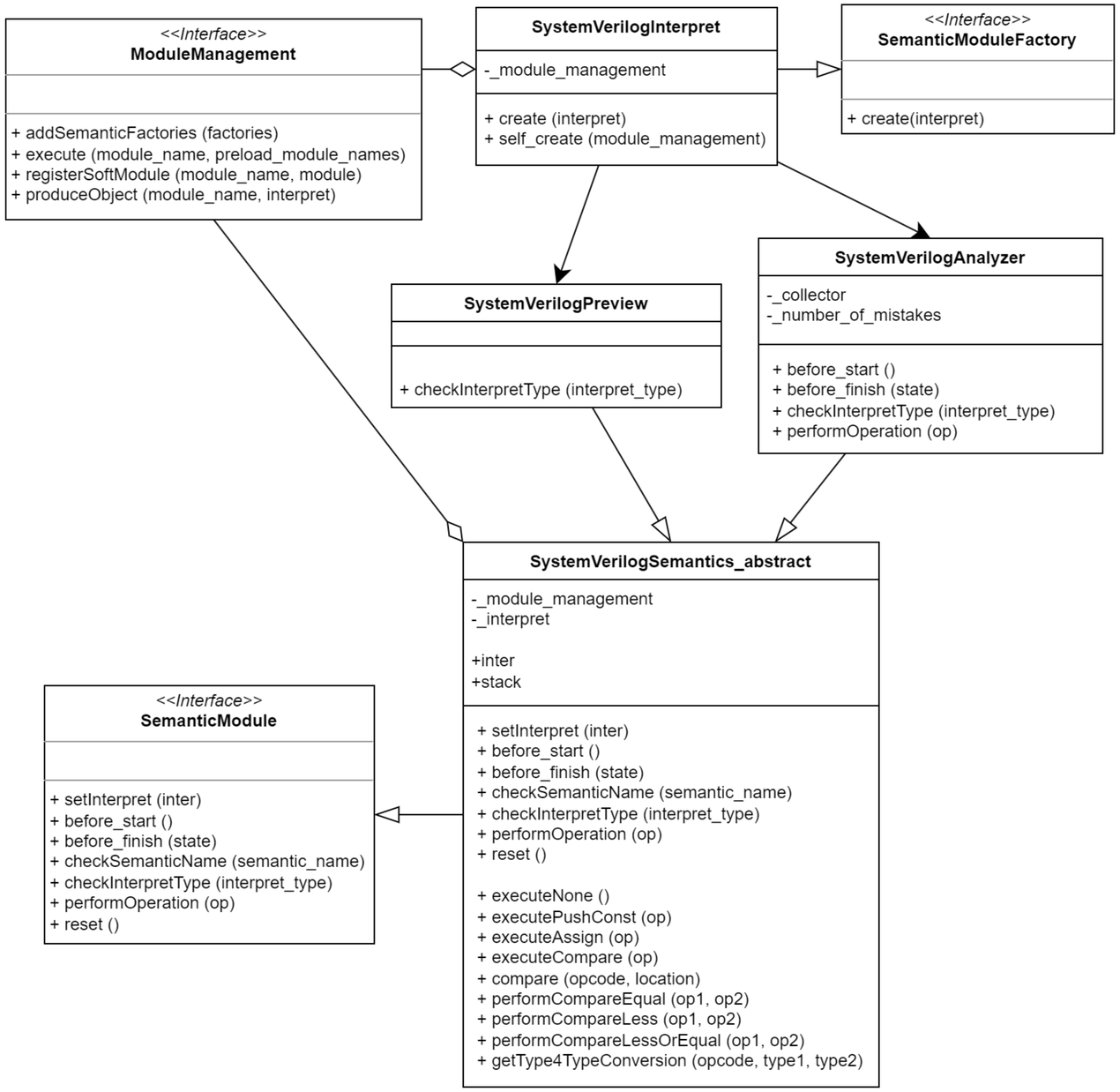
