Федеральное государственное бюджетное образовательное учреждение
Высшего профессионального образования
Вологодский государственный университет

Институт математики, естественных и компьютерных наук
Кафедра прикладной математики

Отчет по индивидуальному заданию
«Синтаксический анализатор»

Выполнил: Кузнецов Алексей Александрович
Специальность/направление: Прикладная математика и информатика
2 курс, группа ПМ-21

Преподаватель: Свердлов Сергей Залманович

Вологда, 2018

Формулировка задания

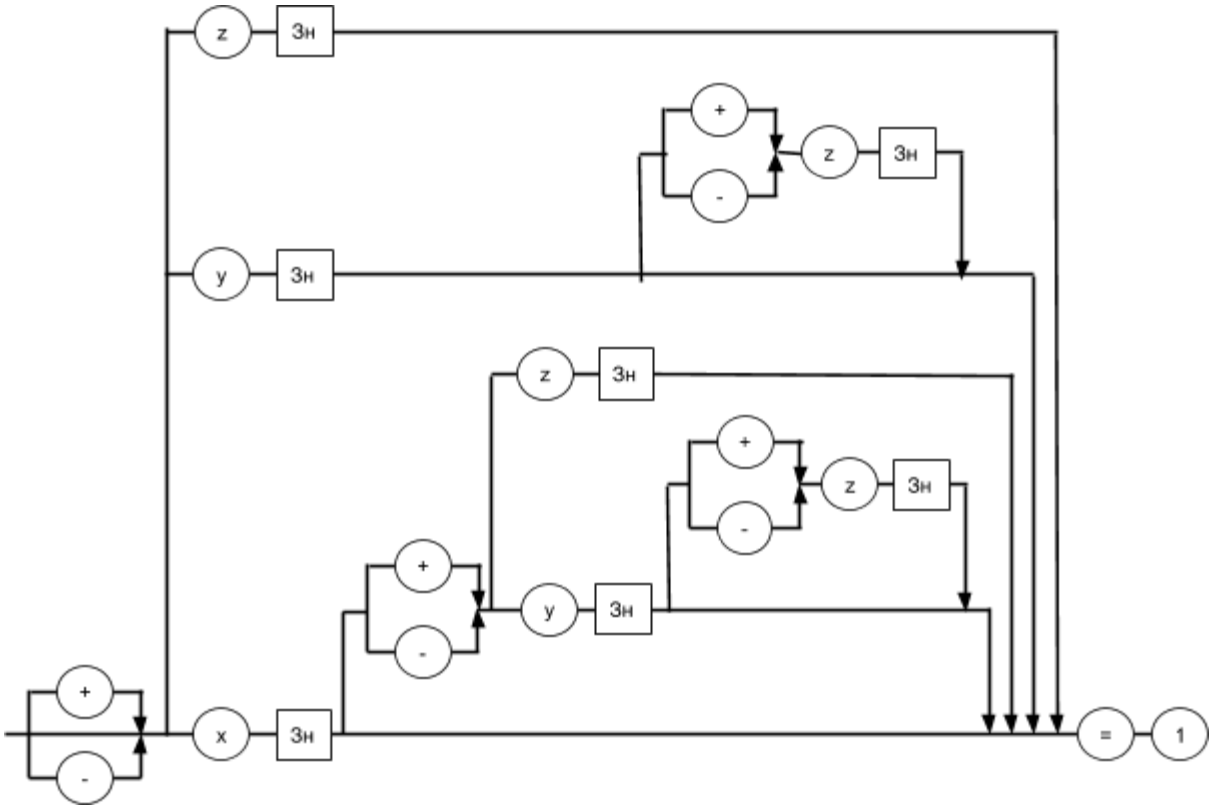
Построить синтаксический анализатор понятия “Уравнение плоскости в отрезках”:

$$x/a + y/b + z/c = 1$$

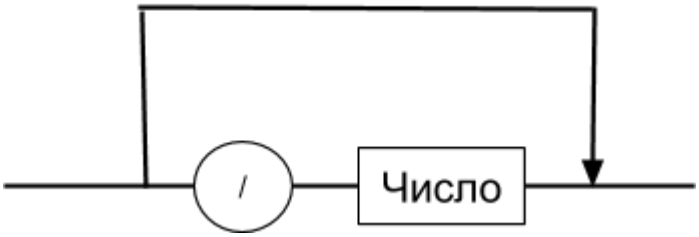
где a, b, c - ненулевые коэффициенты

Синтаксические диаграммы

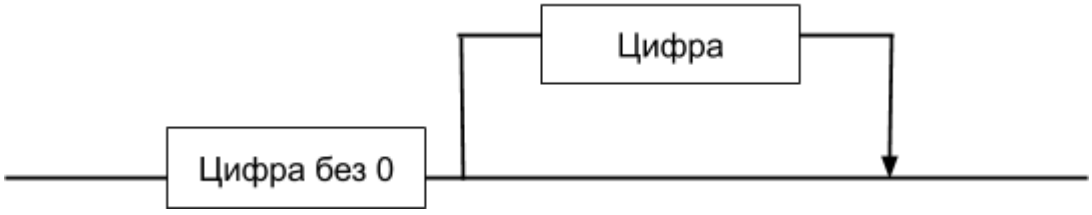
Уравнение плоскости в отрезках:



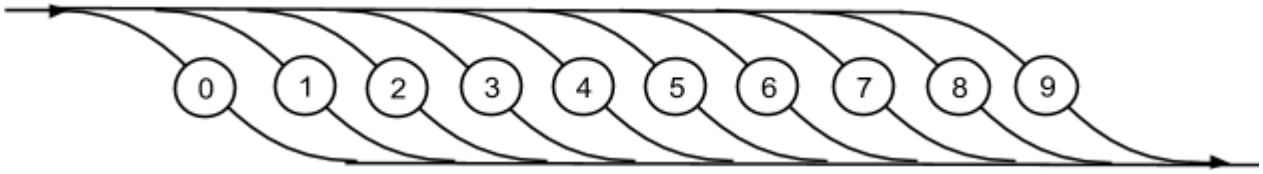
Знаменатель:



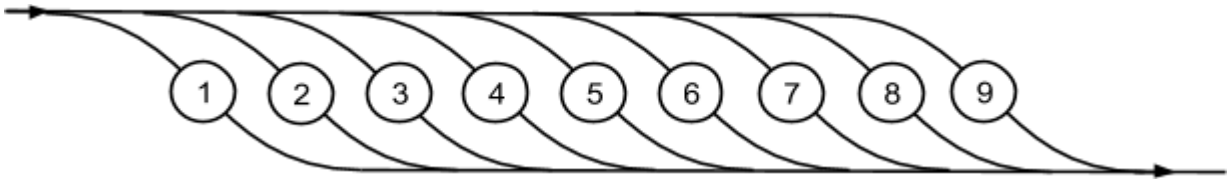
Число:



Цифра:



Цифра без 0:



Текст синтаксического анализатора

на языке Java

```
import java.lang.*;
class PlaneEquation
{
    static final char EOT = (char)10;
    static char Ch;
    static int i, ErrPos;

    public static void main (String[] args)
    {
        System.out.println("Уравнение плоскости в отрезках");
        ResetText();
        ErrPos = 0;
        Equation();
        if (Ch != EOT)
            Error("конец текста");
        else if(ErrPos == 0)
            System.out.println("Правильно");
    }

    static void ResetText()
    {
        i = 0;
        NextCh();
    }

    static void Equation()
    {
        //+ - ...
        if(Ch == '+' || Ch == '-')
            NextCh();
        if(Ch == 'x')
        {
            NextCh();
            Denominator();
            if (Ch == '+' || Ch == '-')
            {
                NextCh();
                if (Ch == 'y')
                {
                    NextCh();
```

```
            Denominator();
            if (Ch == '+' || Ch == '-')
            {
                NextCh();
                if (Ch == 'z')
                {
                    NextCh();
                } else
                    Error("\"z\"");
                Denominator();
            }
        } else if (Ch == 'z')
        {
            NextCh();
            Denominator();
        } else
            Error("\"y\" или \"z\"");
    }
} else if (Ch == 'y')
{
    NextCh();
    Denominator();
    if (Ch == '+' || Ch == '-')
    {
        NextCh();
        if (Ch == 'z')
        {
            NextCh();
        } else
            Error("\"z\"");
        Denominator();
    }
} else if (Ch == 'z')
{
    NextCh();
    Denominator();
} else
    Error("\"x\" или \"y\" или \"z\"");
if (Ch == '=')
{
    NextCh();
} else
    Error("=");
if (Ch == '1')
{
    NextCh();
} else
    Error("1");
```

```
    }

    static void Denominator()
    {
        if (Ch == '/')
        {
            NextCh();
            Number();
        }
    }

    static void Number() {
        if( Ch >= '1' && Ch <= '9' )
            NextCh();
        else
            Error("цифра, отличная от нуля");
        while(Ch >= '0' && Ch <= '9')
            NextCh();
    }

    static void NextCh() {
        do {
            try {
                Ch = (char)System.in.read();
            } catch (Exception e) {
                Error("правильный ввод");
            }
            System.out.print(Ch);
            i++;
        } while( Ch == ' ');
    }

    static void Error(String msg)
    {
        if( ErrPos == 0 )
        {
            ErrPos = i;
            while( Ch != EOT )
                NextCh();
            //System.out.println();
            for(int j = 1; j < ErrPos; j++ )
                System.out.print(" ");
            System.out.println("^\nОжидается " + msg);
        }
    }
}
```

