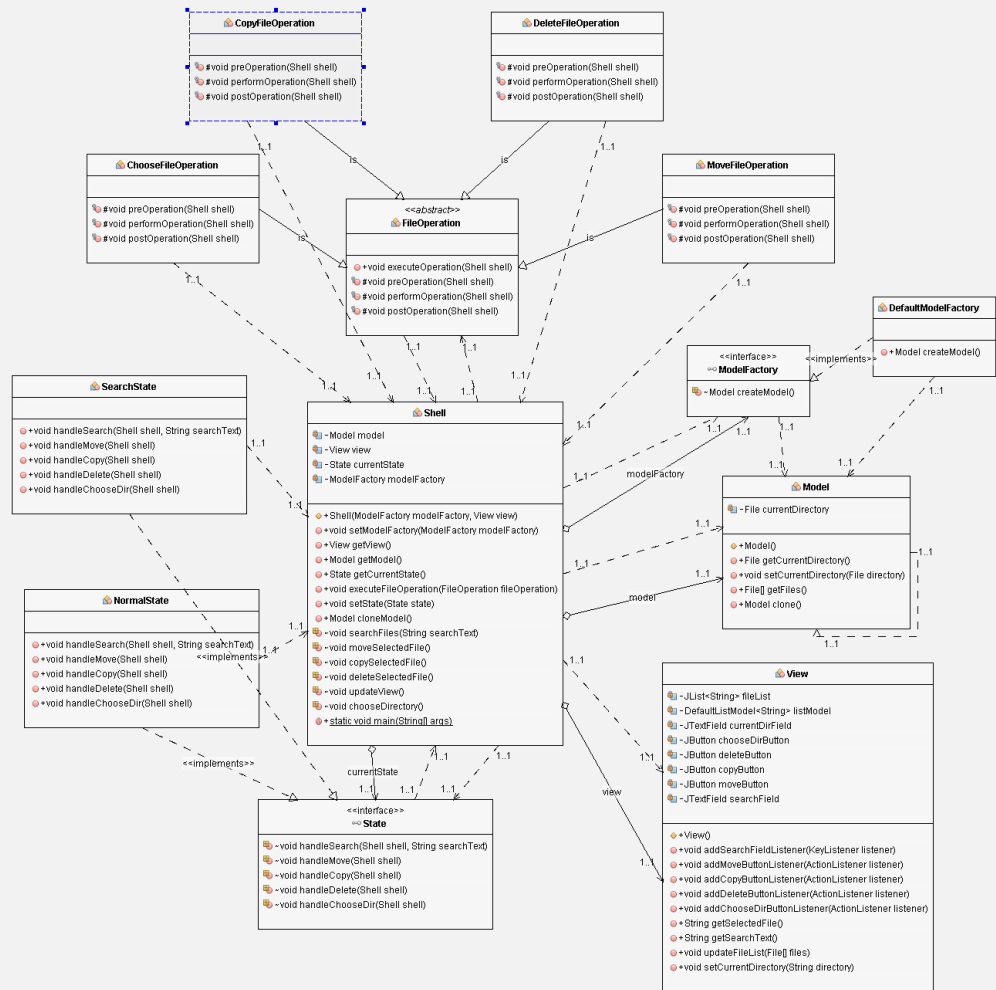


Лабораторна работа #7

Тема: “#18. Shell (total commander)”

Студента групи ІА-14

Розметова Т. Р.



Вихідні коди

```
package com.tyims.shell;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.File;
import javax.swing.SwingUtilities;

public class Shell {

    private Model model;

    private View view;

    private State currentState;

    private ModelFactory modelFactory;

    public Shell(ModelFactory modelFactory, View view) {

        this.modelFactory = modelFactory;

        this.view = view;

        this.model = modelFactory.createModel();

        view.addChooseDirButtonListener(new ChooseDirButtonListener());

        view.addCopyButtonListener(new CopyButtonListener());

        view.addDeleteButtonListener(new DeleteButtonListener());

        view.addMoveButtonListener(new MoveButtonListener());

        view.addSearchFieldListener(new SearchFieldListener());

        updateView();
    }
}
```

```

        currentState = new NormalState();
    }

    public void setModelFactory(ModelFactory modelFactory) {

        this.modelFactory = modelFactory;

        this.model = modelFactory.createModel();
    }

    public View getView() {

        return view;
    }

    public Model getModel() {

        return model;
    }

    public State getCurrentState() {

        return currentState;
    }

    private class SearchFieldListener implements KeyListener {

        @Override

        public void keyTyped(KeyEvent e) {

        }

        @Override

        public void keyPressed(KeyEvent e) {

        }
    }

