

Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Математическое обеспечение и применение ЭВМ»

ОТЧЕТ
по лабораторной работе №4
по курсу «Теория вычислительных процессов»
на тему «Разработка магазинного автомата»
Вариант 9

Выполнили:

студенты группы 20ВП1

Кубанычбекова А.К.

Макаричева Е.М.

Принял:

к.т.н. Дорофеева О.С.

Пенза 2021

Цель работы:

Освоение методики и получение практических навыков разработки и тестирования магазинного автомата (МА).

Задание:

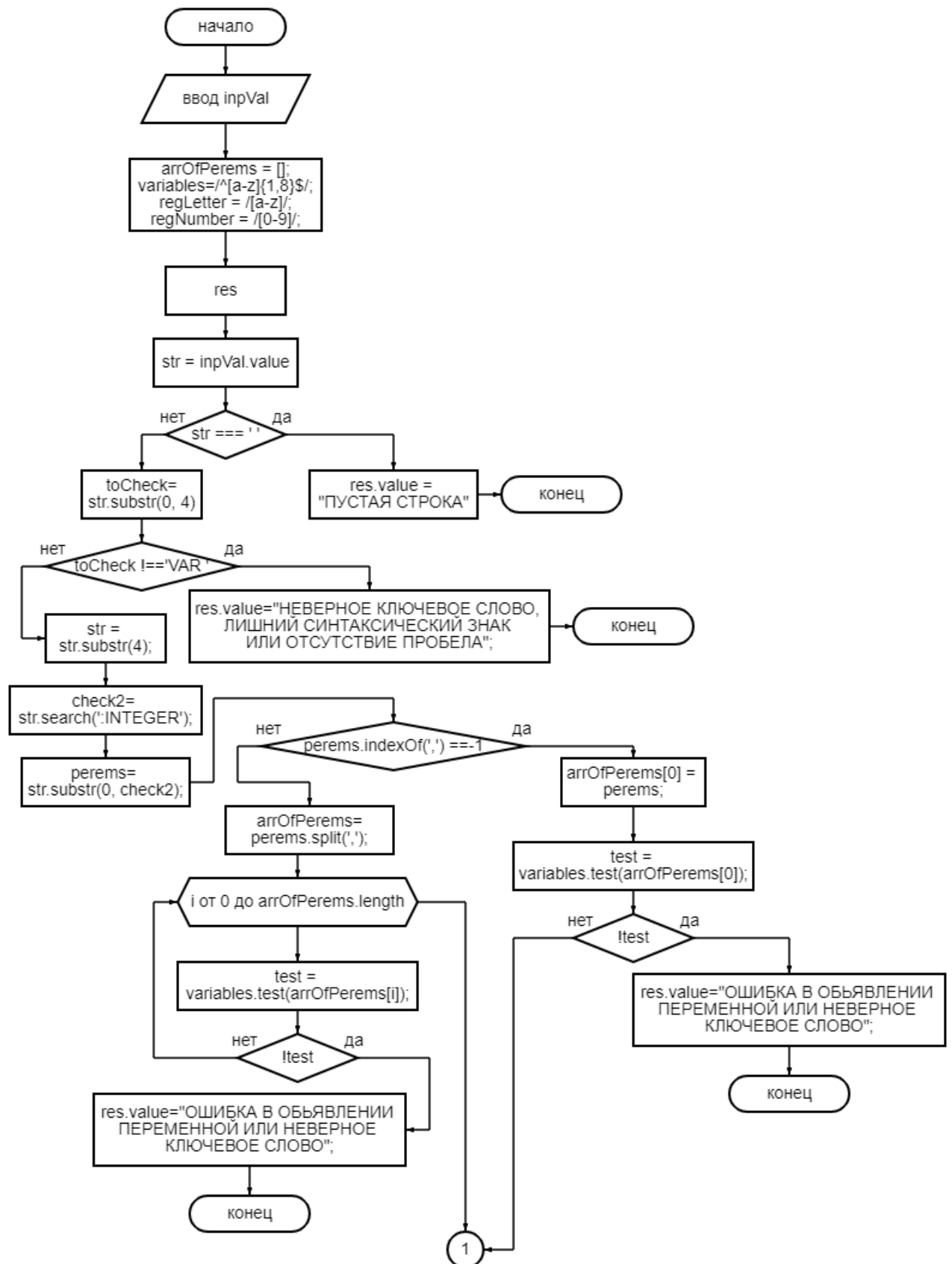
Разработать МА с заданного языка, осуществить программную реализацию и тестирование.

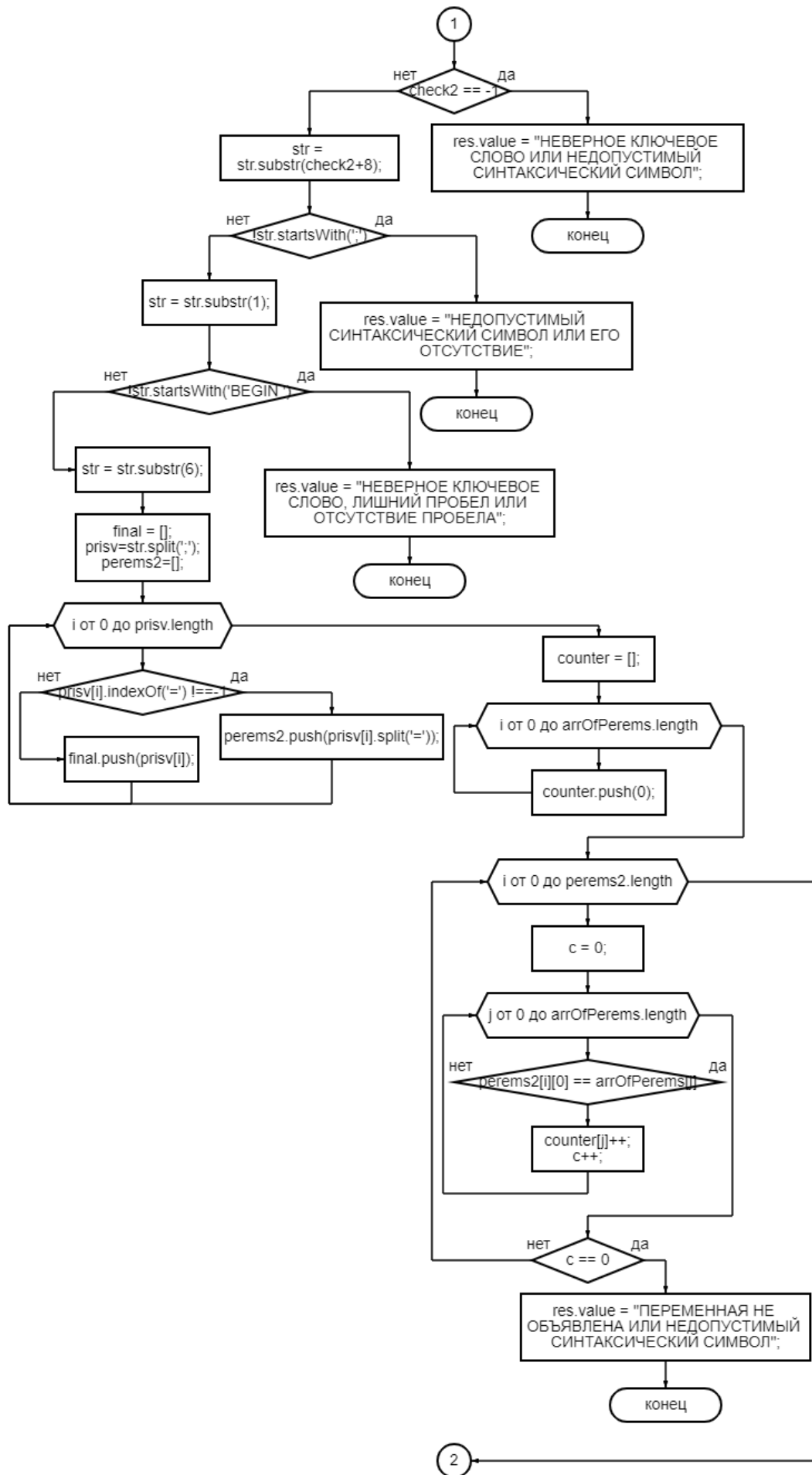
На основании базового описания языка и в соответствии с вариантом задания разработать МА для данного языка. Базовое описание языка имеет следующий вид:

