Exp-1:multi threading

class RunnableDemo implements Runnable {

private Thread t;

private String threadName;

RunnableDemo( String name) {

threadName = name;

System.out.println("Creating " + threadName );

}

public void run() {

System.out.println("Running " + threadName );

try {

for(int i = 4; i > 0; i--) {

System.out.println("Thread: " + threadName + ", " + i);

Thread.sleep(50);

}

} catch (InterruptedException e) {

System.out.println("Thread " + threadName + " interrupted.");

}

System.out.println("Thread " + threadName + " exiting.");

}

public void start () {

System.out.println("Starting " + threadName );

if (t == null) {

t = new Thread (this, threadName);

t.start ();

}

}

}

class Testthread {

public static void main(String args[]) {

RunnableDemo R1 = new RunnableDemo( "Thread-1");

R1.start();

RunnableDemo R2 = new RunnableDemo( "Thread-2");

R2.start();

}

}

Exp-2:scroll msg left to right

import java.awt.\*;

public class ScrollMessageApplet extends Frame implements Runnable {

private String message = "This is a scrolling message from left to right";

private Thread thread;

private boolean stopFlag;

public ScrollMessageApplet() {

setBackground(Color.black);

setForeground(Color.white);

setSize(400, 200);

setVisible(true);

}

public void start() {

thread = new Thread(this);

stopFlag = false;

thread.start();

}

public void run() {

while (!stopFlag) {

try {

Thread.sleep(100);

repaint();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

public void stop() {

stopFlag = true;

thread = null;

}

public void paint(Graphics g) {

int x = (int) (getWidth() - g.getFontMetrics().getStringBounds(message, g).getWidth()) / 2;

int y = getHeight() / 2;

g.clearRect(0, 0, getWidth(), getHeight());

g.drawString(message, x, y);

message = message.substring(1) + message.charAt(0);

}

public static void main(String[] args) {

ScrollMessageApplet applet = new ScrollMessageApplet();

applet.start();

}

}

Exp-3:split a give string

import java.util.Scanner;

public class TokenizeString {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String s = scanner.nextLine().trim();

scanner.close();

if (s.length() == 0) {

System.out.println(0);

} else {

String[] tokens = s.split("[^A-Za-z]+");

System.out.println(tokens.length);

for (String token : tokens) {

System.out.println(token);} } } }

Exp-4: print tha sums of negetive sub arrays of an array

import java.util.Scanner;

public class SubarraySum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the size of an array:");

int n = scanner.nextInt();

System.out.println("Enter the elements of the array:");

int[] array = new int[n];

for (int i = 0; i < n; i++) {

array[i] = scanner.nextInt();

}

scanner.close();

System.out.println("Negative subarrays and their sums:");

for (int i=0; i<n; i++) {

int sum = 0;

for (int j=i; j<n; j++) {

sum += array[j];

if (sum < 0) {

System.out.print("[");

for (int k=i; k<=j; k++) {

System.out.print(array[k]);

if (k < j) {

System.out.print(", ");

}

}

System.out.println("] - sum: " + sum);

} } } } }

Exp-5:nambiar number

import java.util.Scanner;

public class Number{

static String nambiarNumber(String str, int i)

{

if (i >= str.length())

return "";

int firstDigit = (str.charAt(i)-'0');

int digitParity = firstDigit % 2;

int sumDigits = 0;

while (i < str.length()) {

sumDigits += (str.charAt(i) - '0');

int sumParity = sumDigits % 2;

if (digitParity != sumParity) {

break;

}

i++;

}

return ("" + sumDigits + nambiarNumber(str, i + 1));

}

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

System.out.println("enter a mobile number:");

String str =obj.nextLine();

System.out.println(nambiarNumber(str,0));

}

}

Exp-6: simple calculator and perform some basic operations.

import javax.swing.\*;import java.awt.\*;import java.awt.event.\*;

public class Cal extends JFrame implements ActionListener {

String msg = "";

int v1, v2, result;

JTextField t1;

JButton b[] = new JButton[10];

JButton add, sub, mul, div, clear, mod, EQ;

char OP;

public Cal() {

setTitle("Calculator");

setLayout(new BorderLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

t1 = new JTextField(10);

t1.setHorizontalAlignment(SwingConstants.RIGHT);

t1.setEditable(false);

add(t1, BorderLayout.NORTH);

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 5));

for (int i = 0; i < 10; i++) {

b[i] = new JButton("" + i);

panel.add(b[i]);

b[i].addActionListener(this); }

add = new JButton("add");

sub = new JButton("sub");

mul = new JButton("mul");

div = new JButton("div");

mod = new JButton("mod");

clear = new JButton("clear");

EQ = new JButton("EQ");

panel.add(add);

panel.add(sub);

panel.add(mul);

panel.add(div);

panel.add(mod);

panel.add(clear);

panel.add(EQ);

add(panel, BorderLayout.CENTER);

for (int i = 0; i < 10; i++) {

b[i].addActionListener(this); }

add.addActionListener(this);

sub.addActionListener(this);

mul.addActionListener(this);

div.addActionListener(this);

mod.addActionListener(this);

clear.addActionListener(this);

EQ.addActionListener(this);

pack();

setLocationRelativeTo(null);

setVisible(true);

}

public void actionPerformed(ActionEvent ae) {

String str = ae.getActionCommand();

char ch = str.charAt(0);

if (Character.isDigit(ch))

t1.setText(t1.getText() + str);

else if (str.equals("add")) {

v1 = Integer.parseInt(t1.getText());

OP = '+';

t1.setText("");

} else if (str.equals("sub")) {

v1 = Integer.parseInt(t1.getText());

OP = '-';

t1.setText("");

} else if (str.equals("mul")) {

v1 = Integer.parseInt(t1.getText());

OP = '\*';

t1.setText("");

} else if (str.equals("div")) {

v1 = Integer.parseInt(t1.getText());

OP = '/';

t1.setText("");

} else if (str.equals("mod")) {

v1 = Integer.parseInt(t1.getText());

OP = '%';

t1.setText("");

} else if (str.equals("EQ")) {

v2 = Integer.parseInt(t1.getText());

if (OP == '+')

result = v1 + v2;

else if (OP == '-')

result = v1 - v2;

else if (OP == '\*')

result = v1 \* v2;

else if (OP == '/')

result = v1 / v2;

else if (OP == '%')

result = v1 % v2;

t1.setText("" + result);

} else if (str.equals("clear")) {

t1.setText(""); } }

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new Cal()); } }