

```
/*Conversione da numero romano ad arabo con validazione dell'input
```

```
Conversione da numero arabo a romano con validazione dell'input
```

```
Il massimo numero arabo convertibile con questo programma è 3999
```

```
Algoritmo
```

```
Creo l'array 'romani[]' dei numeri romani corrispondenti ai numeri arabi da 0 a 3999.
```

```
Nella conversione arabo-romano il calcolo è già fatto. Ad esempio se
```

```
il numero da convertire è 2020, il numero romano corrispondente è romani[2020]
```

```
Nella conversione romano->arabo, controllo se il numero romano da convertire  
è presente nell'array 'romani[]'. Se è presente, il numero arabo corrispondente  
è l'indice, se non è presente vuol dire che il numero romano fornito in input  
non è corretto.
```

```
Ad es. se il numero romano è 'DCCII', lo trovo nell'array alla posizione (indice)
```

```
702 è 702 è il valore arabo corretto, se il numero romano è ad esempio "IIIIV",
```

```
non lo trovo nell'array e segnalo un errore di input */
```

```
import java.awt.*;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.*;
```

```
import javax.swing.border.Border;
```

```
import javax.swing.border.EmptyBorder;
```

```
public class RomaniArabi extends JFrame implements ActionListener{
```

```
    Container c;
```

```
    JTextField nromanoin,naraboin,nromanoout,naraboout;;
```

```
    JLabel labelout1,labelout2;
```

```
    JPanel pconv1;
```

```
    JButton bi,bv,bx,bl,bc,bd,bm,bdel,bres,bconv;
```

```
    JButton[] bnumeri=new JButton[10];
```

```
    JButton bdel2,bres2,bconv2;
```

```
    JRadioButton rb1,rb2;
```

```
    String[] m ={"", "M", "MM", "MMM"} ;
```

```
    String[] ce = {"", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"};
```

```
    String[] d = {"", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"};
```

```
    String[] u = {"", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"};
```

```
    String[] romani= new String[4000];
```

```
    public RomaniArabi() {
```

```
        /* utilizzando la funzione arabo_numero() genero l'array dei numeri
```

```
        Romani corrispondenti agli interi arabi da 0 a 3999 */
```

```
        for(int i=1;i<4000;i++)
```

```
        {
```

```
            romani[i]=arabo_romano(i);
```

```
        }
```

```

Container c=this.getContentPane();
c.setLayout(new GridLayout(1,2));
Font fs=new Font("Arial",Font.PLAIN,14);
JPanel p_sinistro=new JPanel(new GridLayout(7,1));
ButtonGroup bg=new ButtonGroup();
rb1=new JRadioButton("Da Romano (max 15 caratteri) ad Arabo",true);
rb2=new JRadioButton("Da Arabo (max 3999) a Romano");
rb1.addActionListener(this);
rb2.addActionListener(this);
bg.add(rb1);
bg.add(rb2);
nromanoin=new JTextField();
nromanoin.setBackground(Color.white);
nromanoin.setFont(fs);
nromanoin.setEditable(false);
nromanoin.setPreferredSize(new Dimension(200, 35));
nromanoin.setHorizontalAlignment(SwingConstants.CENTER);
JPanel p_sin_bottoni1=new JPanel(new GridLayout(1,7));
//Panel per i 7 bottoni dei cartatteri consentiti nei numeri romanit
bi=new JButton("I");
bv=new JButton("V");
bx=new JButton("X");
bl=new JButton("L");
bc=new JButton("C");
bd=new JButton("D");
bm=new JButton("M");
p_sin_bottoni1.add(bi); p_sin_bottoni1.add(bv);p_sin_bottoni1.add(bx);
p_sin_bottoni1.add(bl); p_sin_bottoni1.add(bc);
p_sin_bottoni1.add(bd); p_sin_bottoni1.add(bm);
JPanel p_sin_bottoni2=new JPanel(new GridLayout(1,3));
//Panel per i 3 bottoni Delete, Converti, Reset
bdel=new JButton("BS");
bres=new JButton("Reset");
bconv=new JButton("Converti");
p_sin_bottoni2.add(bdel);p_sin_bottoni2.add(bres);
p_sin_bottoni2.add(bconv);
JPanel p_sin_bottoni3=new JPanel(new GridLayout(1,3));
labelout1=new JLabel("Risultato");
naraboout=new JTextField();
naraboout.setBackground(Color.white);
naraboout.setFont(fs);
naraboout.setEditable(false);
naraboout.setPreferredSize(new Dimension(200, 35));
naraboout.setHorizontalAlignment(SwingConstants.CENTER);
// costruisco il panel p_sinistro
p_sinistro.add(rb1);
p_sinistro.add(nromanoin);
p_sinistro.add(p_sin_bottoni1);
p_sinistro.add(p_sin_bottoni2);
p_sinistro.add(p_sin_bottoni3);
p_sinistro.add(labelout1);
p_sinistro.add(naraboout);
bi.addActionListener(this);
bv.addActionListener(this);
bx.addActionListener(this);