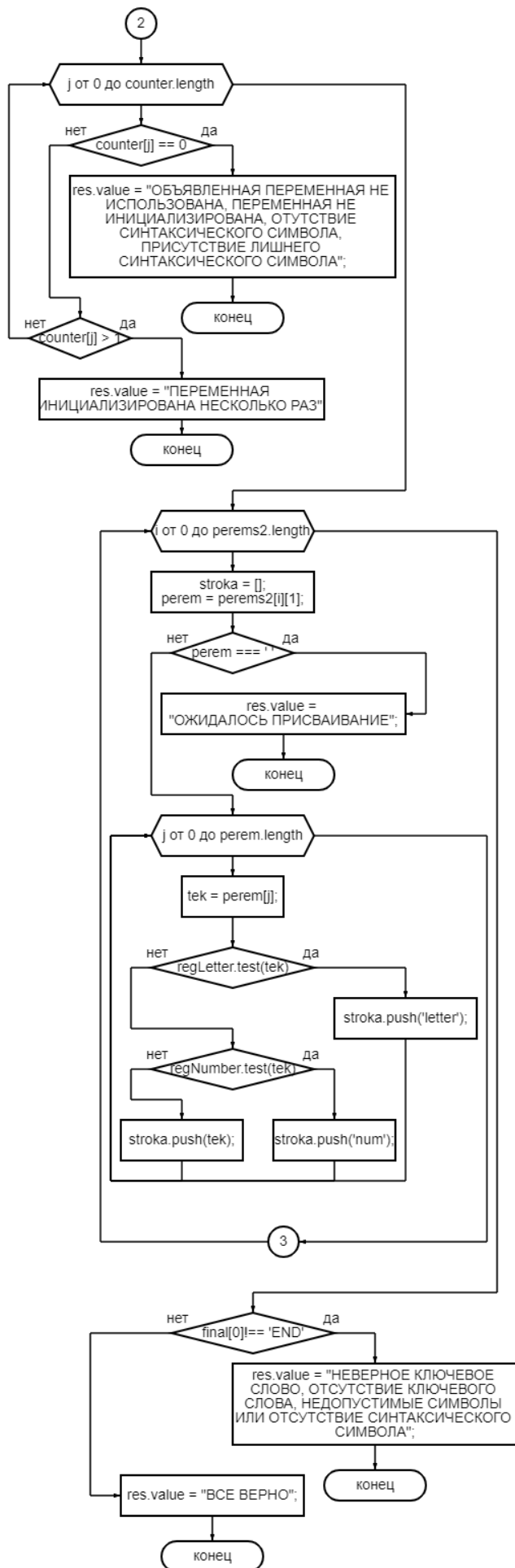
<Программа> ::= <Объявление переменных> <Описание вычислений>
<Описание вычислений> ::= BEGIN <список присваиваний> END
<Объявление переменных> ::= VAR <список переменных> : тип ;
<Список переменных> ::= <Идент>|<Идент>,<Список переменных>
<Список присваиваний> ::= <Присваивание>|<Присваивание> <Список присваиваний>
<Присваивание> ::= <Идент> = <Выражение>;
<Выражение> ::= <Ун.оп.> <Подвыражение> | <Подвыражение>
<Подвыражение> ::= (<Выражение>) | <Операнд> | <Подвыражение>
<Бин.оп.><Подвыражение>
<Ун.оп> ::= вид
<Бин.оп.> ::= вид
<Операнд> ::= <Идент>|<Конст>
<Идент> ::= <Буква><Идент>|<Буква>
<Конст> ::= вид

