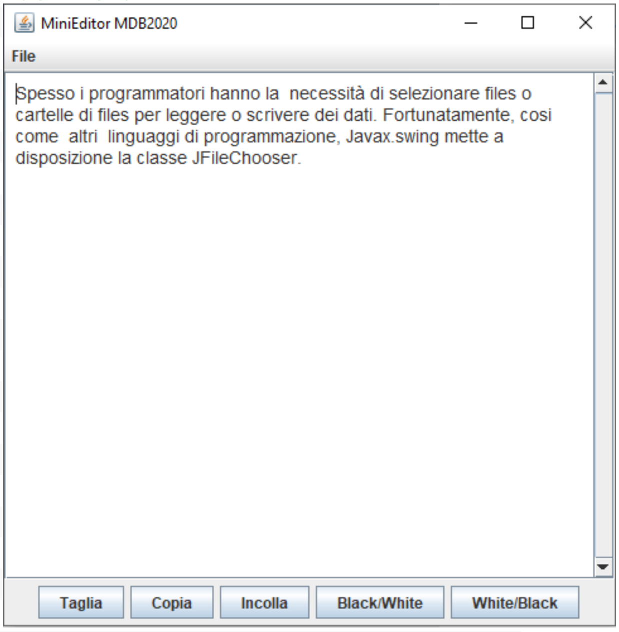
```java
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;

public class MiniEditor extends JFrame implements ActionListener{

    Container c;
    JTextArea testo;
    JScrollPane scroll;
    JButton taglia,copia,incolla,nerobianco,bianconero;

    public MiniEditor() {
        c=this.getContentPane();
        testo=new JTextArea();
        testo.setFont(new Font("Arial", Font.PLAIN, 14));
        testo.setMargin(new Insets(8, 8, 8, 8));

        //imposto la modalità di auto-wrap tenendo conto
        //delle parole
        testo.setWrapStyleWord(true);
        testo.setLineWrap(true);
        JPanel ptesto=new JPanel();
        ptesto.setLayout(new BorderLayout());
        scroll = new JScrollPane(testo);
        scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);;
        ptesto.add(scroll);
        JMenuBar barramenu=new JMenuBar();//barra del menù
        JMenu menufile=new JMenu("File");
        JMenuItem it1=new JMenuItem("Nuovo");
        JMenuItem it2=new JMenuItem("Apri");
        JMenuItem it3=new JMenuItem("Salva");
        menufile.add(it1);
        menufile.add(it2);
        menufile.add(it3);
        it1.addActionListener(this);
        it2.addActionListener(this);
        it3.addActionListener(this);
        taglia=new JButton("Taglia");
        copia=new JButton("Copia");
        incolla=new JButton("Incolla");
        nerobianco=new JButton("Black/White");
        bianconero=new JButton("White/Black");
        taglia.addActionListener(this);
        copia.addActionListener(this);
        incolla.addActionListener(this);
        nerobianco.addActionListener(this);
        bianconero.addActionListener(this);
        // se si vuole usare un bottone con icona si può utilizzare il metodo
        // setIcon(immagine)
        // ad es. taglia.setIcon(new ImageIcon("c:/prova/forbici.png"));
        JPanel panelBottoni=new JPanel();
        panelBottoni.add(taglia);
        panelBottoni.add(copia);
        panelBottoni.add(incolla);
        panelBottoni.add(nerobianco);
        panelBottoni.add(bianconero);
        c.add(ptesto, BorderLayout.CENTER);
        c.add(panelBottoni, BorderLayout.PAGE_END);
```



Spesso i programmatori hanno la necessità di selezionare files o cartelle di files per leggere o scrivere dei dati. Fortunatamente, cosi come altri linguaggi di programmazione, Javax.swing mette a disposizione la classe JFileChooser.

```
        barramenu.add(menufile);//add menufile alla barra dei menu
        this.setJMenuBar(barramenu);
        this.setTitle("MiniEditor MDB2020");
        //this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.pack();//
        this.setSize(500, 500);
        this.setVisible(true);
    }// chiude costruttore--------

    public static void main(String[] args) {
        new MiniEditor();

    }//chiude main()----------------------------------

    // processo gli eventi
    @Override
    public void actionPerformed(ActionEvent e) {
        String action=e.getActionCommand();
        char c0 = action.charAt(0);
        FileNameExtensionFilter filter = new FileNameExtensionFilter("Files di testo
", "txt", "text");
        switch(c0) {
        case 'T':
            testo.cut();        break;
        case 'C':
            testo.copy();       break;
        case 'I':
            testo.paste();      break;
        case 'B':
            testo.setBackground(Color.WHITE);
            testo.setForeground(Color.BLACK);
            testo.requestFocus();
            break;
        case 'W':
            testo.setBackground(Color.BLACK);
            testo.setForeground(Color.WHITE);
            testo.requestFocus();
            break;
        case 'S':
          if(testo.getText().equals("")) return;
            //creo l'oggetto fcw della classe JFileChooser e
            //invoco il metodo showSaveDialog(null)
            JFileChooser fcw = new JFileChooser();
            fcw.setFileFilter(filter);
            int r1 = fcw.showSaveDialog(null);
            if(r1==JFileChooser.APPROVE_OPTION) {
                  File fw = fcw.getSelectedFile();
                  try {
                    BufferedWriter bwrite = new BufferedWriter(new FileWriter(fw));
                    bwrite.append(testo.getText());
                    bwrite.flush();
                    bwrite.close();
                  }
                    catch (Exception exc) {
                        JOptionPane.showMessageDialog(this, exc.getMessage());
                 }
            }
            break;
          case 'A':
            //creo l'oggetto fco della classe JFileChooser e
            //invoco il metodo showOpenDialog(null)
            JFileChooser fco = new JFileChooser();
            fco.setFileFilter(filter);
            int r2 = fco.showOpenDialog(null);
            if(r2==JFileChooser.APPROVE_OPTION) {
                testo.setText("");//pulisco la TextArea
                File fo=fco.getSelectedFile();
                try {
                      BufferedReader bread = new BufferedReader(new FileReader(fo));

                      String line = bread.readLine();
                      while(line != null) {
                        testo.append(line);
                        line = bread.readLine();
                      }
                      bread.close();

                  } catch (Exception exc) {
                      JOptionPane.showMessageDialog(this, exc.getMessage());
                  }
            }
            break;
        case 'N':
            testo.setText("");//pulisco la TextArea
            break;

        }// chiude switch(c0)-------------

    }

}//---------------------------------chiude class MiniEditor-------------------
```