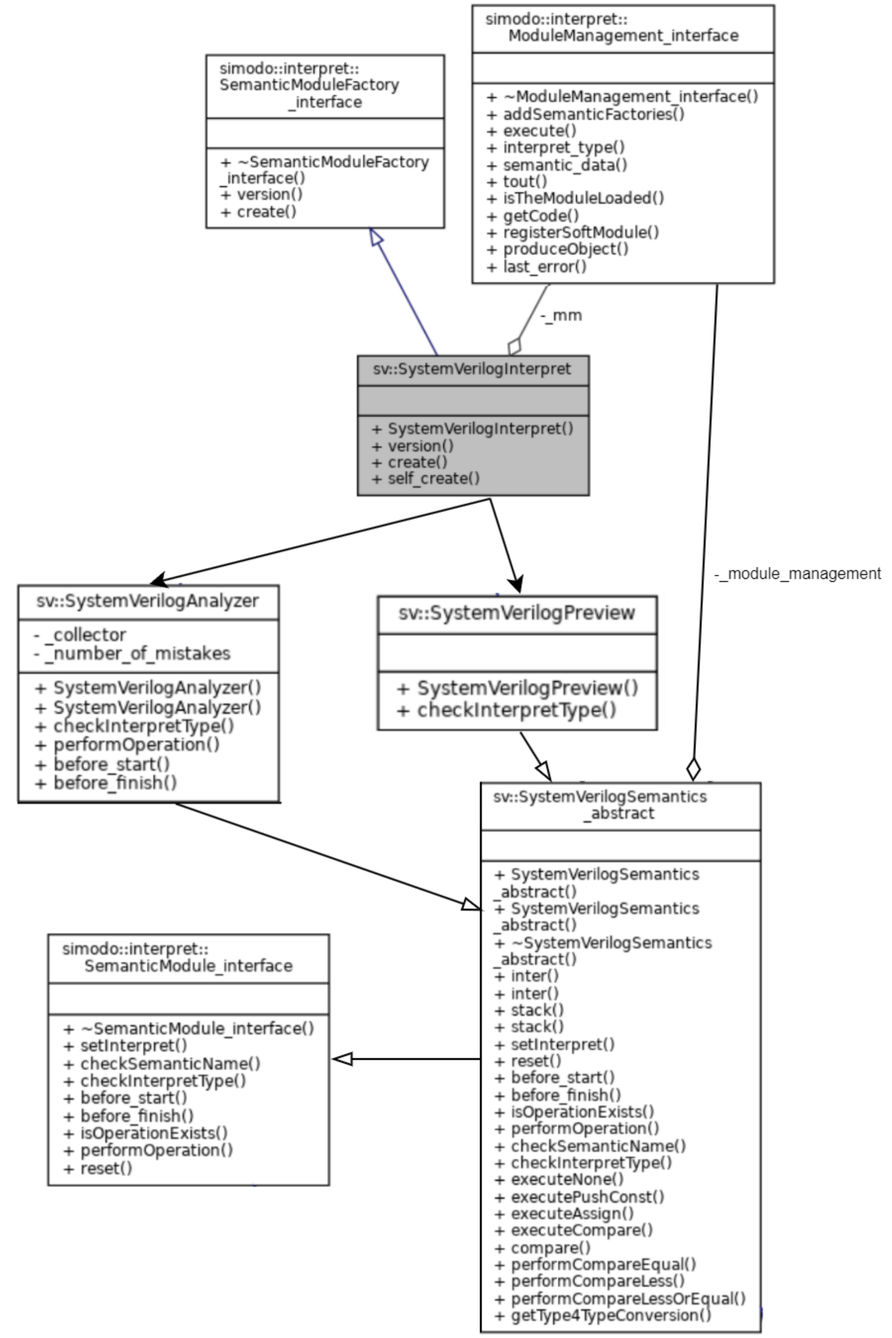