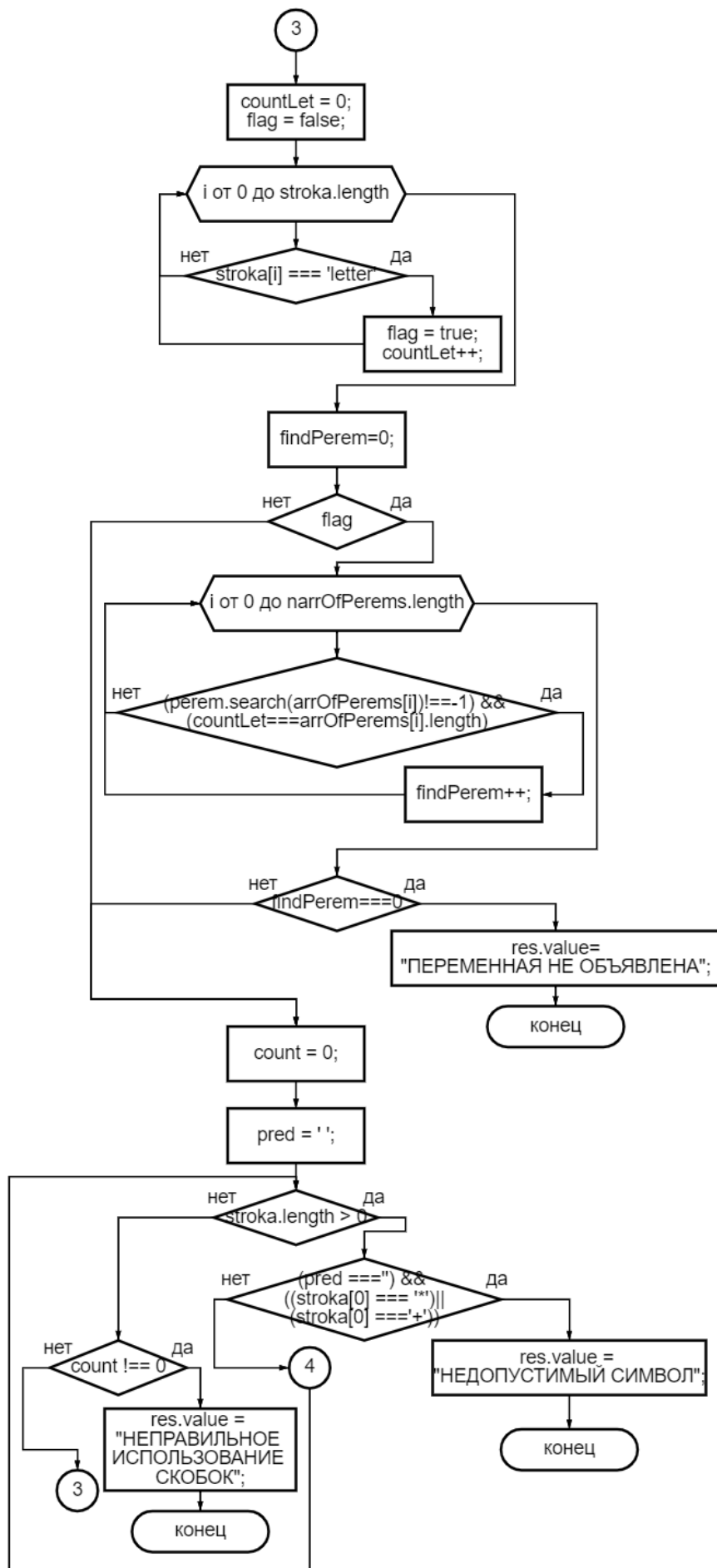
№ варианта	Тип переменных	Вид <Ун. оп.>	Вид <Бин. оп.>	Вид <конст.>	Макс. длина идентификатора
9	INTEGER	-	+ - *	<цифра> <конст> <цифра>	8

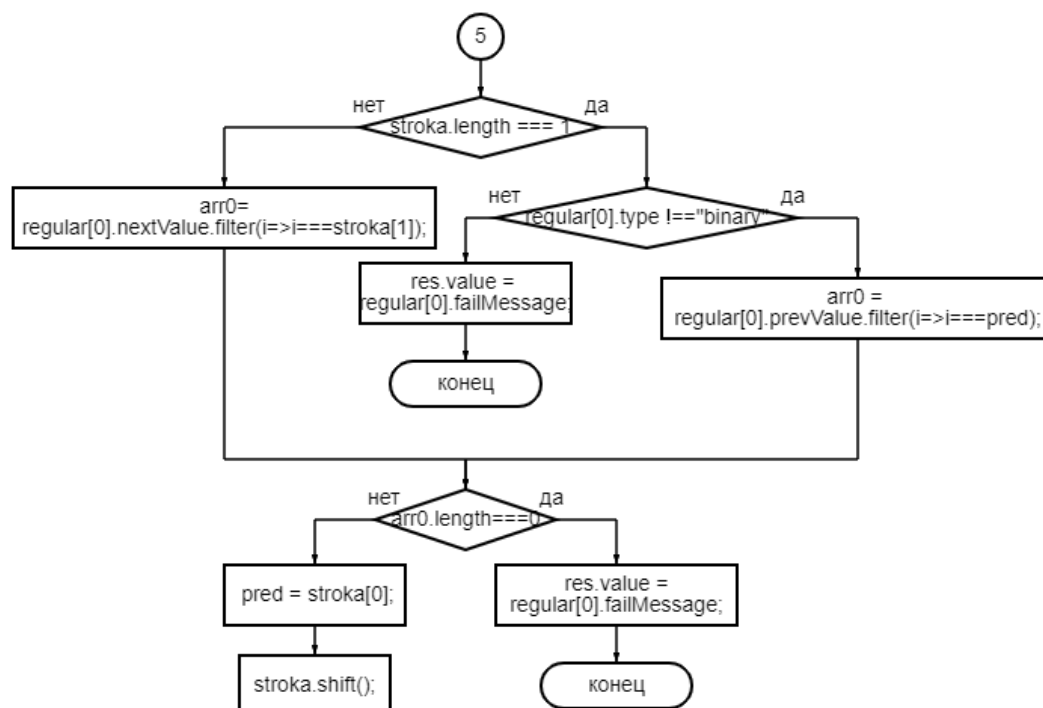
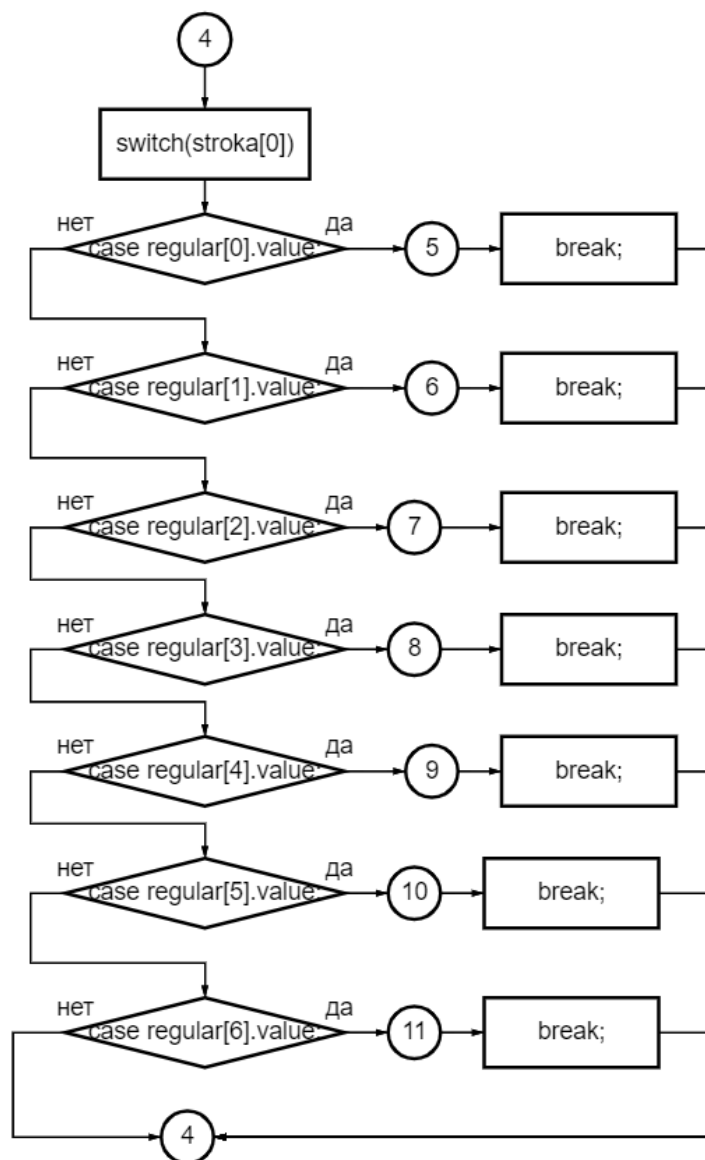
Укрупненная схема алгоритма:

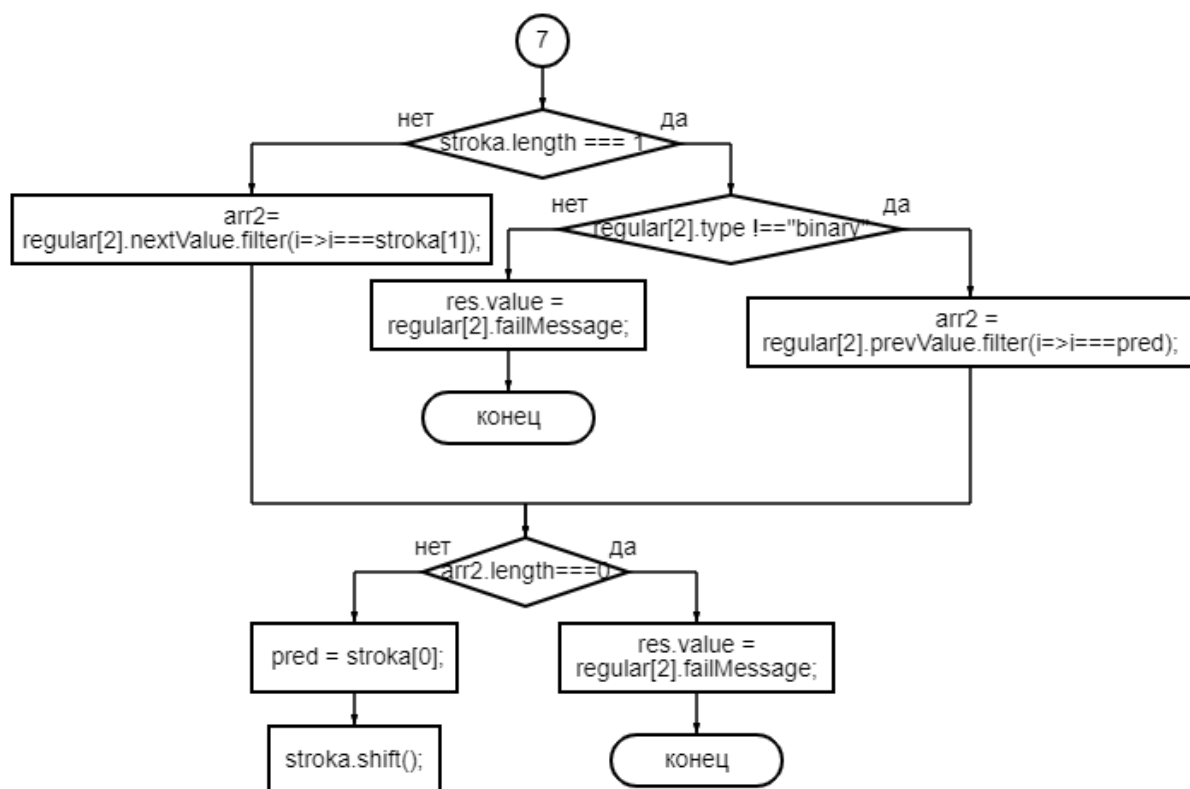
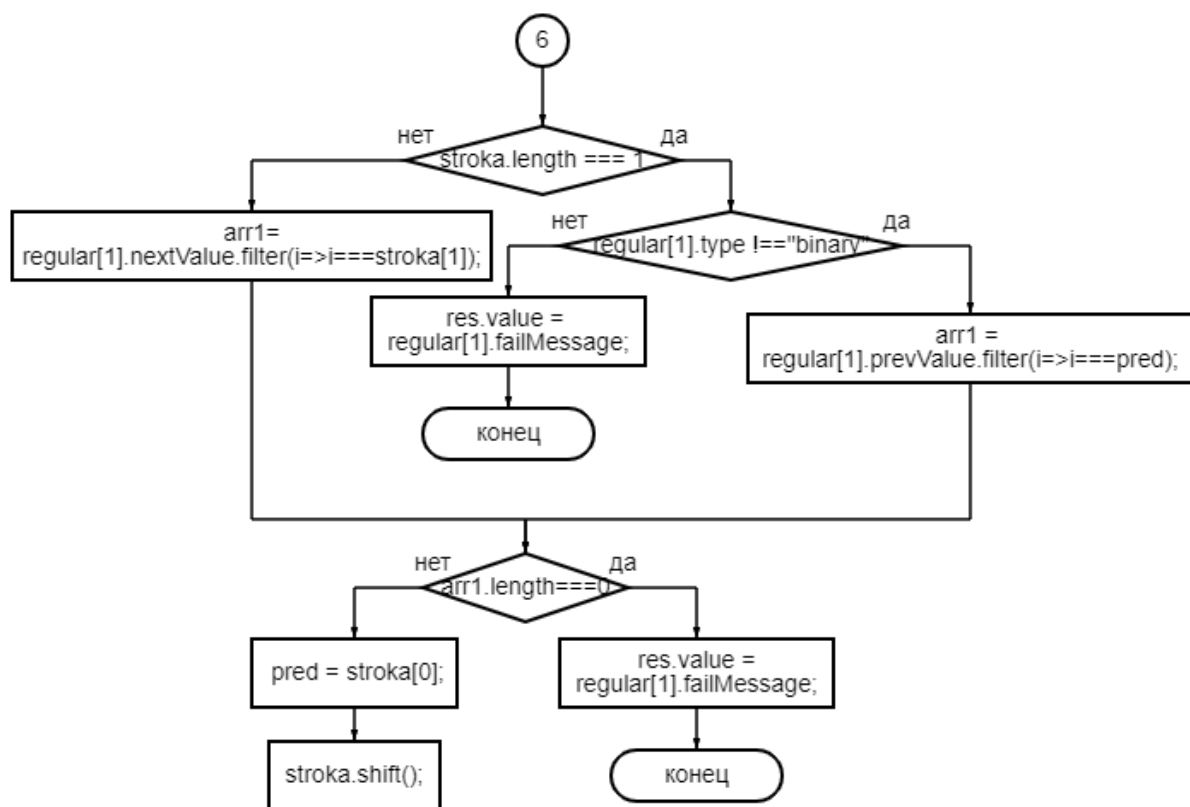


