

Документация по проекту «Компилятор языка Паскаль» на 1 аттестацию.

**Подготовили Елфимова Екатерина, Мещеряков Илья и Стребкова
Ольга**

Описание проекта.

Компилятор языка Паскаль, написанный на языке python с использованием библиотеки `ruparsing`.

На данный момент в проекте реализовано построение ast-дерева для кода на языке Паскаль.

Структура проекта:

Класс `PascalGrammar` который отвечает за описание грамматики языка Паскаль. В нем присутствует метод `_make_parser` грамматику языка и правила ее разбора. Здесь мы воспользовались модулем `ruparsing` для создания грамматики, используя предоставляемую модулем библиотеку классов для построения грамматики непосредственно в коде Python.

`main.py` используется для вызова функции.

В файле `nodes.py` описаны узлы для построения ast дерева:

`AstNode` - класс предок для всех последующих классов, определяет для узлов такие свойства как получение потомков узла, и рисование узла

`LiteralNode` - узел для описания литерала

`IdentNode` - узел для описания идентификатора

`ArrayIdentNode` - узел для описания элементов массива

`BinOpNode` - узел для описания бинарной операции

`StmtNode` - узел для описания выражения

`IdentListNode` - узел для описания списка идентификаторов

`TypeSpecNode` - узел для описания типа

`VarDeclNode` - узел для описания объявления переменной

`ArrayDeclNode` - узел для описания объявления массива

`VarsDeclNode` - узел для описания объявлений переменных

`CallNode` - узел для описания вызова функции или процедуры

AssignNode - узел для описания присваивания переменной значения

IfNode - узел для описания условного оператора if

WhileNode - узел для описания цикла while

ForNode - узел для описания цикла for

StmtListNode - узел для описания списка выражений

BodyNode - узел для описания тела (внутренности между begin и end)
содержащий список выражений

ParamsNode - узел для описания параметров функции/процедуры

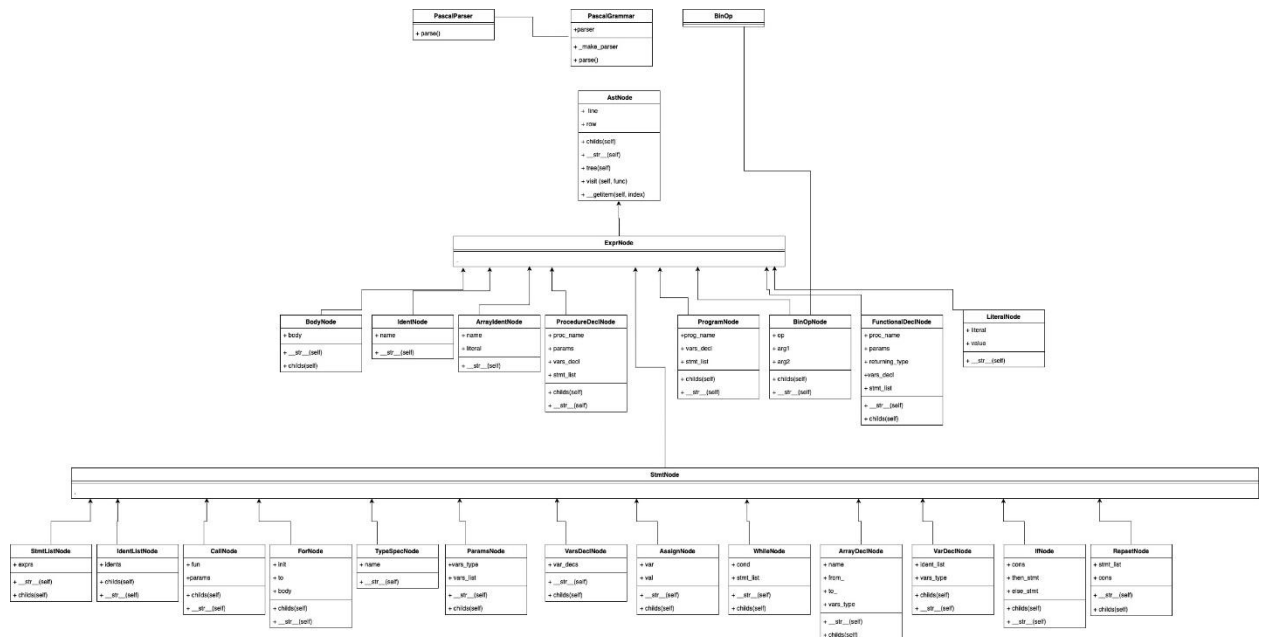
ProgramNode - узел для описания программы

ProcedureDeclNode - узел для описания процедуры

FunctionDeclNode- узел для описания функции

Так же в коде проекта присутствуют комментарии с пояснениями по ходу выполнения программы.