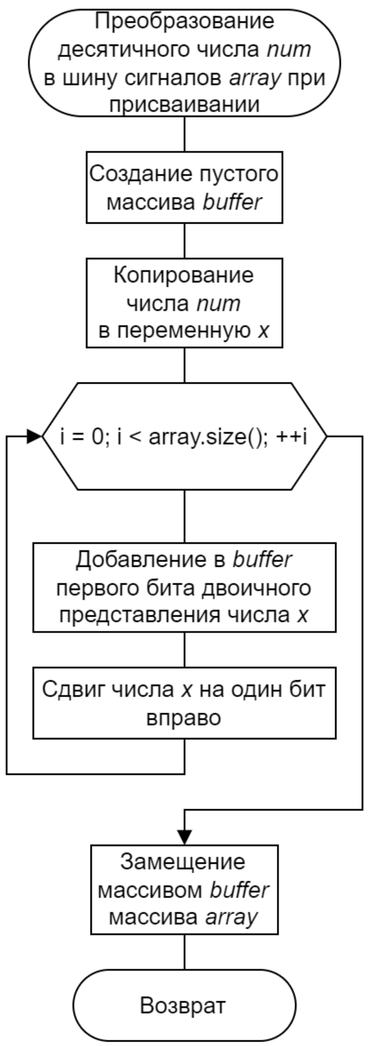
Диаграмма классов уровня реализации



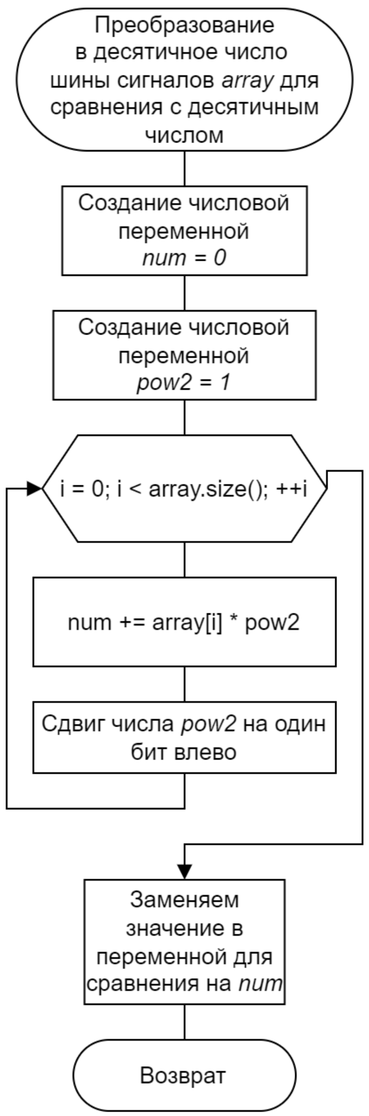
Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования		
Изм	Лист	№ документа	Подпись	Дата	Диаграммы классов	Литер.	Масса / Масштаб
Разраб.		Гринченко К.В.					
Руковод.		Фомин М.М.				Лист 6	Листов 9
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б	

