Program 1

import java.util.Scanner;

public class Main

{

public static void main(String args[])

{

int i;

String str;

int counter[] = new int[256];

Scanner in = new Scanner(System.in);

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

str=in.nextLine();

for (i = 0; i < str.length(); i++) {

counter[(int) str.charAt(i)]++;

}

// Print Frequency of characters

for (i = 0; i < 256; i++) {

if (counter[i] != 0) {

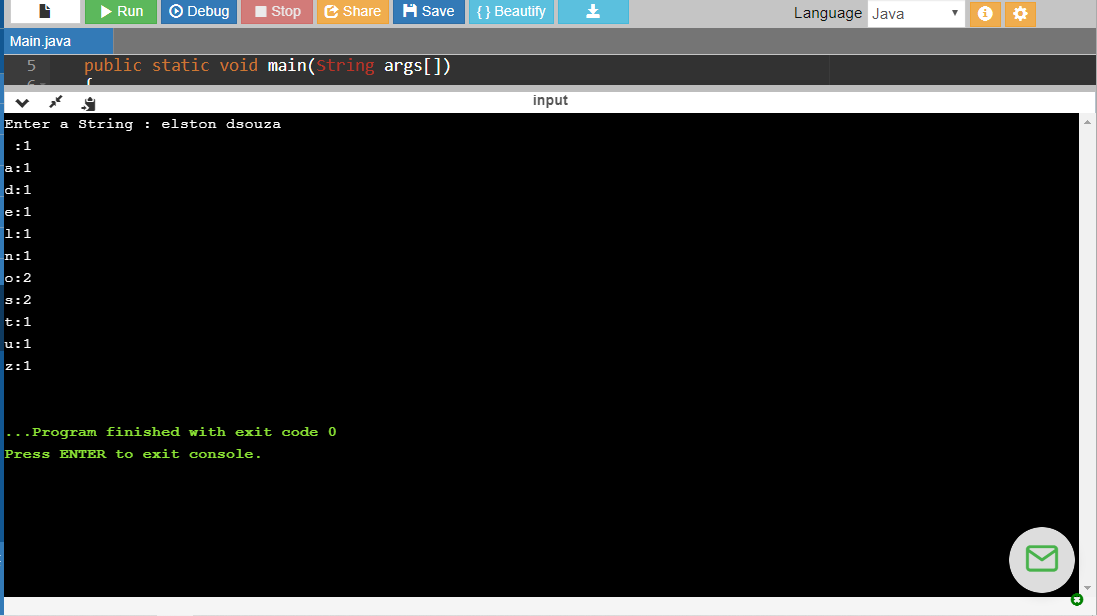
System.out.println((char) i + ":" + counter[i]);

}

}

}

}



Program 2

class OddThread extends Thread

{

int limit;

sharedPrinter printer;

public OddThread(int limit, sharedPrinter printer)

{

this.limit = limit;

this.printer = printer;

}

@Override

public void run()

{

int oddNumber = 1;

while (oddNumber <= limit)

{

printer.printOdd(oddNumber);

oddNumber = oddNumber + 2;

}

}

}

class EvenThread extends Thread

{

int limit;

sharedPrinter printer;

public EvenThread(int limit, sharedPrinter printer)

{

this.limit = limit;

this.printer = printer;

}

@Override

public void run()

{

int evenNumber = 2;

while (evenNumber <= limit)

{

printer.printEven(evenNumber);

evenNumber = evenNumber + 2;

}

}

}

class sharedPrinter

{

boolean isOddPrinted = false;

synchronized void printOdd(int number)

{

while (isOddPrinted)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName()+" "+number);

isOddPrinted = true;

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

notify();

}

synchronized void printEven(int number)

{

while (! isOddPrinted)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName()+" "+number);

isOddPrinted = false;

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

notify();

}

}

public class Main

{

public static void main(String[] args)

{

sharedPrinter printer = new sharedPrinter();

OddThread oddThread = new OddThread(20, printer);

oddThread.setName("—-pong");

EvenThread evenThread = new EvenThread(20, printer);

evenThread.setName("ping — >");

oddThread.start();

evenThread.start();

}

}