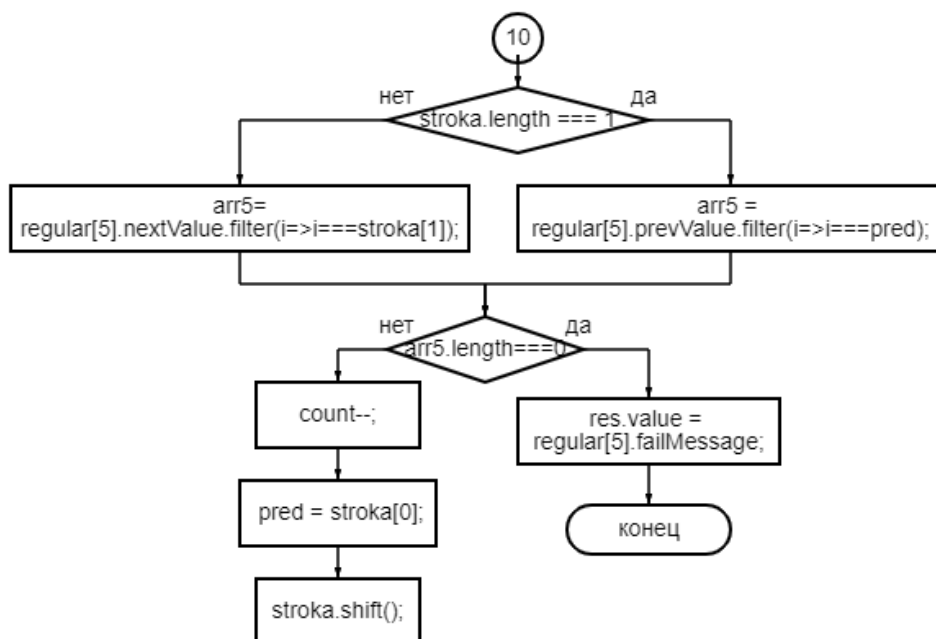
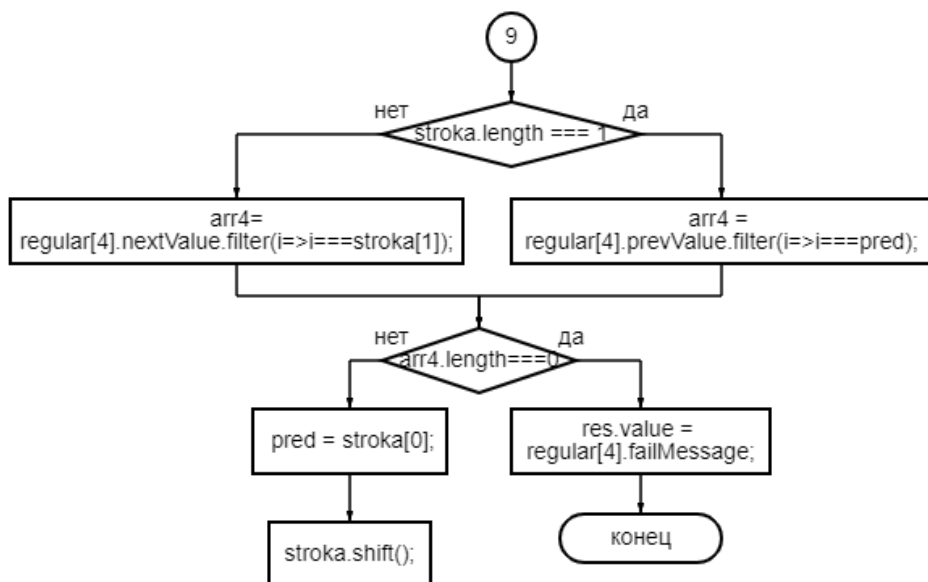
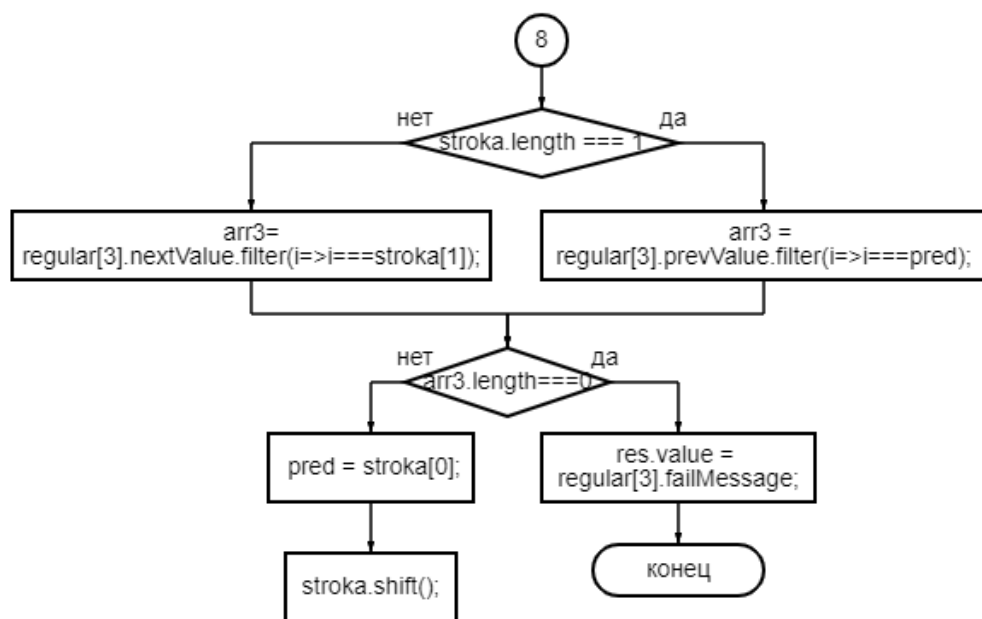


