A Narendiran
PES1UG19CS001

MVC

Problem statement: To get the possible moves of a chess piece given its current position.

Main

```
public class Main {
   public static void main(String[] args){
      IModel m = new Model();
       IView v = new View(m);
   }
}
```

IModel

```
import java.util.Observer;
public interface IModel {
   void moves();
   void setPos(String curr);
   void currPiece(String curr);
   void addObserver(Observer o);
   String getMoves();
}
```

IView

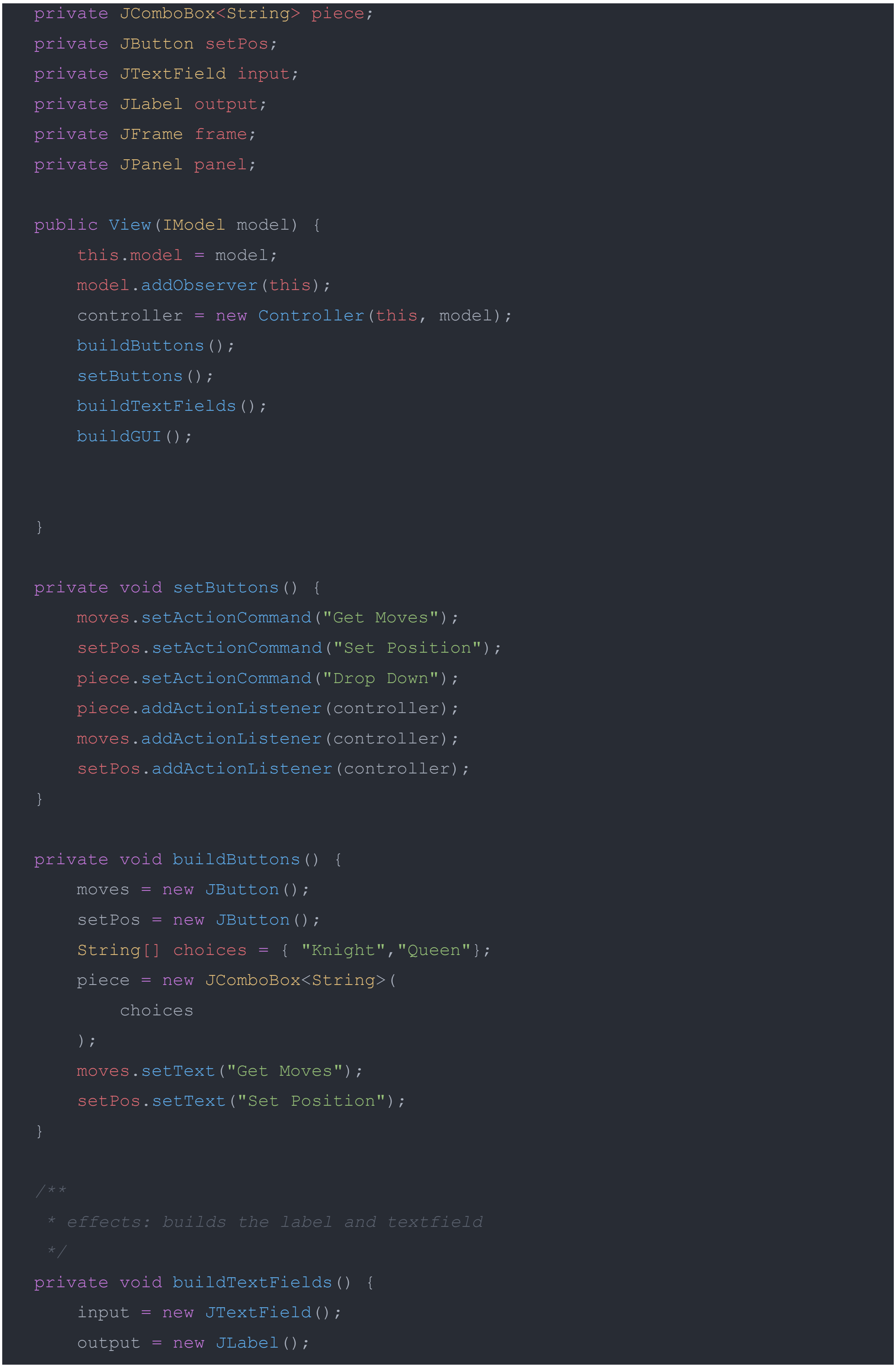
```
public interface IView {
    String getInput();
    String getPiece();
}
```

