Модель таблицы:

```
import java.io.File;
import java.util.*;
import javax.swing.JOptionPane;
import javax.swing.table.AbstractTableModel;
//Модель таблицы
public class FileTable extends AbstractTableModel{
      private static final String[] columnNames = {"Имя файла","Размер","Дата последней
модификации","Доступ","Расширение","Тип"};
      private static final Class[] columnClasses = {String.class, String.class, String.class,
String.class, String.class, String.class};
      public static ArrayList<File> files = new ArrayList<File>();
      public String path = "C:\\";
      //Добавление файла
      public void addFile(File file) {
        files.add(file);
        fireTableRowsInserted(getRowCount() - 1, getRowCount() - 1);
    }
      //Получение файла
      public File getFile(int row) {
            if (row < 0 | row>files.size()) {
                  return null;
            } else {
        return files.get(row);
            }
    }
      //Очистка таблицы
      public void clearTable() {
            if (files.size()==0) {

            } else {
            fireTableRowsDeleted(0,files.size()-1);
          files.clear();
            }
      }
      //Очищение определённой строки
      public void clearFile(int row) {
        files.remove(row);
        fireTableRowsDeleted(row,row);
    }
      //Получение имён столбцов
      public String getColumnName(int col) {
        return columnNames[col];
    }
      //Получение классов столбцов
    public Class getColumnClass(int col) {
        return columnClasses[col];
    }
    //Получение количества столбцов
      public int getColumnCount() {
            return columnNames.length;
      }
      //получение количество строк
      public int getRowCount() {
            return files.size();
      }
      //получение значения из ячейки [row,col]
      public Object getValueAt(int row, int col) {
            File f1 = files.get(row);
        switch (col) {
            case 0: return f1.getName();
```

```
        case 1: long size = f1.length();
         String res;
         if (size >=1024) {
                if (size >= 1024*1024) {
                      if (size >= 1024*1024*1024) {
                            if (size >= 1024*1024*1024*1024) {
                                  size=(long) size/1024;
                                  size=(long) size/1024;
                                  size=(long) size/1024;
                                  size=(long) size/1024;
                                  res = Long.toString(size) + " ТБ";
                                  return res;
                            }
                            size=(long) size/1024;
                                  size=(long) size/1024;
                                  size=(long) size/1024;
                                  res = Long.toString(size) + " ГБ";
                                  return res;
                      }
                      size=(long) size/1024;
                            size=(long) size/1024;
                            res = Long.toString(size) + " МБ";
                            return res;
                }
                size=(long) size/1024;
                res = Long.toString(size) + " КБ";
         }
             if (size == -1) {
                 return "";
             } else {
                 res = Long.toString(size) + " Б";
             }
             return res;
        case 2: Date date = new Date(f1.lastModified());
        Formatter fmt = new Formatter();
        fmt.format("%tH:%tM:%tS %td %tB %tY", date,date,date,date,date,date);
        return fmt;
        case 3:String access = "";
        if (f1.canRead()) { access+="r"; } else { access+="-"; }
        if (f1.canWrite()) { access+="w"; } else { access+="-"; }
        if (f1.canExecute()) { access+="x"; } else {access += "-"; }
        return access;
        case 4: if (f1.isFile()) {
         String name = f1.getName();
         if (name.contains(".")) {
         name = name.substring(name.lastIndexOf('.')+1, name.length());
         return name;
         } else {
                return "-";
         }
        } else {
         return "-";
        }
        case 5: if (f1.isFile()) { return "Файл"; } else {return "Директория";}
      }
      return "";
    }
}
```

Графический интерфейс:

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.nio.file.*;
import javax.swing.event.*;
//Графический интерфейс
public class GUI extends JFrame{
	private JTable table;
	public static FileTable tableModel;
	File f1;
	JLabel Lpath;
	Path copyPath;
	JButton open, go, copy, paste, delete, edit, run,create, createDir, back;
	JMenuBar menuBar = new JMenuBar();
    JMenu fileMenu = new JMenu("Файл");
    JMenu help = new JMenu("Помощь");
    JMenuItem exit = new JMenuItem("Выход", KeyEvent.VK_X);
    JMenuItem about = new JMenuItem("О программе", KeyEvent.VK_H);
    //Конструктор
	GUI() {
		setTitle("DFM");
		setSize(1124, 768);
		addWindowListener(new WindowAdapter() {
		public void windowClosing(WindowEvent e) {
			System.exit(0);
		}
	 });
		fileMenu.add(exit);
		help.add(about);
		//Обработчик событий по нажатию на пунтк меню - Выход
		exit.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				System.exit(0);
			}
		});
		//Обработчик событий по нажатию на пунтк меню - О программе
		about.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				JOptionPane.showMessageDialog(null, "Автор: DenSoLo777");
			}
		});
		menuBar.add(fileMenu);
		menuBar.add(help);
		setJMenuBar(menuBar);
		JPanel PathPanel = new JPanel();
		Lpath = new JLabel("");
		PathPanel.add(Lpath);
		tableModel = new FileTable();
		table = new JTable(tableModel);
		//Добавление обработчика события выделение строки
	table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
	table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
		public void valueChanged(ListSelectionEvent e) {
			tableSelectionChanged();
		}
	});
	JPanel filesPanel = new JPanel();
	filesPanel.setBorder(BorderFactory.createTitledBorder("Файлы"));
	filesPanel.setLayout(new BorderLayout());
	filesPanel.add(new JScrollPane(table),BorderLayout.CENTER);
	createDir = new JButton("Создать каталог");
```

```
//Создание каталога
createDir.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
		String str = JOptionPane.showInputDialog("Введите имя директории:");
		String path = Lpath.getText();
		path+=" ";
		if (path.contains(":\\ ")) {
			path=path.trim()+str;
			File f1 = new File(path);
			f1.mkdir();
		} else {
		File f1 = new File(Lpath.getText()+"\\"+str);
		f1.mkdir();
		}
		tableModel.addFile(f1);
	}
});
createDir.setEnabled(true);
create = new JButton("Создать файл");
//Создание файла
create.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
		String name = JOptionPane.showInputDialog("Введите имя файла:");
		String path=Lpath.getText();
		path+=" ";
		if (path.contains(":\\ ")) {
			path=path.trim();
			path=path+name;
			File f1 = new File(path);
			try {
				f1.createNewFile();
			} catch (IOException e1) {
				e1.printStackTrace();
			}
			tableModel.addFile(f1);
		} else {
			path=path.trim();
			path=path+"\\"+name;
			File f1 = new File(path);
			try {
				f1.createNewFile();
			} catch (IOException e1) {
				e1.printStackTrace();
			}
		}
	}
});
create.setEnabled(true);
delete = new JButton("Удалить");
//удаление файла/директории
delete.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
		if (table.getSelectedRow() == -1) {
			JOptionPane.showMessageDialog(null, "Файл не выбран");
		} else {
		File f1 = tableModel.getFile(table.getSelectedRow());
		if (f1.isDirectory()) {
		deleteDir(f1);
		} else {
			if (f1.delete()) {
			JOptionPane.showMessageDialog(null, "Файл успешно удалён.");
			tableModel.clearFile(table.getSelectedRow());
```

```
                } else {
                        JOptionPane.showMessageDialog(null, "Удалить файл не удалось.");
                }
                }
        }
        }
});
delete.setEnabled(false);
copy = new JButton("Копировать");
//копирование файла
copy.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                if (table.getSelectedRow() == -1) {
                        JOptionPane.showMessageDialog(null, "Файл не выбран");
                } else {
                        copyPath = tableModel.getFile(table.getSelectedRow()).toPath();
                }
        }
});
copy.setEnabled(false);
run = new JButton("Запустить");
//запуск exe-файла
run.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                if (table.getSelectedRow() == -1) {
                        JOptionPane.showMessageDialog(null, "Файл не выбран");
                } else {
                        File f1 = tableModel.getFile(table.getSelectedRow());
                        try {
                                        Runtime.getRuntime().exec(f1.getPath());
                                } catch (IOException e1) {
                                        JOptionPane.showMessageDialog(null, "Не удалось запустить
файл");
                                }
        }
        }
});
run.setEnabled(false);
open = new JButton("Открыть");
open.setEnabled(false);
//Открытие файла
open.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        if (table.getSelectedRow() == -1) {
                                JOptionPane.showMessageDialog(null, "Файл не выбран");
                        } else {
                        File f1 = tableModel.getFile(table.getSelectedRow());
                        String str = (String) tableModel.getValueAt(table.getSelectedRow(), 4);
                        switch (str) {
                        case "jpg":
                                Iframe f = new Iframe(f1);
                                f.setVisible(true);
                                break;
                        case "bmp":
                                Iframe fr = new Iframe(f1);
                                fr.setVisible(true);
                        case "png":
                                Iframe fra = new Iframe(f1);
                                fra.setVisible(true);
                        case "gif":
                                Iframe fram = new Iframe(f1);
                                fram.setVisible(true);
```