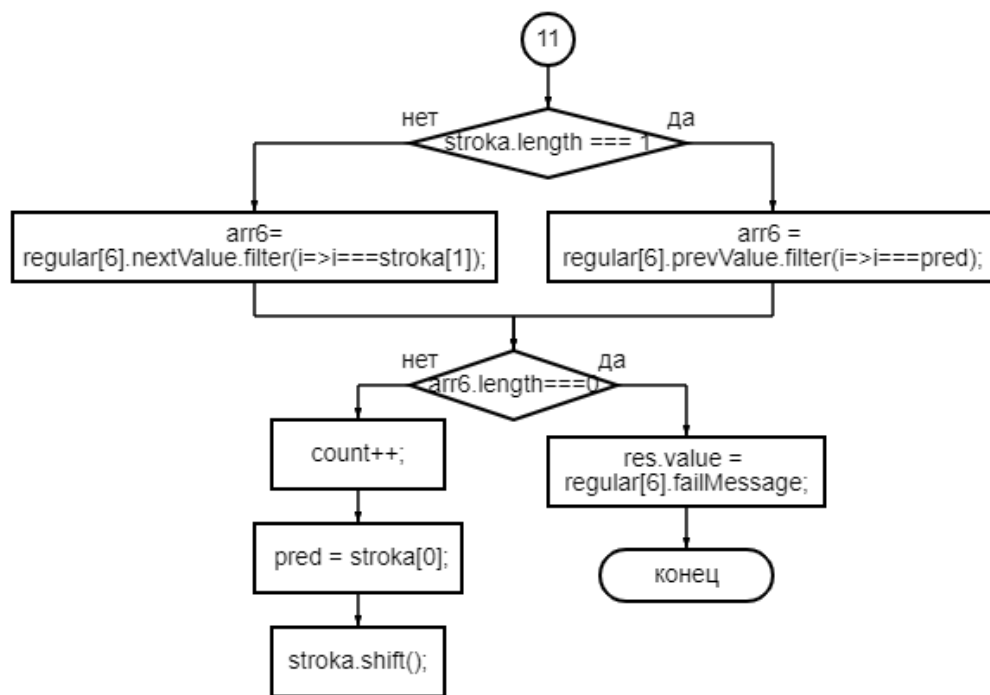












Текст программы:

Файл index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="container">
      <div>
        <input id="inp-val" type="text" />
        <textarea id="res" rows="10" readonly ></textarea>
        <button class="getRes" onclick="startProgram()">Check</button>
      </div>
    </div>
    <script src="main.js"></script>
  </body>
</html>

```

Файл main.js

```

let regular=[
  {
    value:"- ",
    nextValue:['letter','(', 'num'],

```

```

    prevValue:['letter',''),'num','','('],
    failMessage:"НЕ НАЙДЕНА ПЕРЕМЕННАЯ, НЕ НАЙДЕНА СКОБКА, ИЛИ НЕПРАВИЛЬНОЕ
ИСПОЛЬЗОВАНИЕ СИНТАКСИЧЕСКОГО ЗНАКА",
    type:"binary"
  },
  {
    value:"+",
    nextValue:['letter','(','num'],
    prevValue:['letter',''),'num'],
    failMessage:"НЕ НАЙДЕНА ПЕРЕМЕННАЯ, НЕ НАЙДЕНА СКОБКА, ИЛИ НЕПРАВИЛЬНОЕ
ИСПОЛЬЗОВАНИЕ СИНТАКСИЧЕСКОГО ЗНАКА",
    type:"binary"
  },
  {
    value:"*",
    nextValue:['letter','(','num'],
    prevValue:['letter',''),'num'],
    failMessage:"НЕ НАЙДЕНА ПЕРЕМЕННАЯ, НЕ НАЙДЕНА СКОБКА, ИЛИ НЕПРАВИЛЬНОЕ
ИСПОЛЬЗОВАНИЕ СИНТАКСИЧЕСКОГО ЗНАКА",
    type:"binary"
  },
  {
    value:"letter",
    nextValue:['letter','*','-','+','','')'],
    prevValue:['letter','*','-','+','(',''],
    failMessage:"НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ПОСЛЕ ПЕРЕМЕННОЙ",
    type:"none"
  },
  {
    value:"num",
    nextValue:['num','*','-','+','','')'],
    prevValue:['num','(','*','-','+',''],
    failMessage:"НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ПОСЛЕ КОНСТАНТЫ",
    type:"none"
  },
  {
    value:"){",
    nextValue:['*','-','+','','')'],
    prevValue:['num','letter','')'],
    failMessage:"НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СКОБОК, ПРОПУЩЕНА ПЕРЕМЕННАЯ ИЛИ
КОНСТАНТА, НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ",
    type:"none"
  },
  {
    value:"(",
    nextValue:['num','letter','-','('],
    prevValue:['*','-','+','','('],
    failMessage:"НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СКОБОК, ПРОПУЩЕНА ПЕРЕМЕННАЯ ИЛИ
КОНСТАНТА, НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ",
    type:"none"
  }
}

```

```

]
let arrOfPerems = []; //массив объявленных переменных
let inpVal = document.getElementById('inp-val');
let res = document.getElementById('res');
let variables=/^[a-z]{1,8}$/ //регулярное выражение для переменных
let regLetter = /[a-z]/;
let regNumber = /[0-9]/;

function startProgram() {
    let str = inpVal.value;
    if(str === ''){//проверка на пустоту строки
        res.value = "ПУСТАЯ СТРОКА";
        return;
    }
    let toCheck=str.substr(0, 4)//берем из строки первых 4 символов

    if (toCheck !=='VAR '){
        res.value="НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ЛИШНИЙ СИНТАКСИЧЕСКИЙ ЗНАК ИЛИ
ОТСУТСТВИЕ ПРОБЕЛА";
        return;
    }

    str = str.substr(4);
    let check2= str.search(':INTEGER');//-1 берем блок с переменными от VAR до
:INTEGER
    let perems= str.substr(0, check2);

    if(perems.indexOf(',') ==-1){//если в блоке с переменными одна переменная
        arrOfPerems[0] = perems;
        let test = variables.test(arrOfPerems[0]);//проверка на допустимую
переменную
        if(!test) {
            res.value="ОШИБКА В ОБЪЯВЛЕНИИ ПЕРЕМЕННОЙ ИЛИ НЕВЕРНОЕ КЛЮЧЕВОЕ
СЛОВО";
            return;
        }
    }
    else{
        arrOfPerems=perems.split(',');//если переменных несколько, делим блок с
переменными на отдельные переменные
        for(let i=0;i < arrOfPerems.length;i++){
            let test = variables.test(arrOfPerems[i]);
            if(!test) {
                res.value="ОШИБКА В ОБЪЯВЛЕНИИ ПЕРЕМЕННОЙ ИЛИ НЕВЕРНОЕ КЛЮЧЕВОЕ
СЛОВО";
                return;
            }
        }
    }