Схемы алгоритмов для модификации семантических операторов

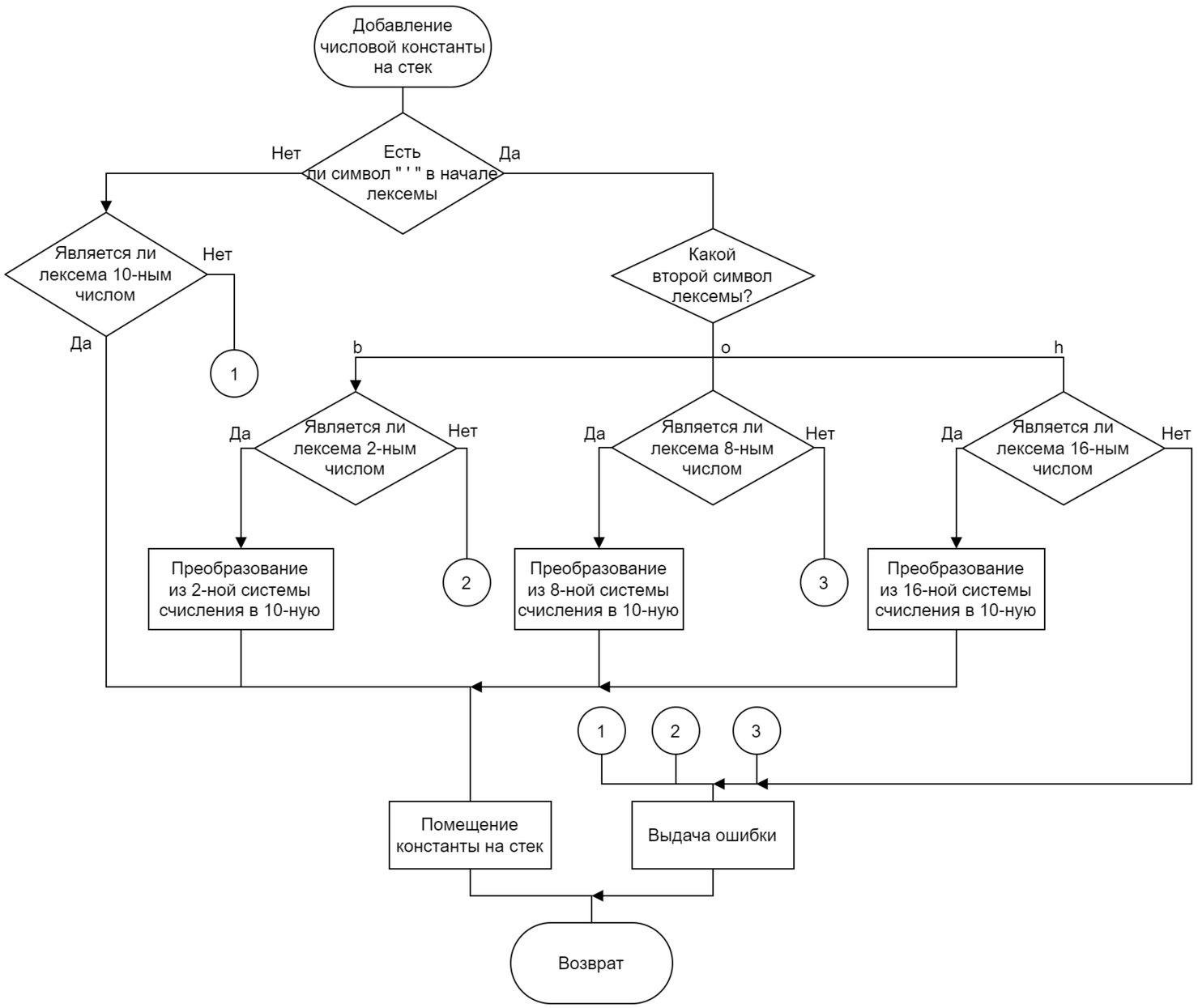
Оператор Assign



Оператор Compare



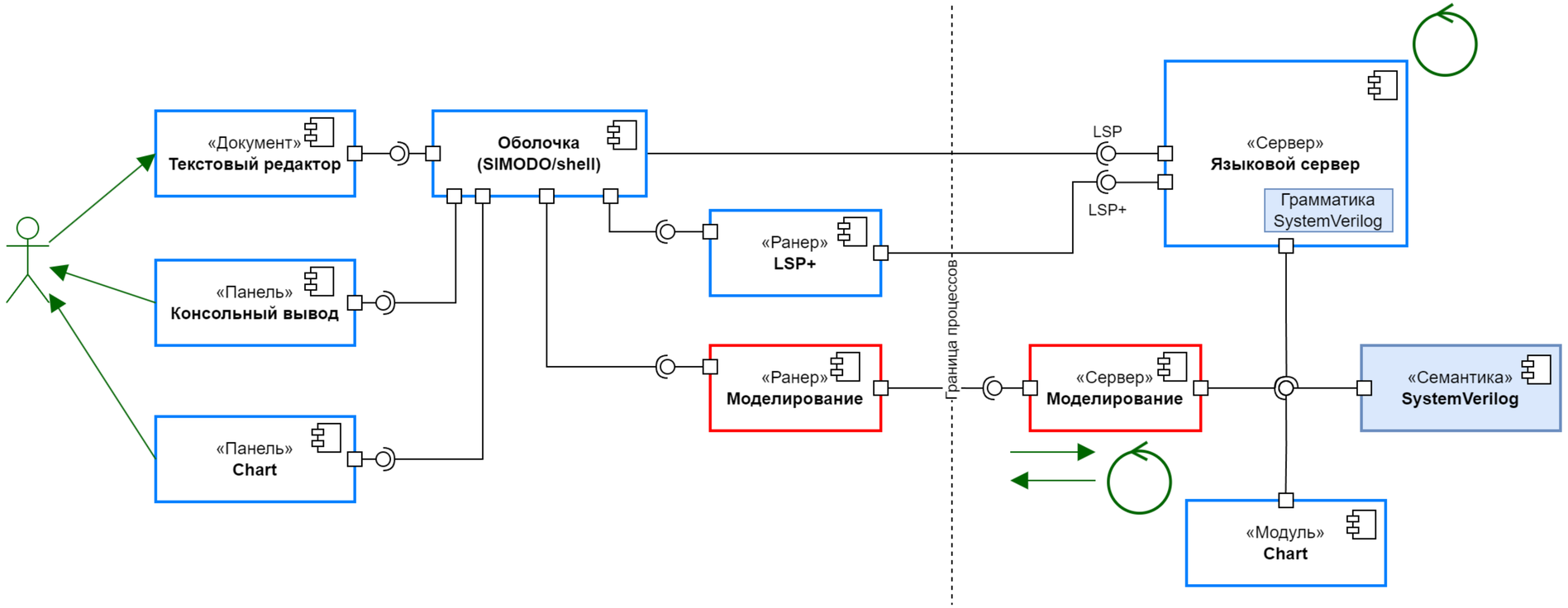
Оператор PushConst



Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования		
Изм	Лист	№ документа	Подпись	Дата	Схемы алгоритмов	Литер.	Масса / Масштаб
Разраб.		Гринченко К.В.					
Руковод.		Фомин М.М.				Лист 7	Листов 9
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б	

Итоги реализации

Диаграмма компонентов



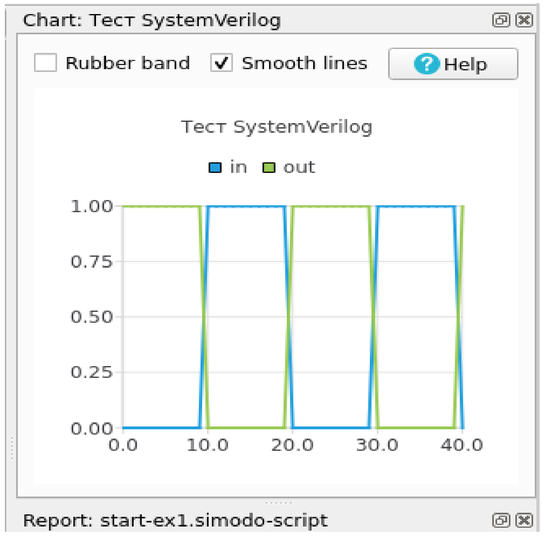
- Пример модуля

```
module ex(
  input logic in_value,
  output logic out_value
);
  always_ff @ (posedge in_value)
  begin
    out_value = 1 - in_value[0];
  end
endmodule
```

- Пример программы на базовом языке Script для симуляции работы модуля

```
import "modules/ex-01.sv" type Ex : ex // модуль для симуляции

import "/data/modules/chart.simodo-script" : chart // модуль для работы с графиками

chart.init("Тест SystemVerilog") // инициализация панели с графиком

def int :