Диаграмма классов, описывающая структуру проекта:



Примеры вход/выход:

вход 1:

Program prog1;

var

```

    j: integer;
BEGIN
    j:=9;
END.

```

Выход 1:

```

Program
└ prog1
└ var
  └ var_dec
    └ idents
      └ j
        └ integer
          └ Body
            └ ...
              └ :=
                └ j
                  └ 9 (int)

```

Вход 2:

```

Program prog1;
var
    k, d: integer;
    j: char;
    g, c: array [1 .. 100] of integer;
BEGIN
    while (i>3) do
        a:=k+2;
    a:=j mod k;
    while (i>3) do
    begin
        a:=k+2;
        a:=j mod k;
    end;
END.

```

Выход 2:

```

Program
└ prog1
└ var
  └ var_dec
    └ idents
      └ k
        └ d
          └ integer

```

```

|  | var_dec
|  | | idents
|  | | | j
|  | | | char
|  | | | arr_dec1
|  | | integer
|  | | idents
|  | | | g
|  | | | | c
|  | | 1 (int)
|  | | | 100 (int)
|  | Body
|  | | ...
|  | | | while
|  | | | | >
|  | | | | | i
|  | | | | | | 3 (int)
|  | | | | | :=
|  | | | | | | a
|  | | | | | | | +
|  | | | | | | | k
|  | | | | | | | 2 (int)
|  | | | | :=
|  | | | | | a
|  | | | | | mod
|  | | | | | | j
|  | | | | | | k
|  | | | while
|  | | | | >
|  | | | | | i
|  | | | | | | 3 (int)
|  | | | | | ...
|  | | | | | | :=
|  | | | | | | | a
|  | | | | | | | | +
|  | | | | | | | | k
|  | | | | | | | | 2 (int)
|  | | | | | | :=
|  | | | | | | | a
|  | | | | | | | | mod
|  | | | | | | | | | j
|  | | | | | | | | | k

```

вход 3:

Program prog1;

var

k, d: integer;

```

function t(j:integer; k: char):integer;
var
    d: integer;
begin
    for (i:=2 to 0 ) do
        g:=0;
        s:=0;
        if (k>2) then
            begin
                f:=9;
                h:=f;
            end;
        else
            v:=3;
        end;
    end;
BEGIN
    t(1,3);
END.

```

выход 3:

Program

```

└ prog1
└ var
  └ var_dec
    └ idents
      └ k
      └ L d
      └ L integer
      └ L function
        └ t
        └ params
          └ integer
            └ k
            └ char
            └ L j
          └ integer
          └ var
            └ L var_dec
              └ idents
                └ L d
                └ L integer
            └ L Body
              └ L ...
                └ for
                  └ :=
                    └ i
                    └ L 2 (int)

```

```

|      | ⊢ 0 (int)
|      |   ⊢ :=
|      |     ⊢ g
|      |       ⊢ 0 (int)
|      ⊢ :=
|      | ⊢ s
|      |   ⊢ 0 (int)
|      ⊢ if
|      |   ⊢ >
|      |     ⊢ k
|      |       ⊢ 2 (int)
|      |     ⊢ ...
|      |       ⊢ :=
|      |         ⊢ ⊢ f
|      |           ⊢ 9 (int)
|      |         ⊢ :=
|      |           ⊢ h
|      |             ⊢ f
|      |         ⊢ :=
|      |           ⊢ v
|      |             ⊢ 3 (int)
|      ⊢ Body
|      ⊢ ...
|      ⊢ call
|      | ⊢ t
|      |   ⊢ 1 (int)
|      |     ⊢ 3 (int)

```

вход 4:

Program prog1;

var

 procedure t(j:integer; k: char);

 var

 d: integer;

 procedure j(d: integer);

 var

 h:integer;

 begin

 d:=h;

 end;

 begin

 a:=78;

 end;

BEGIN

 if (k>2 or l<0) then

```

begin
    f:=9;
    if (k>2 and h>=0) then
        f:=4;
    end;
END.

