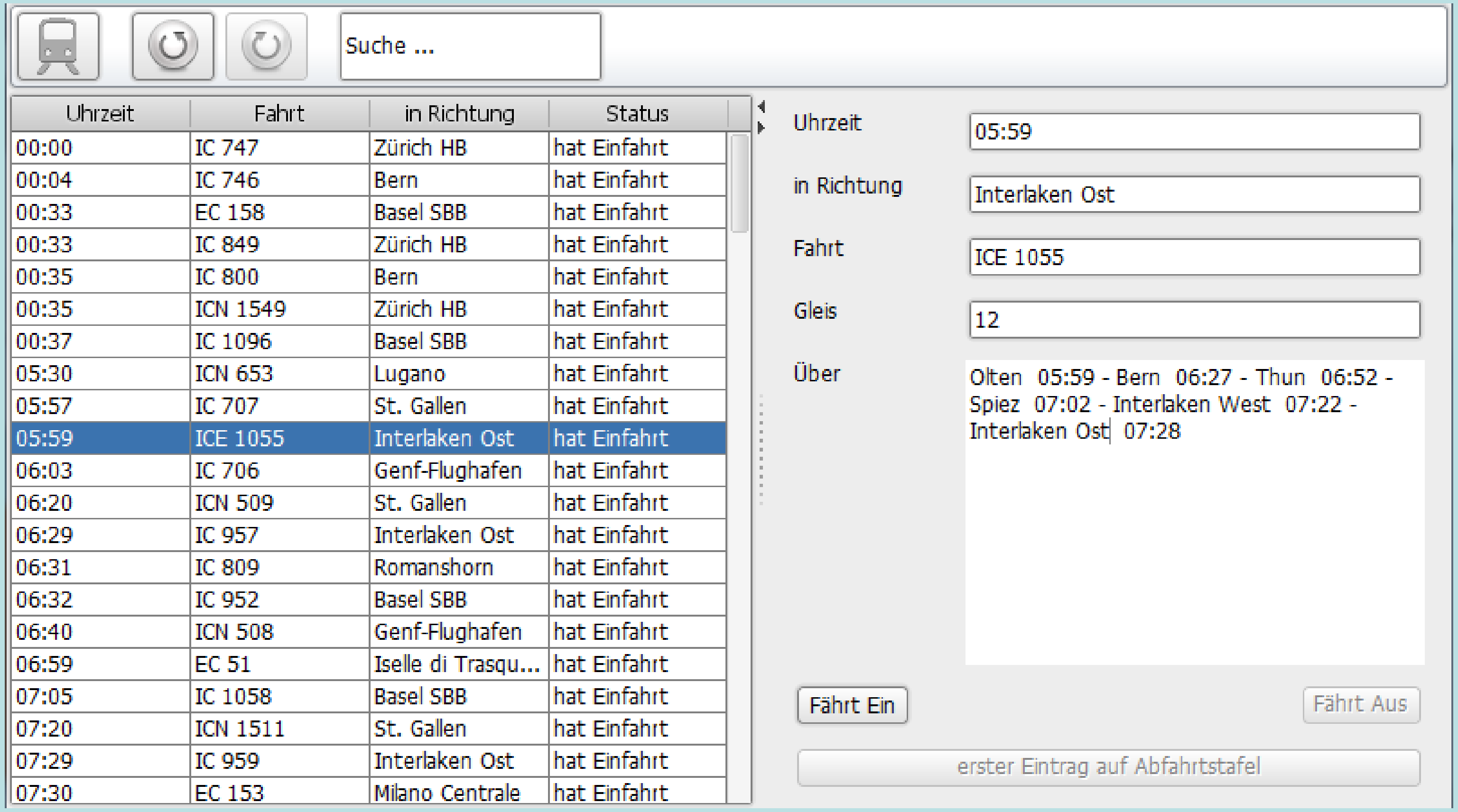
Fachhochschule Nordwestschweiz
Hochschule für Technik