    ff_step   = 10, // шаг симуляции в условных единицах времени
    steps     = 4   // количество шагов, которое будет симулироваться

for int : comb_step in Range(0,steps*ff_step)
{
    if (comb_step % ff_step == 0) // условие для вызова блока always_ff
    {
        ex.in_value = ex.out_value // изменение входного сигнала

        ex.always_ff() // вызов блока always_ff
    }

    // добавление следующей точки на график входного сигнала
    chart.addPoint("in", comb_step, ex.in_value[0])
    // добавление следующей точки на график выходного сигнала
    chart.addPoint("out", comb_step, ex.out_value[0])
}

chart.show() // отображение получившихся графиков
```

- Результат симуляции



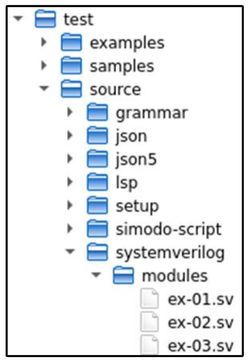
Дипломный проект					Программный модуль интерпретатора сценариев для системы моделирования	
Изм	Лист	№ документа	Подпись	Дата	Диаграмма компоновки программных компонентов	Литер. / Масса / Масштаб
Разраб.		Гринченко К.В.				
Руковод.		Фомин М.М.				Лист 8 / Листов 9
Н. контр.		Ерёмин О.Ю.				МГТУ им. Н.Э. Баумана Группа ИУ6-82Б