```

Выход 4:

Program

```

└ prog1
└ var
└ ┌ procedure
└   └ t
└   └ params
└     └ integer
└     └ k
└     └ char
└     ┌ j
└     └ var
└       └ var_dec
└         └ idents
└           ┌ d
└           ┌ integer
└           ┌ procedure
└             └ j
└             └ params
└               └ integer
└               ┌ d
└               └ var
└                 ┌ var_dec
└                   └ idents
└                     ┌ h
└                     ┌ integer
└                     ┌ Body
└                       ┌ ...
└                       ┌ :=
└                         └ d
└                         ┌ h
└                       ┌ Body
└                         ┌ ...
└                         ┌ :=
└                           └ a
└                           ┌ 78 (int)
└ ┌ Body
└   ┌ ...
└     ┌ if
└       └ or

```

```

|  | >
|  | | k
|  | | ^ 2 (int)
|  | ^ <
|  | | 1
|  | ^ 0 (int)
|  ...
|  :=
|  | f
|  | ^ 9 (int)
|  | if
|  | | and
|  | | | >
|  | | | | k
|  | | | | ^ 2 (int)
|  | | | ^ >=
|  | | | | h
|  | | | ^ 0 (int)
|  | :=
|  | | f
|  | | ^ 4 (int)

```

ВХОД 5:

Program t;

var

k, d: integer;

j: char;

g, c: array [1 .. 100] of integer;

function t(j:integer; k: char):integer;

var

d: integer;

begin

a:= 0;

end;

procedure t;

var

d: integer;

begin

a:=78;

end;

g: integer;

BEGIN

g[0]:=10;

g[1]:=c[0];

writeln(a, 3, "df", 7+9);

for (i:=2 to 10) do

begin


```

        k:=2 mod 3;
        l:=j div 4;
        l:=i+8;
        k:=0;
    end;
while (i>3) do
    a:=k+2;
for (i:=2 to 6 ) do
    g:=0;
    s:=0;
    if (k>2 and j>=2) then
        begin
            f:=9;
        end;
    else
        v:=3;
    end;
END.

```

ВЫХОД 5:

Program

```

└ t
└ var
  └ var_dec
    └ idents
      └ k
      └ L d
      └ L integer
    └ var_dec
      └ idents
        └ L j
        └ L char
      └ arr_decl
        └ integer
        └ idents
          └ g
          └ L c
        └ 1 (int)
        └ L 100 (int)
      └ function
        └ t
        └ params
          └ integer
          └ k
          └ char
          └ L j
        └ integer
        └ var

```

```

| | | L var_dec
| | |   | idents
| | |   | L d
| | |   L integer
| | L Body
| |   L ...
| |     L :=
| |       | a
| |       L 0 (int)
| | procedure
| |   | t
| |   | ...
| |   | var
| |   | L var_dec
| |   |   | idents
| |   |   | L d
| |   |   L integer
| |   L Body
| |     L ...
| |     L :=
| |       | a
| |       L 78 (int)
| L var_dec
|   | idents
|   | L g
|   L integer
L Body
  L ...
    | :=
    | | g [0 (int)]
    | L 10 (int)
    | :=
    | | g [1 (int)]
    | L c [0 (int)]
    | call
    | | writeln
    | | a
    | | 3 (int)
    | | "df" (str)
    | L +
    |   | 7 (int)
    |   L 9 (int)
    | for
    | | :=
    | | | i
    | | L 2 (int)
    | | 10 (int)
    | L ...

```

```

|   | :=
|   | | k
|   | | ^ mod
|   | |   | 2 (int)
|   | |   | ^ 3 (int)
|   | :=
|   | | 1
|   | | ^ div
|   | |   | j
|   | |   | ^ 4 (int)
|   | :=
|   | | 1
|   | | ^ +
|   | |   | i
|   | |   | ^ 8 (int)
|   | ^ :=
|   |   | k
|   |   | ^ 0 (int)
| while
| | >
| | | i
| | | ^ 3 (int)
| | ^ :=
| |   | a
| |   | ^ +
| |   |   | k
| |   |   | ^ 2 (int)
| for
| | :=
| | | i
| | | ^ 2 (int)
| | 6 (int)
| | ^ :=
| |   | g
| |   | ^ 0 (int)
| :=
| | s
| | ^ 0 (int)
| ^ if
| | and
| | | >
| | | | k
| | | | ^ 2 (int)
| | | ^ >=
| | |   | j
| | |   | ^ 2 (int)
| | ...
| | ^ :=

```

```
|   ⊢ f
|   ⌞ 9 (int)
⌞ :=
  ⊢ v
  ⌞ 3 (int)
```