```

```

bl.addActionListener(this);
bc.addActionListener(this);
bd.addActionListener(this);
bm.addActionListener(this);
bdel.addActionListener(this);
bres.addActionListener(this);
bconv.addActionListener(this);
//-----Panel destro-----
JPanel p_destro=new JPanel(new GridLayout(7,1));
naraboin=new JTextField();
naraboin.setBackground(Color.white);
naraboin.setFont(fs);
naraboin.setEditable(false);
naraboin.setPreferredSize(new Dimension(200, 35));
naraboin.setHorizontalAlignment(SwingConstants.CENTER);
JPanel p_des_bottoni1=new JPanel(new GridLayout(1,5));
for(int i=0;i<=4;i++){
    bnumeri[i]=new JButton(String.valueOf(i));
    p_des_bottoni1.add(bnumeri[i]);
    bnumeri[i].addActionListener(this);
}
JPanel p_des_bottoni2=new JPanel(new GridLayout(1,5));
for(int i=5;i<=9;i++){
    bnumeri[i]=new JButton(String.valueOf(i));
    p_des_bottoni2.add(bnumeri[i]);
    bnumeri[i].addActionListener(this);
}
JPanel p_des_bottoni3=new JPanel(new GridLayout(1,3));
bdel2=new JButton("BS");
bres2=new JButton("Reset");
bconv2=new JButton("Converti");
p_des_bottoni3.add(bdel2); p_des_bottoni3.add(bres2);
p_des_bottoni3.add(bconv2);
labelout2=new JLabel("Risultato");
nromanoout=new JTextField();
nromanoout.setBackground(Color.white);
nromanoout.setFont(fs);
nromanoout.setEditable(false);
nromanoout.setPreferredSize(new Dimension(200, 35));
nromanoout.setHorizontalAlignment(SwingConstants.CENTER);
// costruisco il panel p_destro
p_destro.add(rb2);
p_destro.add(naraboin);
p_destro.add(p_des_bottoni1);
p_destro.add(p_des_bottoni2);
p_destro.add(p_des_bottoni3);
p_destro.add(labelout2);
p_destro.add(nromanoout);
bdel2.addActionListener(this);
bres2.addActionListener(this);
bconv2.addActionListener(this);
// Margini dei 2 pannelli che saranno aggiunti al contenitore
Border margine = new EmptyBorder(0,5,5, 5);
p_sinistro.setBorder(margine);
p_destro.setBorder(margine);

```

```

        c.add(p_sinistro);
        c.add(p_destro);
        impostaneldestro("D");//di default il panel destro è Disabled
        this.setTitle("Convertitore da numeri romani ad arabi e viceversa MDB2
020");
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();//
        this.setSize(700,250);
        this.setVisible(true);

    }//chiude costruttore-----

    public static void main(String[] args) {
        new RomaniArabi();
    }//chiude main()-----

    String arabo_romano(int numero)
    {
        int qm = (int)(numero / 1000); //migliaia
        int qc = (int)((numero - qm * 1000) / 100); // centinaia
        int qd = (int)((numero - qm * 1000 - qc * 100) / 10); // decine
        int qu = (int)(numero - qm * 1000 - qc * 100 - qd * 10); //unità
        String mm = m[qm];
        String cc = ce[qc];
        String dd = d[qd];
        String uu = u[qu];
        return (mm + cc + dd + uu);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String az=e.getActionCommand(); // azione da fare
        Object azione=e.getSource();
        switch(az) {
            case "Da Romano (max 15 caratteri) ad Arabo":
                impostanelsinistro("A"); //abilita panel sinistro
                impostaneldestro("D");
                break;
            case "Da Arabo (max 3999) a Romano":
                impostanelsinistro("D");
                impostaneldestro("A");//abilita panel destro
                break;
            case "I":
                if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"I");break;
            case "V":
                if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"V");break;
            case "X":
                if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"X");break;
            case "L":
                if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"L");break;