Текст синтаксического анализатора

на языке Pascal

```
program PlaneEquation;
const
   chEOLN = #10;

var
   ErrPos: integer;

   //указатель на символ
   chIt: integer;
   NewEquation: string;
   Ch: char;


procedure NextCh;
begin
   Ch := ' ';
   while Ch = ' ' do begin
      try
         Ch := NewEquation[chIt];
      except
         Writeln('Ожидается правильный ввод');
      end;
      chIt := chIt + 1;
   end;
end;


procedure Error(message: string);
var
   j: integer;
begin
   if ErrPos = 0 then begin
      ErrPos := chIt - 1;
      while Ch <> chEOLN do
         NextCh;
      for j := 1 to ErrPos - 1 do begin
            Write(' ');
      end;
      Writeln('^');
      Writeln('Ожидается ' + message);
```

```
      chIt := chIt + 1;
   end;
end;


procedure ResetText;
begin
   chIt := 1;
   NextCh;
end;


procedure Number;
begin
   if Ch in ['1'..'9'] then
      NextCh
   else
      Error('цифра, отличная от нуля');

   while Ch in ['0'..'9'] do
      NextCh;
end;


procedure Denominator;
begin
   if Ch = '/' then begin
      NextCh;
      Number;
   end;
end;


procedure Equation;
begin
   //+ - ...
   if (Ch = '+') or (Ch = '-') then
      NextCh;

   //AFTER X
   if Ch = 'x' then begin
      NextCh;
      Denominator;

      if (Ch = '+') or (Ch = '-') then begin
         NextCh;

         //AFTER Y
```

```
      if Ch = 'y' then begin
         NextCh;
         Denominator;

         if (Ch = '+') or (Ch = '-') then begin
            NextCh;

            //AFTER Z
            if Ch = 'z' then
               NextCh
            else
               Error('"z"');
            Denominator;
            //END AFTER Z

            end
         end
      //END AFTER Y

      else if Ch = 'z' then begin
      //AFTER Z
         NextCh;
         Denominator;
         end
      else
         Error('"y" или "z"');
      //END AFTER Z

      end
   end
//END AFTER X

//AFTER Y
else if Ch = 'y' then begin
   NextCh;
   Denominator;

   if (Ch = '+') or (Ch = '-') then begin
      NextCh;

      //AFTER Z
      if Ch = 'z' then
         NextCh
      else
         Error('"z"');
      Denominator;
      //END AFTER Z
```

```
        end
      end
   //END AFTER Y

   //AFTER Z
   else if Ch = 'z' then begin
      NextCh;
      Denominator;
      end
   //END AFTER Z

   else
      Error('"x", "y" или "z"');


   if Ch = '=' then
      NextCh
   else
      Error('"="');

   if Ch = '1' then
      NextCh
   else
      Error('1');
end;


begin
   Writeln('Уравнение плоскости в отрезках');
   Readln(NewEquation);
   chIt := 1;
   NewEquation := NewEquation + chEOLN;

   ResetText;

   ErrPos := 0;

   Equation;
   if Ch <> chEOLN then
      Error('конец текста')
   else if ErrPos = 0 then
      Writeln('Правильно');
end.
```