Model

```
import java.util.Observable;
import java.util.Observer;

public class Model extends Observable implements IModel {
   private String getmoves = "";
   private String curr;
   private String currPiece;
   public void model(){

   }
   @Override
   public void moves() {
       if (currPiece.equals("Queen")){
           this.getmoves = " c1, c2, c3, c4, c6, c7, c8";
       }
       else{
       this.getmoves = " e4, e6, e8, f4 , f8, f6";
   }
       System.out.println("Curr "+curr);
       setChanged();
       notifyObservers();
   }
   @Override
   public String getMoves(){
       return getmoves;
   }
   @Override
   public void currPiece(String curr){
       System.out.println("Curr Piece is "+ curr);
       this.currPiece = curr;
       setChanged();
       notifyObservers();
   }
   @Override
   public void setPos(String curr) {
       System.out.println("pos is" + curr);
       this.curr = curr;
       setChanged();
       notifyObservers();
   }
   @Override
       public void addObserver(Observer o) {
           super.addObserver(o);
       }
}
```

View

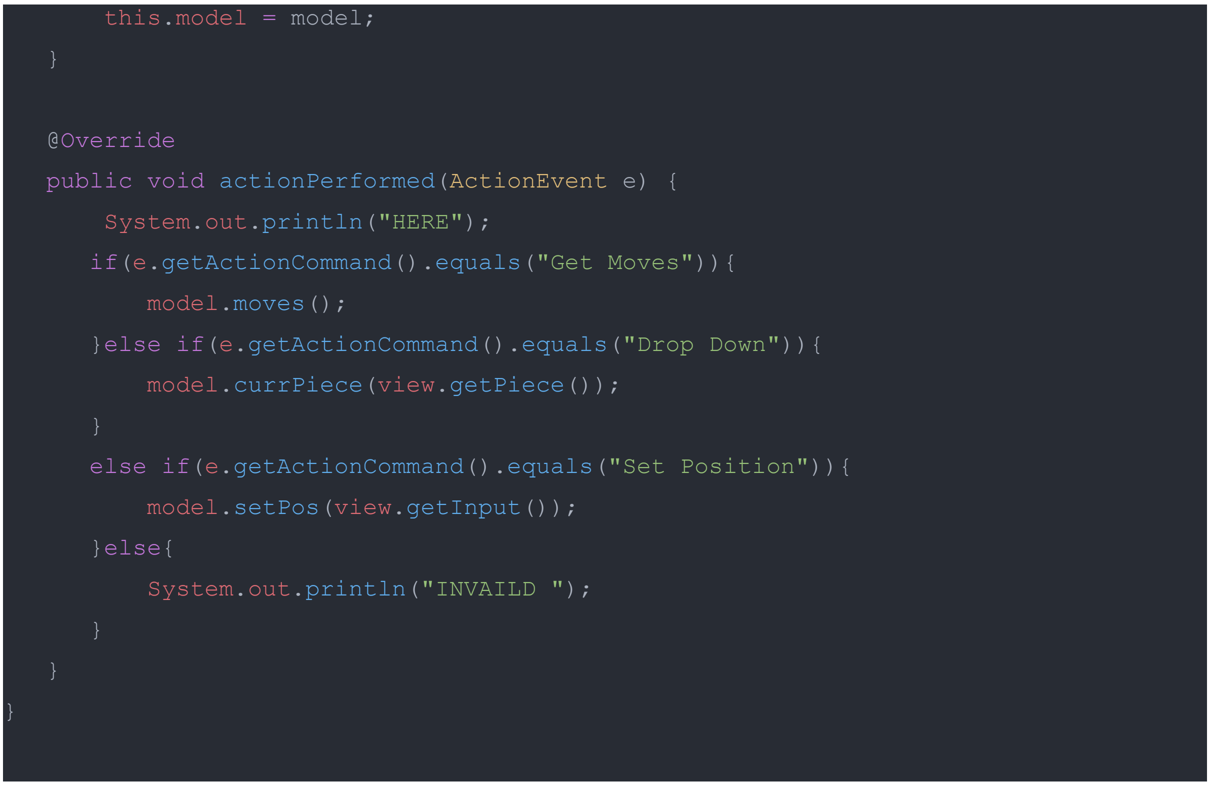
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
import java.util.Observable;
import java.util.Observer;

public class View implements IView, Observer {
    private IModel model;
    private ActionListener controller;

    private JButton moves;
```



```
private JComboBox<String> piece;
private JButton setPos;
private JTextField input;
private JLabel output;
private JFrame frame;
private JPanel panel;

public View(IModel model) {
    this.model = model;
    model.addObserver(this);
    controller = new Controller(this, model);
    buildButtons();
    setButtons();
    buildTextFields();
    buildGUI();