```

```

        case "C":
            if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"C");break;
        case "D":
            if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"D");break;
        case "M":
            if(nromanoin.getText().length()<15)nromanoin.setText(nromanoin.getT
ext()+"M");break;
        case "0":
            if(naraboin.getText().length()==0)return;
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"0");break;
        case "1":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"1");break;
        case "2":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"2");break;
        case "3":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"3");break;
        case "4":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"4");break;
        case "5":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"5");break;
        case "6":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"6");break;
        case "7":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"7");break;
        case "8":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"8");break;
        case "9":
            if(naraboin.getText().length()<4)naraboin.setText(naraboin.getText(
)+"9");break;
    } //---chiude switch-----
    //
    // per i casi in cui il comando non è univoco
    if(azione==bdel)
    {
        naraboout.setText("");
        if(nromanoin.getText().equals(""))return;
        int l=nromanoin.getText().length();
        nromanoin.setText(nromanoin.getText().substring(0,l-1));
    }
    if(azione==bdel2)
    {
        nromanoout.setText("");
        if(naraboin.getText().equals(""))return;
        int l=naraboin.getText().length();

```

```

        naraboin.setText(naraboin.getText().substring(0,l-1));
    }
    if(azione==bres)
    {
        nromanoin.setText("");
        naraboout.setText("");
    }
    if(azione==bres2)
    {
        naraboin.setText("");
        nromanoout.setText("");
    }

    if(azione==bconv)
    {
        if(nromanoin.getText().equals(""))return;
        for(int i=1;i<4000;i++)
        {
            if(nromanoin.getText().equals(romani[i]))
            {
                naraboout.setText(String.valueOf(i));
                return;
            }
        }
        String testo="Numero romano non corretto!";
        JOptionPane.showMessageDialog(this, testo, "Info MDB2020", JOptionPane.PLAIN_MESSAGE);
    }
    if(azione==bconv2)
    {
        if(naraboin.getText().equals(""))return;
        int num=Integer.parseInt(naraboin.getText());
        if(num>3999)
        {
            String testo="Numero arabo troppo grande!";
            JOptionPane.showMessageDialog(this, testo, "Info MDB2020", JOptionPane.PLAIN_MESSAGE);
        }
        nromanoout.setText(romani[num]);
    }

} // chiude actionPerformed()-----

void impostapanelstinistro(String p)
{
    boolean bool=true;
    if(p.contentEquals("D")) bool=false;
    bi.setEnabled(bool);
    bv.setEnabled(bool);
    bx.setEnabled(bool);
    bl.setEnabled(bool);
    bc.setEnabled(bool);
    bd.setEnabled(bool);
    bm.setEnabled(bool);
    bdel.setEnabled(bool);
}

```

```
        bres.setEnabled(bool);
        bconv.setEnabled(bool);
        labelout1.setEnabled(bool);
        nromanoin.setEnabled(bool);
        naraboout.setEnabled(bool);
    }
    void impostapaneldestro(String p) {
        boolean bool=true;
        if(p.contentEquals("D")) bool=false;
        for(int i=0;i<=9;i++)
        {
            bnumeri[i].setEnabled(bool);
            bdel2.setEnabled(bool);
            bres2.setEnabled(bool);
            bconv2.setEnabled(bool);
            labelout2.setEnabled(bool);
            naraboin.setEnabled(bool);
            nromanoout.setEnabled(bool);
        }
    }
}
} //-----chiude class-----
```