Регрессионное функциональное тестирование

Автотесты

Сценарий непрерывной интеграции (GitLab CI)

Набор модулей



Программы для симуляции

```
import "modules/ex-01.sv" type Ex : ex // модуль для симуляции

import "/data/modules/chart.simodo-script" : chart // модуль для работы с графиками

chart.init("Тест SystemVerilog") // инициализация панели с графиком

def int :
    ff_step   = 10, // шаг симуляции в условных единицах времени
    steps     = 4   // количество шагов, которое будет симулироваться

for int : comb_step in Range(0,steps*ff_step)
{
    if (comb_step % ff_step == 0) // условие для вызова блока always_ff
    {
        ex.in_value = ex.out_value // изменение входного сигнала

        ex.always_ff() // вызов блока always_ff
    }

    // добавление следующей точки на график входного сигнала
    chart.addPoint("in", comb_step, ex.in_value[0])
    // добавление следующей точки на график выходного сигнала
    chart.addPoint("out", comb_step, ex.out_value[0])
}

chart.show() // отображение получившихся графиков
```

Результаты выполнения тестов

```
Тест работы интерпретатора SystemVerilog ========================
--- test/source/systemverilog/start-ex1.simodo-script :
import "modules/ex-01.sv" type Ex : ex

import "/data/modules/chart.simodo-script" : chart

chart.init("Тест SystemVerilog")

def int :
    ff_step   = 10,
    steps     = 4

for int : comb_step in Range(0,steps*ff_step)
{
    if (comb_step % ff_step == 0)
    {
        ex.in_value = ex.out_value

        ex.always_ff()
    }

    chart.addPoint("in", comb_step, ex.in_value[0])
    chart.addPoint("out", comb_step, ex.out_value[0])
}

chart.show()
--- interpret :
#Values:Chart.0.S.init:Тест SystemVerilog
#Values:Chart.0.U.addPoint:in/0.0/0.0
#Values:Chart.0.U.addPoint:out/0.0/1.0
#Values:Chart.0.U.addPoint:in/1.0/0.0
```

Сравнение результата с ожидаемым

```
user@LAPTOP-MM71TGHH:/mnt/d/PrevPC/Desktop/BMSTU/Diplom/SIMODO/Project/shell$ diff -Z test/tmp/test-interpret-systemverilog.out test/samples/test-interpret-systemverilog.out --color
3c3
< import "modules/ex-01.sv" type Ex : ex // модуль для симуляции
---
> import "modules/ex-01.sv" type Ex : ex
5c5
< import "/data/modules/chart.simodo-script" : chart // модуль для работы с графиками
---
> import "/data/modules/chart.simodo-script" : chart
7c7
< chart.init("Тест SystemVerilog") // инициализация панели с графиком
---
> chart.init("Тест SystemVerilog")
10,11c10,11
```

Сравнение результата с ожидаемым (GitLab CI)

```
user@LAPTOP-MM71TGHH:/mnt/d/PrevPC/Desktop/BMSTU/Diplom/SIMODO/Project/shell$ sudo test/libs
test/mr-test-token
test/mr-test-json-serialization-JsonSerialization
test/mr-test-json-serialization-LexicalParametersLoader
test/mr-test-json-Parser
test/mr-test-json-AnalyzeData
test/mr-test-json5-AnalyzeData
test/mr-test-module
test/mr-test-grammatize
test/mr-test-grammatize2
test/mr-test-parse
test/mr-test-setup-load
test/mr-test-interpret-simodo-script-op
test/mr-test-interpret-simodo-script-check
test/mr-test-interpret-simodo-script-probe
test/mr-test-interpret-simodo-script-debug
test/mr-test-interpret-simodo-script-modules
test/mr-test-interpret-systemverilog
test/mr-test-lsp-client
```