    if(check2 == -1){//проверка на :INTEGER

```

```

        res.value = "НЕ ВЕРНОЕ КЛЮЧЕВОЕ СЛОВО ИЛИ НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ
СИМВОЛ";
        return;
    }
    str = str.substr(check2+8);
    if(!str.startsWith(';')){//проверка на ; после INTEGER
        res.value = "НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ИЛИ ЕГО ОТСУТСТВИЕ";
        return;
    }
    str = str.substr(1);
    if(!str.startsWith('BEGIN ')){// проверка на BEGIN
        res.value = "НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ЛИШНИЙ ПРОБЕЛ ИЛИ ОТСУТСТВИЕ
ПРОБЕЛА";
        return;
    }
    str = str.substr(6);
    let final = [];//END
    let prsv=str.split(';');//одно из присваиваний
    let perems2=[];//массив пар идентификатор, выражение в присваивании
    for(let i=0;i<prsv.length;i++){
        if(prsv[i].indexOf('=') !==-1){
            perems2.push(prsv[i].split('='));
        }
        else{
            final.push(prsv[i]);
            console.log("final ",final);
        }
    }
    let counter = [];//количество каждого элемента из arrOfPerems перед = в
присваивании
    for(let i = 0; i <arrOfPerems.length;i++){
        counter.push(0);
    }
    for(let i = 0; i <perems2.length;i++){
        let c = 0;//количество инициализированных переменных до = в присваивании

        for(let j = 0; j < arrOfPerems.length; j++){
            if(perems2[i][0] == arrOfPerems[j]){
                counter[j]++;
                c++;
            }
        }
        if(c == 0){
            res.value = "ПЕРЕМЕННАЯ НЕ ОБЪЯВЛЕНА ИЛИ НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ
СИМВОЛ";
            return;
        }
    }
    for(let j = 0; j < counter.length; j++){
        if(counter[j] == 0){

```

```

        res.value = "ОБЪЯВЛЕННАЯ ПЕРЕМЕННАЯ НЕ ИСПОЛЬЗОВАНА, ПЕРЕМЕННАЯ НЕ
ИНИЦИАЛИЗИРОВАНА, ОТСУТСТВИЕ СИНТАКСИЧЕСКОГО СИМВОЛА, ПРИСУТСТВИЕ ЛИШНЕГО
СИНТАКСИЧЕСКОГО СИМВОЛА";
        return;
    }
    if(counter[j] > 1){
        res.value = "ПЕРЕМЕННАЯ ИНИЦИАЛИЗИРОВАНА НЕСКОЛЬКО РАЗ";
        return;
    }
}
for(let i = 0; i < perems2.length; i++){
    let stroka = [];
    let perem = perems2[i][1];
    if(perem === ''){
        res.value = "ОЖИДАЛОСЬ ПРИСВАИВАНИЕ";
        return;
    }
    for(let j = 0; j < perem.length; j++){
        let tek = perem[j];
        if(regLetter.test(tek)){
            stroka.push('letter');
        }
        else if(regNumber.test(tek)){
            stroka.push('num');
        }
        else{
            stroka.push(tek);
        }
    }
}
let flag = false;
for(let i = 0; i < stroka.length; i++){
    if(stroka[i] === 'letter'){
        flag = true;
    }
}
let findPerem=0;
if(flag){
    for(let i = 0; i < arrOfPerems.length; i++){
        if(perem.search(arrOfPerems[i])!==-1){
            findPerem++;
        }
    }
    if(findPerem===0){
        res.value="ПЕРЕМЕННАЯ НЕ ОБЪЯВЛЕНА";
        return;
    }
}
let count = 0;
let pred = '';
while(stroka.length > 0){
    if((pred === '') && ((stroka[0] === '*')||(stroka[0] ==='+'))){

```

```

res.value = "НЕДОПУСТИМЫЙ СИМВОЛ";
return;
}
switch(stroka[0]){
case regular[0].value:
    if(stroka.length === 1){
        if(regular[0].type !== "binary"){
            arr0 = regular[0].prevValue.filter(i=>i===pred);
        }
        else{
            res.value = regular[0].failMessage;
            return;
        }
    }
    else{
        arr0=regular[0].nextValue.filter(i=>i===stroka[1]);
    }
    if(arr0.length===0){
        res.value = regular[0].failMessage;
        return;
    }
    pred = stroka[0];
    stroka.shift();
    break;
case regular[1].value:
    if(stroka.length === 1){
        if(regular[1].type !== "binary"){
            arr1 = regular[1].prevValue.filter(i=>i===pred);
        }
        else{
            res.value = regular[1].failMessage;
            return;
        }
    }
    else{
        arr1=regular[1].nextValue.filter(i=>i===stroka[1]);
    }
    if(arr1.length===0){
        res.value = regular[1].failMessage;
        return;
    }
    pred = stroka[0];
    stroka.shift();
    break;
case regular[2].value:
    if(stroka.length === 1){
        if(regular[2].type !== "binary"){
            arr2 = regular[2].prevValue.filter(i=>i===pred);
        }
        else{
            res.value = regular[2].failMessage;

```

```

        return;
    }
}
else{
    arr2=regular[2].nextValue.filter(i=>i===stroka[1]);
}
if(arr2.length===0){
    res.value = regular[2].failMessage;
    return;
}
pred = stroka[0];
stroka.shift();
break;
case regular[3].value:
    if(stroka.length === 1){
        arr3 = regular[3].prevValue.filter(i=>i===pred);
    }
    else{
        arr3=regular[3].nextValue.filter(i=>i===stroka[1]);
    }
    if(arr3.length===0){
        res.value = regular[3].failMessage;
        return;
    }
    pred = stroka[0];
    stroka.shift();
    break;
case regular[4].value:
    if(stroka.length === 1){
        arr4= regular[4].prevValue.filter(i=>i===pred);
    }
    else{
        arr4=regular[4].nextValue.filter(i=>i===stroka[1]);
    }
    if(arr4.length===0){
        res.value = regular[4].failMessage;
        return;
    }
    pred = stroka[0];
    stroka.shift();
    break;
case regular[5].value:
    if(stroka.length === 1){
        arr5= regular[5].prevValue.filter(i=>i===pred);
    }
    else{
        arr5=regular[5].nextValue.filter(i=>i===stroka[1]);
    }
    if(arr5.length===0){
        res.value = regular[5].failMessage;
        return;
    }

```