}

private void setButtons() {
    moves.setActionCommand("Get Moves");
    setPos.setActionCommand("Set Position");
    piece.setActionCommand("Drop Down");
    piece.addActionListener(controller);
    moves.addActionListener(controller);
    setPos.addActionListener(controller);
}

private void buildButtons() {
    moves = new JButton();
    setPos = new JButton();
    String[] choices = { "Knight","Queen"};
    piece = new JComboBox<String>(
        choices
    );
    moves.setText("Get Moves");
    setPos.setText("Set Position");
}

/**
 * effects: builds the label and textfield
 */
private void buildTextFields() {
    input = new JTextField();
    output = new JLabel();
    }

    private void buildGUI() {
        frame = new JFrame();
        panel = new JPanel();
        panel.add(piece);
        panel.add(setPos);
        panel.add(input);
        panel.add(moves);
        panel.add(output);
        panel.setLayout(new GridLayout(5, 1));
        frame.add(panel);
        frame.pack();
        update(null, null);
        frame.setVisible(true);
    }
    @Override
    public String getPiece() {
        System.out.println(piece.getItemAt((piece.getSelectedIndex())));
        return "" + piece.getItemAt((piece.getSelectedIndex()));
    }

    @Override
    public String getInput() {
        return input.getText();
    }

    @Override
    public void update(Observable o, Object arg) {
        output.setText(model.getMoves());
        System.out.println("UPDATE");
    }

}
```

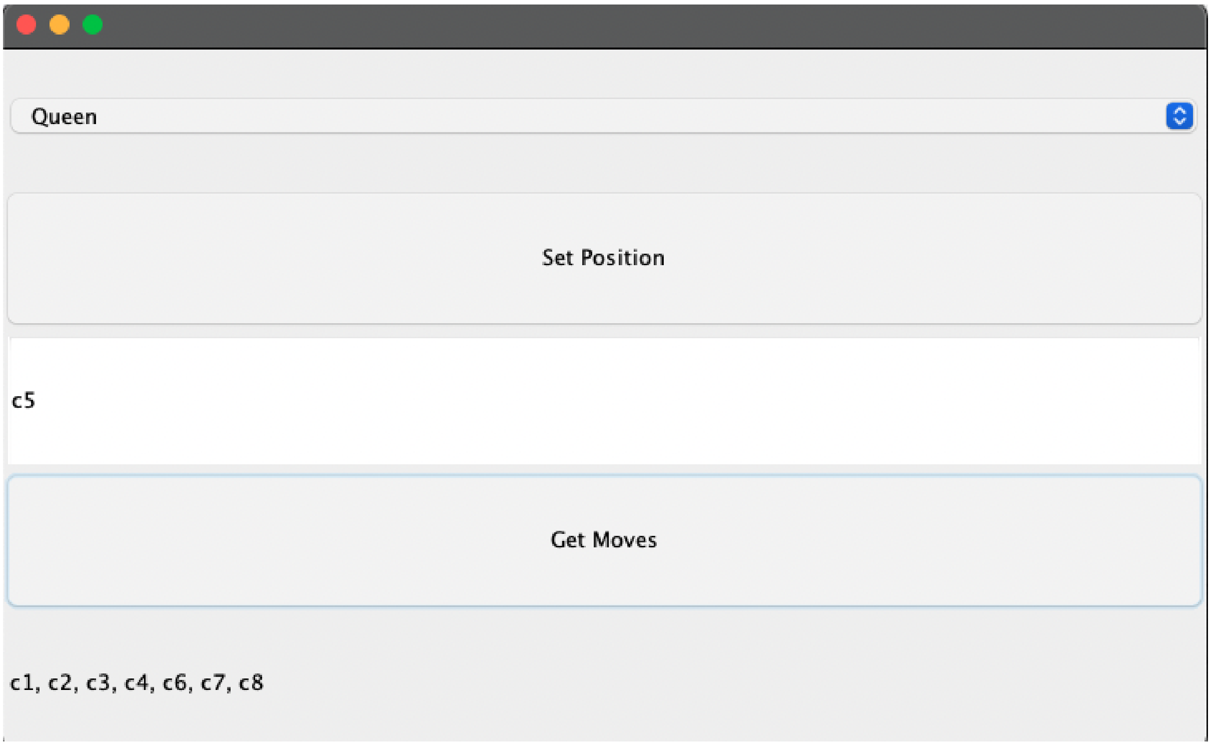
Controller

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

public class Controller implements ActionListener {
    private IView view;
    private IModel model;

    public Controller(IView view, IModel model){
        this.view = view;
```



```
        this.model = model;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
         System.out.println("HERE");
        if(e.getActionCommand().equals("Get Moves")){
            model.moves();
        }else if(e.getActionCommand().equals("Drop Down")){
            model.currPiece(view.getPiece());
        }
        else if(e.getActionCommand().equals("Set Position")){
            model.setPos(view.getInput());
        }else{
            System.out.println("INVAILD ");
        }
    }
}
```