```

```

@Override

public void keyReleased(KeyEvent e) {

    String searchText = view.getSearchText();

    if (!searchText.isEmpty()) {

        setState(new SearchState());

        //System.out.print("SearchState");

    } else {

        setState(new NormalState());

        //System.out.print("NormalState");

    }

    searchFiles(searchText);

}

}

private class MoveButtonListener implements ActionListener {

    @Override

    public void actionPerformed(ActionEvent e) {

        MoveFileOperation moveFileOperation = new MoveFileOperation();

        executeFileOperation(moveFileOperation);

    }

}

private class CopyButtonListener implements ActionListener {

    @Override

    public void actionPerformed(ActionEvent e) {

        CopyFileOperation copyFileOperation = new CopyFileOperation();

        executeFileOperation(copyFileOperation);

    }

}

```

```
}  
}
```

```
private class DeleteButtonListener implements ActionListener {  
  
    @Override  
  
    public void actionPerformed(ActionEvent e) {  
  
        DeleteFileOperation deleteFileOperation = new DeleteFileOperation();  
  
        executeFileOperation(deleteFileOperation);  
  
    }  
}
```

```
private class ChooseDirButtonListener implements ActionListener {  
  
    @Override  
  
    public void actionPerformed(ActionEvent e) {  
  
        ChooseFileOperation chooseFileOperation = new ChooseFileOperation();  
  
        executeFileOperation(chooseFileOperation);  
  
    }  
}
```

```
public void executeFileOperation(FileOperation fileOperation) {  
  
    fileOperation.executeOperation(this);  
  
}
```

```
public void setState(State state) {  
  
    currentState = state;  
  
}
```

```
public Model cloneModel() {
```

```
        return model.clone();
    }
}
```

```
void searchFiles(String searchText) {
    currentState.handleSearch(this, searchText);
}
```

```
void moveSelectedFile() {
    currentState.handleMove(this);
}
```

```
void copySelectedFile() {
    currentState.handleCopy(this);
}
```

```
void deleteSelectedFile() {
    currentState.handleDelete(this);
}
```

```
void updateView() {
    File[] files = model.GetFiles();
    view.updateFileList(files);
    view.setCurrentDirectory(model.getCurrentDirectory().getAbsolutePath());
}
```

```
void chooseDirectory() {
    currentState.handleChooseDir(this);
}
```

```

public static void main(String[] args) {

    SwingUtilities.invokeLater(new Runnable() {

        @Override

        public void run() {

            ModelFactory model = new DefaultModelFactory();

            View view = new View();

            new Shell(model, view);

        }

    });

}
}

```

```

package com.tyims.shell;

```

```

public abstract class FileOperation {

    public final void executeOperation(Shell shell) {

        preOperation(shell);

        performOperation(shell);

        postOperation(shell);

    }

}

```

```

protected abstract void preOperation(Shell shell);

```

```

protected abstract void performOperation(Shell shell);

```

```

protected abstract void postOperation(Shell shell);

}

```



```
package com.tyims.shell;
```

```
public class ChooseFileOperation extends FileOperation {
```

```
    @Override
```

```
    protected void preOperation(Shell shell) {
```

```
    }
```

```
    @Override
```

```
    protected void performOperation(Shell shell) {
```

```
        shell.chooseDirectory();
```

```
    }
```

```
    @Override
```

```
    protected void postOperation(Shell shell) {
```

```
        shell.updateView();
```

```
    }
```

```
}
```

```
package com.tyims.shell;
```

```
public class MoveFileOperation extends FileOperation {
```

```
    @Override
```

```
    protected void preOperation(Shell shell) {
```

```
    }
```

```
    @Override
```

```
    protected void performOperation(Shell shell) {
```

```
        shell.moveSelectedFile();  
    }  
}
```

```
@Override  
protected void postOperation(Shell shell) {  
    shell.updateView();  
}  
}
```

```
package com.tyims.shell;
```

```
public class CopyFileOperation extends FileOperation {  
    @Override  
    protected void preOperation(Shell shell) {  
    }  
}
```

```
@Override  
protected void performOperation(Shell shell) {  
    shell.copySelectedFile();  
}  
}
```

```
@Override  
protected void postOperation(Shell shell) {  
    shell.updateView();  
}  
}
```

```
package com.tyims.shell;
```

```
public class DeleteFileOperation extends FileOperation {

    @Override

    protected void preOperation(Shell shell) {

    }


    @Override

    protected void performOperation(Shell shell) {

        shell.deleteSelectedFile();

    }


    @Override

    protected void postOperation(Shell shell) {

        shell.updateView();

    }

}
```