```
                        case "txt":
                              TextEditor te = new TextEditor(f1);
                              te.loadText(f1);
                              try {
                                    Files.deleteIfExists(f1.toPath());
                              } catch (IOException e1) {
                                    e1.printStackTrace();
                              }
                              tableModel.clearFile(table.getSelectedRow());
                              break;
                        case "html":
                              TextEditor te2 = new TextEditor(f1);
                              te2.loadText(f1);
                              try {
                                    Files.deleteIfExists(f1.toPath());
                              } catch (IOException e1) {
                                    e1.printStackTrace();
                              }
                              tableModel.clearFile(table.getSelectedRow());
                              break;
                        }
                        }
                  }
            });
            paste = new JButton("Вставить");
            //вставка файла
            paste.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        File f1 = copyPath.toFile();
                        String path = Lpath.getText();
                        path+=" ";
                        File f2;
                        if (path.contains(":\\ ")) {
                              path=path.trim();
                              path+=f1.getName();
                              f2 = new File(path);
                        } else {
                              path=path.trim();
                              path=path+"\\"+f1.getName();
                              f2 = new File(path);
                        }
                        try {
                                    Files.copy(f1.toPath(), f2.toPath(),
StandardCopyOption.REPLACE_EXISTING);
                                    tableModel.addFile(f2);
                              } catch (IOException e1) {
                                    e1.printStackTrace();
                              }
                  }
            });
            paste.setEnabled(false);
            go = new JButton("Перейти");
            //переход в выделенную директорию
            go.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        if (table.getSelectedRow() == -1) {

                        } else {
                        String path = Lpath.getText();
                        path+=" ";
                        String target;
                        String newPath;
```

```
                if (path.contains(":\\ ")) {
                        path = path.trim();
                        target = path.concat(tableModel.getFile(table.getSelectedRow()).getName());
                        Lpath.setText(target);
                        File f1 = new File(target);
                        String s[] = f1.list();
                        tableModel.clearTable();
                        for (int i = 0; i<s.length; i++) {
                                newPath=target.trim()+"\\"+s[i];
                                tableModel.addFile(new File(newPath));
                        }
                } else {
                        target =
path.trim().concat("\\".concat(tableModel.getFile(table.getSelectedRow()).getName()));
                        Lpath.setText(target);
                        File f1 = new File(target);
                        String s[] = f1.list();
                        tableModel.clearTable();
                        back.setEnabled(true);
                        for (int i = 0; i<s.length; i++) {
                                newPath=target.trim()+"\\"+s[i];
                                tableModel.addFile(new File(newPath));
                        }
                }
            }
            }
        });
        go.setEnabled(false);
        back = new JButton("Назад");
        //переход в родительский каталог
        back.addActionListener(new ActionListener() {
             public void actionPerformed(ActionEvent e) {
                File f1 = new File(Lpath.getText());
                        String path = f1.getParent();
                        File f2 = new File(f1.getParent());
                        String target;
                        String s[] = f2.list();
                        tableModel.clearTable();
                        Lpath.setText(path);
                        for (int i = 0;i<s.length; i++) {
                                tableModel.addFile(new File(Lpath.getText()+"\\"+s[i]));
                        }
                        path+=" ";
                        if (path.contains(":\\ ")) {
                                back.setEnabled(false);
                        } else {
                                back.setEnabled(true);
                        }
            }
        });
        back.setEnabled(false);
        edit = new JButton("Изменить путь");
        //Изменение пути
        edit.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent arg0) {
                        Lpath.setText(JOptionPane.showInputDialog("Введите путь:"));
                        tableModel.clearTable();
                        String path = Lpath.getText();
                        path+=" ";
                        if (path.contains(":\\ ")) {
                                File f2 = new File(path);
                                String s[] = f2.list();
```

