**МИНИСТЕРСТВО ОБРАЗВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Частное учреждение образования**

**«Гродненский колледж бизнеса и права»**

**Лабораторная работа № 3**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

**Тема:** Использование структуры стек для преобразования выражений в обратную польскую связь и вычисление их значений

для учащихся 2 курса специальности

2-40 01 01 «Программное обеспечение информационных технологий»

**ЛАБОРАТОРНАЯ РАБОТА № 3**

Тема: Использование структуры стек для преобразования выражений в обратную польскую связь и вычисление их значений.

Цель: Изучить правила формирования постфиксной записи арифметических выражений с использованием стека.

Задачи: Ознакомление с примерами, выполнение индивидуальной работы.

**ЗАДАЧИ**

Условие: Написать программу формирования ОПЗ и расчета полученного выражения. Разработать удобный интерфейс ввода исходных данных и вывода результатов. Работу программы проверить на конкретном примере:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Выражение | *a* | *b* | *c* | *d* | *e* | Результат |
| 1 | a\*(b+c)/(d– e) | *0.4* | *2.3* | *6.7* | *5.8* | *9.1* | – 1.091 |

Алгоритм: Предоставлен в письменном виде преподавателю.

Решение:

**const**

op1=['+','-'];

op2=['\*','/'];

ops= ['+', '-', '\*', '/'];

symb=['a'..'z','A'..'Z','0'..'9','.'];

numb=['0'..'9'];

**var**

sym :char;

so,si,so1,q:string;

p :word;

i,k,error:integer;

d,a,b,c:real;

stack:**array** [1..100] **of** string;

**procedure** getsym;

**begin**

**while** (p<=length(si))**and**(si[p] **in** [#32, #9]) **do**

inc(p);

**if** p>length(si) **then**

sym:=#0

**else**

**begin**

sym:=si[p];

inc(p)

**end**;

**end**;

**procedure** expression;

**forward**;

**procedure** get\_var;

**begin**

so:=so+#32;

**while** sym **in** symb **do**

**begin**

so:=so+sym;

getsym

**end**

**end**;

**procedure** term;

**var** a1:char;

**begin**

**if** sym='(' **then**

**begin**

getsym;

expression;

getsym

**end**

**else**

get\_var;

**while** sym **in** op2 **do**

**begin**

a1:=sym;

getsym;

term;

so:=so+#32+a1

**end**

**end**;

**procedure** expression;

**var** a1:char;

**begin**

**if** sym='+' **then**

getsym

**else if** sym='-' **then**

**begin**

a1:='-';getsym

**end**

**else**

a1:=#32;

term;

**if** a1='-' **then** so:=so+' (-)';

**while** sym **in** op1 **do**

**begin**

a1:=sym;

getsym;

term;

so:=so+#32+a1

**end**

**end**;

**begin**

si:='0.4\*(2.3+6.7)/(5.8-9.1)';

so:=#32;

p:=1;

getsym;

expression;

writeln('Дано: ',si);

writeln('ОПЗ: ',so);

**for** i:=1 **to** length(so) **do**

**if** so[i] **in** numb **then**

**begin**

so1:=so1+so[i];

so[i]:=' ';

**end**

**else**

**if**(so[i]=' ')**and**(length(so1)>0) **then**

**begin**

inc(k);

stack[k]:=so1;

so1:='';

**end**

**else**

**if** (so[i]='(') **and**(so[i+1]='-') **then**

**begin**

val(stack[k],c,error);

c:=0-c;

str(c,q);

stack[k]:=q;

**end**

**else**

**if** so[i]**in** ops **then**

**begin**

val(stack[k-1],a,error);

val(stack[k],b,error);

**case** so[i] **of**

'+': d:=a+b;

'-': d:=a-b;

'\*': d:=a\*b;

'/': d:=a/b;

**end**;

stack[k]:='';

dec(k);

**end**;

val(stack[k],d,error);

Writeln('Результат: ',d);

**end**.

