**Practical: 2**

**Aim: Write the program to Evaluate Expression with user defined symbol table structure**

**(A). Validate the string as valid variable name.**

**Answer:**

package CD;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Pr1

{

public static String str() throws IOException

{

return new BufferedReader(new InputStreamReader(System.in)).readLine();

}

public static void check(String a)

{

char []b=a.toCharArray();

int i;

if(a.length()>8)

{

System.out.println("Length must be less than 9");

System.exit(0);

}

for(i=0;i<b.length;i++)

{

if(!((b[i]>='a'&& b[i]<='z')||(b[i]>='A'&&b[i]<='Z')||(b[i]=='\_')||(b[i]>='0'&&b[i]<='9')))

{

System.out.println("Only Alphanumeric values and \_ are allowed");

System.exit(0);

}

}

if(b[0]>='0'&&b[0]<='9')

{

System.out.println("Variables must start with Alphabets or \_");

System.exit(0);

}

if(b[i-1]=='\_')

{

System.out.println("Variables must end with Alphanumeric values");

System.exit(0);

}

System.out.println("Given string is variable");

}

public static void main(String[] args) throws IOException

{

System.out.print("Enter String:");

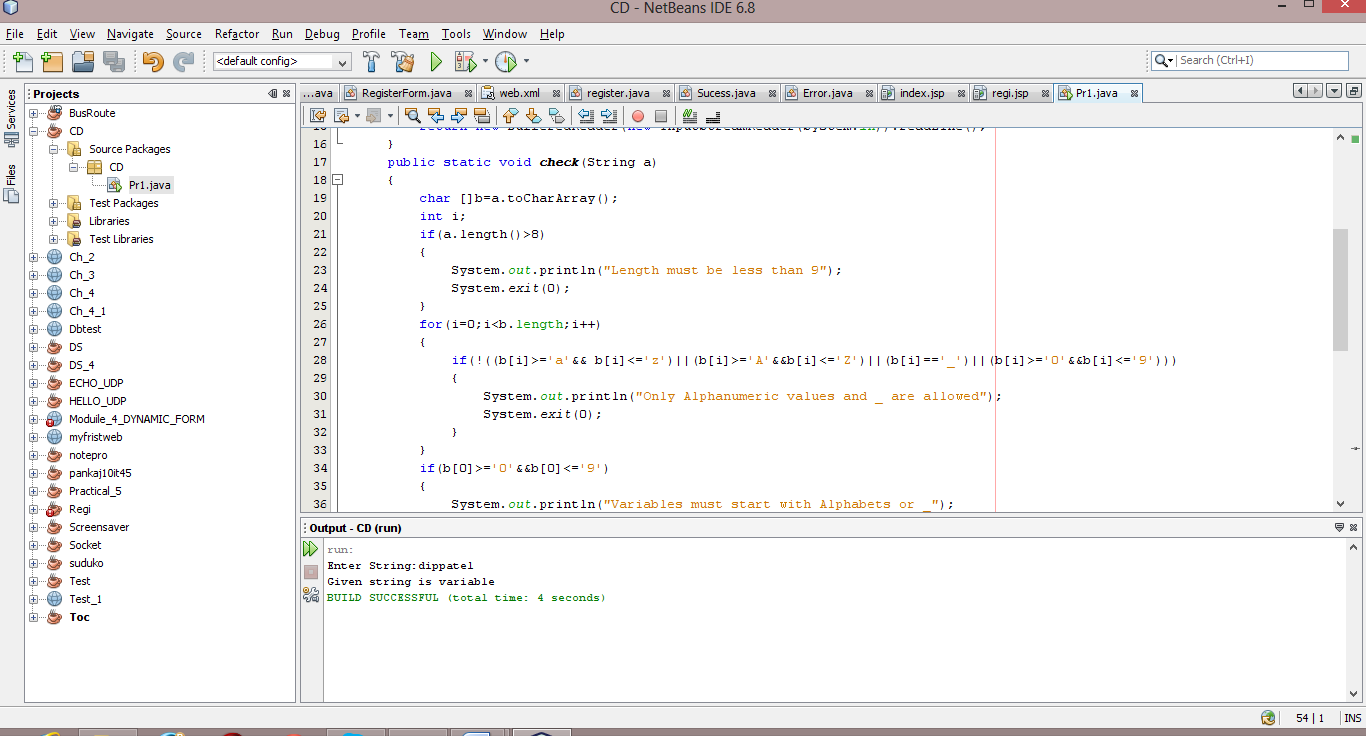
String s=str();

check(s);