```
                        back.setEnabled(false);
                    for (int i = 1; i<s.length; i++) {
                        String target = path.trim().concat(s[i]);
                        tableModel.addFile(new File(target));
                    }
                    } else {
                        File f2 = new File(path);
                        String s[] = f2.list();
                        back.setEnabled(true);
                        for (int i = 1; i<s.length; i++) {
                            String target = path.trim().concat("\\".concat(s[i]));
                            tableModel.addFile(new File(target));
                        }
                    }
                }
  });
  edit.setEnabled(true);
  JPanel buttonPanel = new JPanel();
  buttonPanel.add(back);
  buttonPanel.add(createDir);
  buttonPanel.add(create);
  buttonPanel.add(delete);
  buttonPanel.add(copy);
  buttonPanel.add(paste);
  buttonPanel.add(run);
  buttonPanel.add(edit);
  buttonPanel.add(open);
  buttonPanel.add(go);
  getContentPane().setLayout(new BorderLayout());
  getContentPane().add(PathPanel, BorderLayout.NORTH);
  getContentPane().add(filesPanel, BorderLayout.CENTER);
  getContentPane().add(buttonPanel, BorderLayout.SOUTH);
  Lpath.setText("C:\\");
  File file = new File(Lpath.getText());
  String newFiles[] = file.list();
  for (int i =0; i <newFiles.length; i++) {
       tableModel.addFile(new File(Lpath.getText()+newFiles[i]));
  }
}
//метод удаления
void deleteDir(File f) {
      String s[] = f.list();
      File f2;
      if (s == null) {
            try {
                  Files.delete(f.toPath());
            } catch (IOException e) {
                  e.printStackTrace();
            }
      }
      for (int i=0; i<s.length;i++) {
            f2 = new File(f.getPath()+"\\"+s[i]);
            if (f2.isFile()) {
                  try {
                        Files.delete(f2.toPath());
                  } catch (IOException e) {
                        JOptionPane.showMessageDialog(null, "Не удалось удалить файл");
                  }
            } else {
                  del(f2);
                  try {
                        Files.delete(f2.toPath());
```

```
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }
        try {
                Files.delete(f.toPath());
        } catch (IOException e) {
                e.printStackTrace();
        }
        tableModel.clearFile(table.getSelectedRow());
        JOptionPane.showMessageDialog(null, "Файл успешно удалён");
}
//метод удаления
void del(File f) {
        String s[] = f.list();
        File f2;
        for (int i=0; i<s.length;i++) {
                f2 = new File(f.getPath()+"\\"+s[i]);
                if (f2.isFile()) {
                        try {
                                Files.delete(f2.toPath());
                        } catch (IOException e) {
                                JOptionPane.showMessageDialog(null, "Не удалось удалить файл");
                        }
                } else {
                        del(f2);
                        try {
                                Files.delete(f2.toPath());
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }
}
//изменение состояния кнопок
protected void tableSelectionChanged() {
        f1 = tableModel.getFile(table.getSelectedRow());
        if (f1 == null) {

        } else {
        if (f1.isFile()) {
                String str = (String) tableModel.getValueAt(table.getSelectedRow(), 4);
                str=str.toLowerCase();
                switch (str) {
                        case "jpg":
                                delete.setEnabled(true);
                                open.setEnabled(true);
                                copy.setEnabled(true);
                                paste.setEnabled(true);
                                go.setEnabled(false);
                                run.setEnabled(false);
                                break;
                        case "bmp":
                                delete.setEnabled(true);
                                open.setEnabled(true);
                                copy.setEnabled(true);
                                paste.setEnabled(true);
                                go.setEnabled(false);
                                run.setEnabled(false);
                                break;
                        case "png":
```