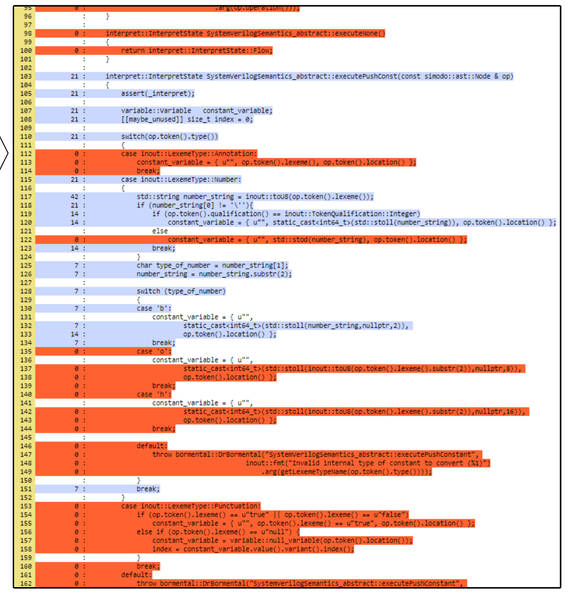
Формирование отчёта о покрытии тестами

LCOV - code coverage report

Current view: top level
Test: sidmodo-shell.info
Date: 2024-05-19 20:26:42

	Hit	Total	Coverage
Lines:	7977	10307	77.4 %
Functions:	998	1182	84.4 %

Directory	Line Coverage		Functions	
include/simodo	100.0 %	6 / 6	100.0 %	4 / 4
include/simodo/ast	100.0 %	21 / 21	100.0 %	12 / 12
include/simodo/ast/generator	100.0 %	3 / 3	75.0 %	3 / 4
include/simodo/bormental	100.0 %	6 / 6	100.0 %	3 / 3
include/simodo/inout/format	100.0 %	12 / 12	90.9 %	10 / 11
include/simodo/inout/log	41.2 %	7 / 17	33.3 %	3 / 9
include/simodo/inout/reporter	86.4 %	19 / 22	77.8 %	7 / 9
include/simodo/inout/token	91.8 %	90 / 98	82.5 %	47 / 57
include/simodo/interpret	88.6 %	39 / 44	71.7 %	33 / 46
include/simodo/interpret/builtins	100.0 %	1 / 1	100.0 %	1 / 1
include/simodo/interpret/builtins/hosts/base	75.0 %	6 / 8	71.4 %	5 / 7
include/simodo/interpret/builtins/hosts/fuze	90.0 %	18 / 20	75.0 %	9 / 12
include/simodo/interpret/builtins/modules	75.0 %	6 / 8	75.0 %	6 / 8
include/simodo/loom	95.7 %	22 / 23	80.0 %	8 / 10
include/simodo/lsp-client	100.0 %	2 / 2	100.0 %	2 / 2
include/simodo/module	87.5 %	14 / 16	66.7 %	10 / 15
include/simodo/parser	100.0 %	28 / 28	92.9 %	13 / 14



Информация о покрытии тестами при запросе на слияние (GitLab MR)

```
117 +               std::string number_string = inout::toU8(op.token().lexeme());
                    if (number_string[0] != '\''){
                        if (op.token().qualification() == inout::TokenQualification::Integer)
Test coverage: 9 hits
120 +                   constant_variable = { u"", static_cast<int64_t>
  (std::stoll(number_string)), op.token().location() };
121 +               else
122 +                   constant_variable = { u"", std::stod(number_string),
  op.token().location() };
123 +               break;
124 +           }
125 +           char type_of_number = number_string[1];
126 +           number_string = number_string.substr(2);
127 +
128 +           switch (type_of_number)
129 +           {
130 +           case 'b':
102 131             constant_variable = { u"",
```

Дипломный проект					*Программный модуль интерпретатора сценариев для системы моделирования*
Изм	*Лист*	*№ документа*	*Подпись*	*Дата*	*Технология тестирования*
Разраб.		*Гриченко К.В.*			*Литер.* *Масса* *Масштаб*
Руковод.		*Фомин М.М.*			*Лист 9* *Листов 9*
Н. контр.		*Ерёмин О.Ю.*			*МГТУ им. Н.Э. Баумана Группа ИУ6-82Б*