}

}

**OUTPUT:**



**(B). Validate the expression with binary operator and symbol defined in phase A.**

**Answer:**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package CD;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Pr2

{

public static void check1(String a,ArrayList<String> b,int r)

{

char c=a.charAt(0);

if(c=='+'||c=='-'||c=='/'||c=='\*'||c=='=')

{

System.out.println("Error at row "+r+" : "+a);

System.out.print("Invalid Expression");

System.exit(0);

}

if(b.contains(a.trim())==false)

{

System.out.println("Error at row "+r+" : "+a);

System.out.print(" not found in symbol table");

System.exit(0);

}

}

public static void check(String a)

{

char []b=a.toCharArray();

int i;

if(a.length()>8)

{

System.out.print("Length must be less than 9");

System.exit(0);

}

for(i=0;i<b.length;i++)

{

if(!((b[i]>='a'&& b[i]<='z')||(b[i]>='A'&&b[i]<='Z')||(b[i]=='\_')||(b[i]>='0'&&b[i]<='9')))

{

System.out.print("Only Alphanumeric values and \_ are allowed");

System.exit(0);

}

}

if(b[0]>='0'&&b[0]<='9')

{

System.out.print("Variables must start with Alphabets or \_");

System.exit(0);

}

if(b[i-1]=='\_')

{

System.out.print("Variables must end with Alphanumeric values");

System.exit(0);

}

System.out.println("Given string is variable");

}

public static void main(String[] args) throws IOException

{

FileReader fr=new FileReader("E:/Sem 7/CD/var.txt");

BufferedReader br=new BufferedReader(fr);

int i,cnt=0,row=0;

String s = "";

ArrayList<String> list=new ArrayList<String>();

ArrayList<String> e=new ArrayList<String>();

while((i=br.read())!=-1)

{

if((char)i==';')

row++;

if((char)i==',')

{

System.out.println(s);

if(list.contains(s)==true)

{

System.out.print("ERROR::Duplicate generation");

System.exit(0);

}

check(s);

list.add(s);

i=br.read();

s="";

}

else if((char)i==';'&& cnt==0)

{

cnt++;

System.out.println(s);

if(list.contains(s)==true)

{

System.out.print("Duplicate generation");

System.exit(0);

}

check(s);

list.add(s);

i=br.read();

s="";

}

else if((char)i=='='||(char)i=='+'||(char)i=='\*'||(char)i=='/'||(char)i=='-'||(char)i==';')

{

e.add(s.trim());

if((char)i!=';')

e.add((char)i+"".trim());

check1(s,list,row);

i=br.read();

s="";

}

s+=(char)i;

}

br.close();

int j=0,ans=0,k=0,x=e.indexOf("=")+1;

List<Integer> no=new ArrayList<Integer>();

for(i=x;i<e.size();i++)

{

if((e.get(i).equals("+")==false)&&(e.get(i).equals("/")==false)&&(e.get(i).equals("-")==false)&&(e.get(i).equals("\*")==false))

{

for(j=i-1;j>e.indexOf("=");j--)

{

if((e.get(i).equals(e.get(j))==true))

break;

}

if(j == x - 1)

{

System.out.print("\nEnter value of "+e.get(i)+":");

no.add(Integer.parseInt(new BufferedReader(new InputStreamReader(System.in)).readLine()));

ans=no.get(k++).intValue();

j=0;

}

}

if(e.get(i-1).equals("+")==true)

ans+=no.get(k-2).intValue();

else if(e.get(i-1).equals("-")==true)

ans=no.get(k-2).intValue()-ans;

else if(e.get(i-1).equals("/")==true)

ans=no.get(k-2).intValue()/ans;

else if(e.get(i-1).equals("\*")==true)

ans\*=no.get(k-2).intValue();

}

System.out.println(e.get(0)+"="+ans);

}

}