```

        }
        count--; //подсчет скобок
        pred = stroka[0];
        stroka.shift();
        break;
    case regular[6].value:
        if(stroka.length === 1){
            arr6= regular[6].prevValue.filter(i=>i===pred);
        }
        else{
            arr6=regular[6].nextValue.filter(i=>i===stroka[1]);
        }
        if(arr6.length===0){
            res.value = regular[6].failMessage;
            return;
        }
        count++; //подсчет скобок
        pred = stroka[0];
        stroka.shift();
        break;
    }
}

if(count !== 0){
    res.value = "НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СКОБОК";
    return;
}

}

if(final[0]!== 'END'){
    res.value = "НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ОТСУТСТВИЕ КЛЮЧЕВОГО СЛОВА,
НЕДОПУСТИМЫЕ СИМВОЛЫ ИЛИ ОТСУТСТВИЕ СИНТАКСИЧЕСКОГО СИМВОЛА";
    return;
}

res.value = "ВСЕ ВЕРНО";
return;
}

```

Файл style.css

```

.container div {
    width: 1000px;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}
.container {
    display: flex;
    align-items: center;
    justify-content: center;
    width: 100vw;
    height: 100vh;
}

```

```

}
#inp-val,
#res {
width: 970px;
height: 150px;
font-size: 20px;
margin-bottom: 25px;
}
.getRes {
width: 150px;
height: 50px;
}

```

Результаты тестирования:

Результаты тестирования представлены на рисунках 1-18.

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3;EN
```

НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ОТСУТСТВИЕ КЛЮЧЕВОГО СЛОВА, НЕДОПУСТИМЫЕ СИМВОЛЫ ИЛИ ОТСУТСТВИЕ СИНТАКСИЧЕСКОГО СИМВОЛА

Рисунок 1 – Неверное ключевое слово END

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

ВСЕ ВЕРНО

Рисунок 2 – Верная последовательность

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-((-7*(6+apple));red=apple;sun=3;END
```

НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СКОБОК

Рисунок 3 – Лишняя скобка в $-((-7*(6+apple))$

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=+;color=-((-7*(6+apple)));red=apple;sun=3;END
```

НЕДОПУСТИМЫЙ СИМВОЛ

Рисунок 4 – Ошибка в apple=+

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=+;color=-(-7*(6+apple));red=apple;sun=3;END
```

ОЖИДАЛОСЬ ПРИСВАИВАНИЕ

Рисунок 5 – Ошибка в apple=

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sunny;color=-(-7*(6+apple));red=apple;sun=3;END
```

ПЕРЕМЕННАЯ НЕ ОБЪЯВЛЕНА

Рисунок 6 – Не объявлена переменная sunny

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sunny;red=6;color=-(-7*(6+apple));red=apple;sun=3;END
```

ПЕРЕМЕННАЯ ИНИЦИАЛИЗИРОВАНА НЕСКОЛЬКО РАЗ

Рисунок 7 – Переменная red проинициализирована несколько раз

```
VAR apple,color,red,sun:INTEGER;BEGIN color=-(-7*(6+apple));red=color;sun=3;END
```

ОБЪЯВЛЕННАЯ ПЕРЕМЕННАЯ НЕ ИСПОЛЬЗОВАНА, ПЕРЕМЕННАЯ НЕ ИНИЦИАЛИЗИРОВАНА, ОТСУТСТВИЕ СИНТАКСИЧЕСКОГО СИМВОЛА, ПРИСУТСТВИЕ ЛИШНЕГО СИНТАКСИЧЕСКОГО СИМВОЛА

Рисунок 8 – Переменная apple не инициализирована

```
VAR apple,color,red,sun:INTEGER;BEGIN apgfple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

ПЕРЕМЕННАЯ НЕ ОБЪЯВЛЕНА ИЛИ НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ

Рисунок 9 – Не объявлена переменная apgfple

```
VAR apple,color,red,sun:INTEGER;BEGINapple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ЛИШНИЙ ПРОБЕЛ ИЛИ ОТСУТСТВИЕ ПРОБЕЛА

Рисунок 10 – Отсутствие пробела после BEGIN

```
VAR apple,color,red,sun:INTEGERBEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ИЛИ ЕГО ОТСУТСТВИЕ

Рисунок 11 – Ошибка в INTEGERBEGIN

```
VAR apple,color,red,sunNTEGER;BEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

ОШИБКА В ОБЪЯВЛЕНИИ ПЕРЕМЕННОЙ ИЛИ НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО

Рисунок 12 – Ошибка в sunNTEGER

```
VR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3;END
```

НЕВЕРНОЕ КЛЮЧЕВОЕ СЛОВО, ЛИШНИЙ СИНТАКСИЧЕСКИЙ ЗНАК ИЛИ ОТСУТСТВИЕ ПРОБЕЛА

Рисунок 13 – Ошибка в VR

ПУСТАЯ СТРОКА

Рисунок 14 – Пустая последовательность

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple+()));red=apple;sun=3;END
```

НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СКОБОК, ПРОПУЩЕНА ПЕРЕМЕННАЯ ИЛИ КОНСТАНТА, НЕДОПУСТИМЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ

Рисунок 15 – Пустые скобки ()

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple*));red=apple;sun=3;END
```

НЕ НАЙДЕНА ПЕРЕМЕННАЯ, НЕ НАЙДЕНА СКОБКА, ИЛИ НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ СИНТАКСИЧЕСКОГО ЗНАКА

Рисунок 16 – Ошибка в apple*

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun(;color=-(-7*(6+apple));red=apple;sun=3;END
```

НЕДОПУСТИМЫЙ ИЛИ ПРОПУЩЕННЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ПОСЛЕ ПЕРЕМЕННОЙ

Рисунок 17 – Ошибка в apple=sun(

```
VAR apple,color,red,sun:INTEGER;BEGIN apple=sun;color=-(-7*(6+apple));red=apple;sun=3(;END
```

НЕДОПУСТИМЫЙ ИЛИ ПРОПУЩЕННЫЙ СИНТАКСИЧЕСКИЙ СИМВОЛ ПОСЛЕ КОНСТАНТЫ

Рисунок 18 – Ошибка в sun=3(

Выводы:

В результате выполнения лабораторной работы были получены навыки разработки и тестирования магазинного автомата. В соответствии с заданием и описанием базового языка, была составлена укрупненная схема алгоритма, написана программа реализации магазинного автомата и проведено ее тестирование.