

```

/*
 * Prof. Mauro De Berardis 2020
 * Il programma, implementato in vari linguaggi di programmazione e qui proposto in Java Swing, visualizza le combinazioni di n bit,
 * con n fornito in input e compreso tra 1 e 24. Viene calcolato il tempo di esecuzione.
 * L'algoritmo è quello utilizzato quando l'esercizio fu proposto nell'anno scolastico 2011-2012 agli alunni della classe IVBI
 * dell'ITS Alessandrini di Teramo e allora implementato in C++:
 * 1. si fornisce il valore di n
 * 2. si calcola il numero di combinazioni k pari a 2^n
 * 3. ciascun numero decimale da 0 a k-1 viene convertito in una stringa binaria che viene formattata aggiungendo uno o più zeri
 * iniziali in modo tale che la lunghezza sia sempre uguale ad n
 * Ad esempio: n=3 Numero di combinazioni: 2^3=8 Si convertono in binario i numeri decimali da 0 a 7 e si formattano a lunghezza 3
 * Risultato: 000 001 010 011 100 101 110 111
 */

```

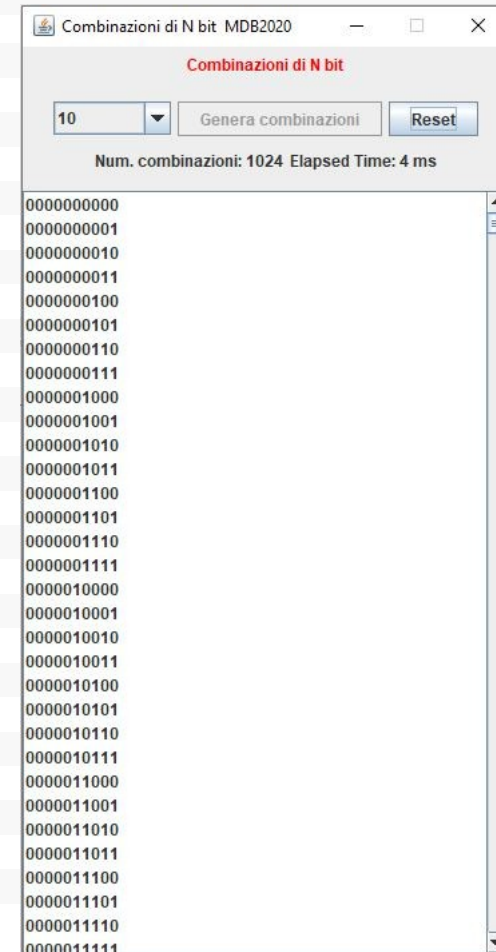
```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.JScrollPane;

import javax.swing.SwingConstants;

public class CombinazioniNbitSwing extends JFrame implements ActionListener{
    Container container;
    JPanel panel1;
    JComboBox<String> cb;
    JLabel lNumeroCombinazioni,lContatore;
    JButton bGenera,bReset;
    JList<String> list;

```



```

JScrollPane listacombinazioni;
String[] valoriN;
JList<String> lista;
DefaultListModel<String> listModel;
JProgressBar progressBar;
public CombinazioniNbitSwing() {
    valoriN=new String[25];
    valoriN[0]="Inserisci N";
    for(int i=1;i<=24;i++)
    {
        valoriN[i]=String.valueOf(i);
    }
    container=this.getContentPane();
    JPanel panel1=new JPanel(new GridLayout(3,1));
    JPanel p1=new JPanel();
    JPanel p2=new JPanel();
    JPanel p3=new JPanel();
    JLabel lTitolo=new JLabel("Combinazioni di N bit");
    lTitolo.setForeground(Color.RED);
    lTitolo.setHorizontalAlignment(SwingConstants.CENTER);
    p1.add(lTitolo);
    cb=new JComboBox<String>(valoriN);
    bGenera=new JButton("Genera combinazioni");
    bReset=new JButton("Reset");
    bGenera.addActionListener(this);
    bReset.addActionListener(this);
    p2.add(cb);p2.add(bGenera);p2.add(bReset);
    lNumeroCombinazioni=new JLabel("");
    lContatore=new JLabel("");
    p3.add(lNumeroCombinazioni);p3.add(lContatore);
    panel1.add(p1);
    panel1.add(p2);
    panel1.add(p3);
    container.add(panel1, BorderLayout.NORTH);
    listModel = new DefaultListModel<String>();
    lista=new JList<String>(listModel);
    listacombinazioni=new JScrollPane(lista);

```

```

container.add(listacombinazioni, BorderLayout.CENTER);
this.setTitle("Combinazioni di N bit MDB2020");
this.setResizable(false);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.pack();
this.setSize(380,720);
this.setVisible(true);
} // chiude costruttore-----
//
public static void main(String[] args) {
    new CombinazioniNbitSwing();
}
//

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getActionCommand()=="Genera combinazioni")
    {
        int nBit=cb.getSelectedIndex();
        if(nBit==0) return;
        //resetta();
        bGenera.setEnabled(false);
        double ncombinazioni=Math.pow(2.0, nBit);
        long nc=(long)ncombinazioni;
        lNumeroCombinazioni.setText("Num. combinazioni: "+String.valueOf(nc));
        long startTime = System.currentTimeMillis();
        for(int i=0;i<ncombinazioni;i++)
        {
            String binario=formatta(Integer.toString(i, 2),nBit);
            listModel.addElement(binario);
        }
        lista=new JList<>(listModel);
        long endTime = System.currentTimeMillis();
        long elapsedTime = endTime- startTime;
        if(elapsedTime<1) {
            lContatore.setText("< 1 ms");
        }else {

```

```

        lContatore.setText("Elapsed Time: "+String.valueOf(elapsedTime)+ " ms");
    }

    this.repaint();
}
if(e.getActionCommand()=="Reset")
{
    resetta();
}
} // chiude actionPerformed()-----
//
String formatta(String s, int l)
{
    String zeri="000000000000000000000000";
    return zeri.substring(0,l-s.length()+s);// 24 zeri utili per inserire gli zeri iniziali
}
//
Void resetta()
{
    cb.setSelectedIndex(0);
    lista.removeAll();
    listModel.removeAllElements();
    listModel.addElement("");
    lista=new JList<>(listModel);
    lNumeroCombinazioni.setText("");
    lContatore.setText("");
    bGenera.setEnabled(true);
    return null;
}
} // chiude CombinazioniNbitSwing-----

```