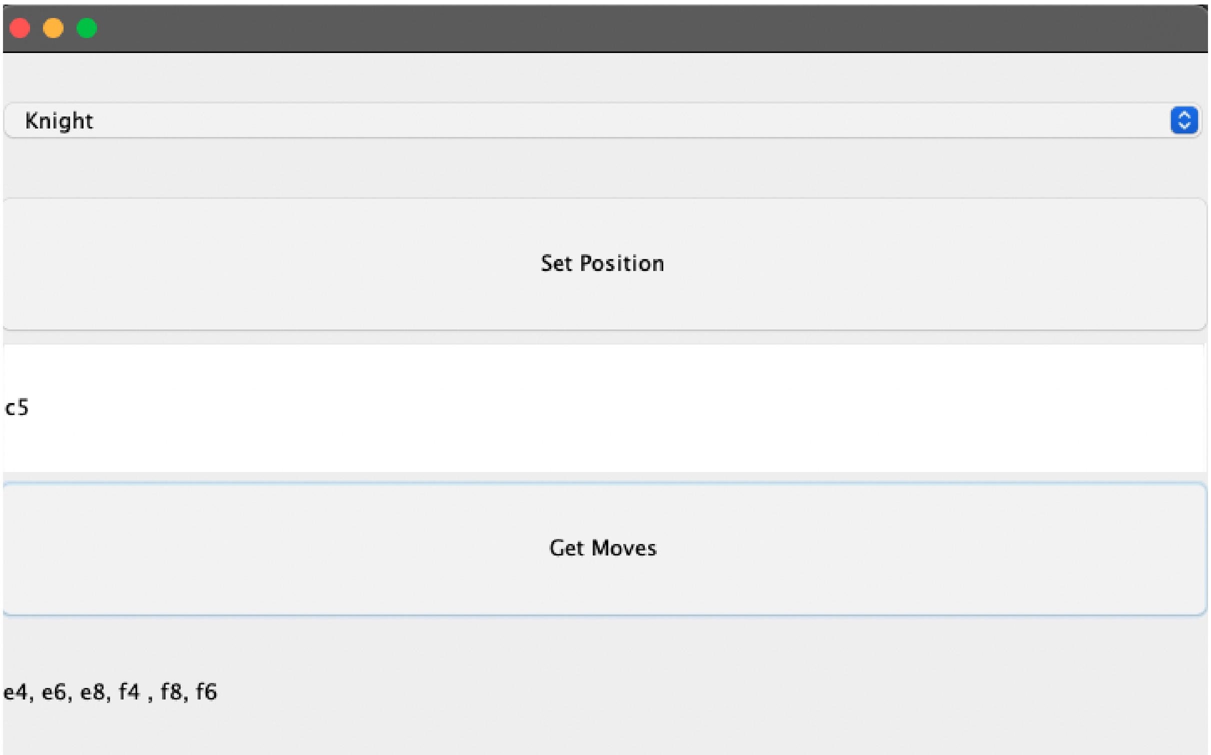
Output



Queen

c5

c1, c2, c3, c4, c6, c7, c8



Knight

c5

```
e4, e6, e8, f4 , f8, f6
```