```
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
                case "gif":
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
                case "txt":
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
                case "exe":
                        delete.setEnabled(true);
                        open.setEnabled(false);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(true);
                        break;
                case "doc":
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
                case "html":
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
                case "java":
                        delete.setEnabled(true);
                        open.setEnabled(true);
                        copy.setEnabled(true);
                        paste.setEnabled(true);
                        go.setEnabled(false);
                        run.setEnabled(false);
                        break;
        }

} else {
        delete.setEnabled(true);
        open.setEnabled(false);
        copy.setEnabled(true);
        paste.setEnabled(true);
```

```
                go.setEnabled(true);
                run.setEnabled(false);
                if (f1.getParent() !=null) {
                      back.setEnabled(true);
                } else {
                      back.setEnabled(false);
                }
          }
          }
    }

    public static void main(String[] args) {
          SwingUtilities.invokeLater(new Runnable() {
         public void run() {
             GUI manager = new GUI();
             manager.setVisible(true);
          }
      });
    }
}
```

Просмотровщик изображений:

```
import java.awt.*;
import java.io.*;
import javax.swing.*;
public class Iframe extends Frame{
    ImageIcon ic;
    myWinAdapter mwa = new myWinAdapter(this);
    public Iframe(File f) {
          ic = new ImageIcon(f.getPath());
          setTitle(f.getName());
          setSize(ic.getIconWidth(), ic.getIconHeight()+22);
          setResizable(false);
          addWindowListener(mwa);
    }
    public void paint(Graphics g) {
          ic.paintIcon(this, g, 0, 22);
    }
}
```

Оконный адаптер:

```
import java.awt.event.*;
import java.awt.*;
public class myWinAdapter extends WindowAdapter {
    Frame f;
    myWinAdapter(Frame fr) {
          f = fr;
    }
    public void windowClosing(WindowEvent e) {
          f.setVisible(false);
    }
}
```

Текстовый редактор:

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.nio.file.Files;

import javax.swing.*;
public class TextEditor extends JFrame{
	JScrollPane sp = new JScrollPane();
	static JTextPane text = new JTextPane();
	JMenuBar menuBar = new JMenuBar();
    JMenu fileMenu = new JMenu("Файл");
    JMenuItem save = new JMenuItem("Сохранить", KeyEvent.VK_S);
    static File f1;
    //Конструктор
	TextEditor(File f1) {
		this.f1 = f1;
		save.addActionListener(new ActionListener () {
			public void actionPerformed(ActionEvent e) {
			saveText(TextEditor.f1);
		  }
		});
		fileMenu.add(save);
	  menuBar.add(fileMenu);
	  setJMenuBar(menuBar);
		setTitle("Text-Editor");
		setSize(800,600);
		addWindowListener(new WindowAdapter() {
			public void windowClosing(WindowEvent arg0) {
				saveText(TextEditor.f1);
			}
		});
		sp.setViewportView(text);
		sp.setBounds(0, 0, 800, 550);
		getContentPane().add(sp);
	  setVisible(true);
	  setResizable(false);
	}
	//загрузка текста
	void loadText(File f1) {
		try {
			int i = 0;
			FileReader fr = new FileReader(f1);
			BufferedReader br = new BufferedReader(fr);
			String line;
			while ((line = br.readLine()) != null) {
				if (i == 0) {
				text.setText(text.getText()+line);
				} else {
					text.setText(text.getText()+"\n"+line);
				}
			}
			br.close();
			fr.close();
		} catch (FileNotFoundException e) {
			e.printStackTrace();
		} catch (IOException e) {
			e.printStackTrace();
		}
	}
	//Сохранение текста
	public static void saveText(File f1) {
		try {
```

```
                f1.createNewFile();
                FileWriter fw = new FileWriter(f1);
                BufferedWriter bw = new BufferedWriter(fw);
                bw.write(text.getText());
                bw.close();
                GUI.tableModel.addFile(f1);
                text.setText("");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
}
```