DepartureBoard

Yves Lauber, Faizan Mohammad



Uhrzeit	Fahrt	in Richtung	Status
00:00	IC 747	Zürich HB	hat Einfahrt
00:04	IC 746	Bern	hat Einfahrt
00:33	EC 158	Basel SBB	hat Einfahrt
00:33	IC 849	Zürich HB	hat Einfahrt
00:35	IC 800	Bern	hat Einfahrt
00:35	ICN 1549	Zürich HB	hat Einfahrt
00:37	IC 1096	Basel SBB	hat Einfahrt
05:30	ICN 653	Lugano	hat Einfahrt
05:57	IC 707	St. Gallen	hat Einfahrt
05:59	ICE 1055	Interlaken Ost	hat Einfahrt
06:03	IC 706	Genf-Flughafen	hat Einfahrt
06:20	ICN 509	St. Gallen	hat Einfahrt
06:29	IC 957	Interlaken Ost	hat Einfahrt
06:31	IC 809	Romanshorn	hat Einfahrt
06:32	IC 952	Basel SBB	hat Einfahrt
06:40	ICN 508	Genf-Flughafen	hat Einfahrt
06:59	EC 51	Iselle di Trasqu...	hat Einfahrt
07:05	IC 1058	Basel SBB	hat Einfahrt
07:20	ICN 1511	St. Gallen	hat Einfahrt
07:29	IC 959	Interlaken Ost	hat Einfahrt
07:30	EC 153	Milano Centrale	hat Einfahrt

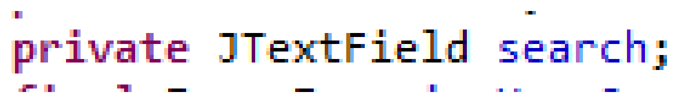
Uhrzeit	05:59
in Richtung	Interlaken Ost
Fahrt	ICE 1055
Gleis	12
Über	Olten 05:59 - Bern 06:27 - Thun 06:52 - Spiez 07:02 - Interlaken West 07:22 - Interlaken Ost 07:28

Implementiert Features

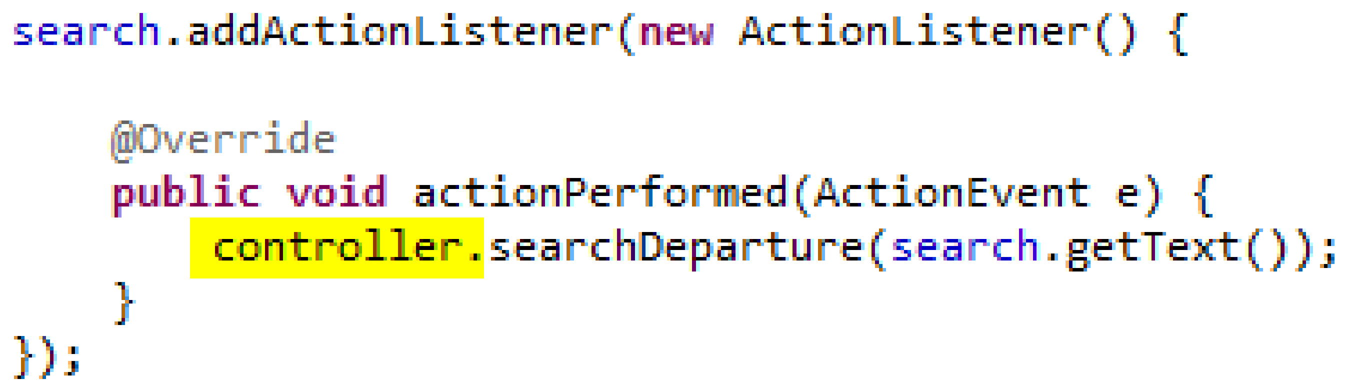
- Basis Features MVC-Pattern & Observer Pattern
- Kleine Zusatz-Features - **Freitextsuche**
- **Undo/Redo**

Demo

Freitextsuche – Implementierung - View

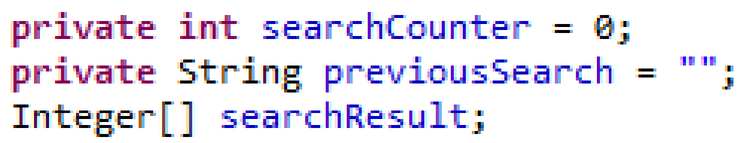


```
private JTextField search;
```

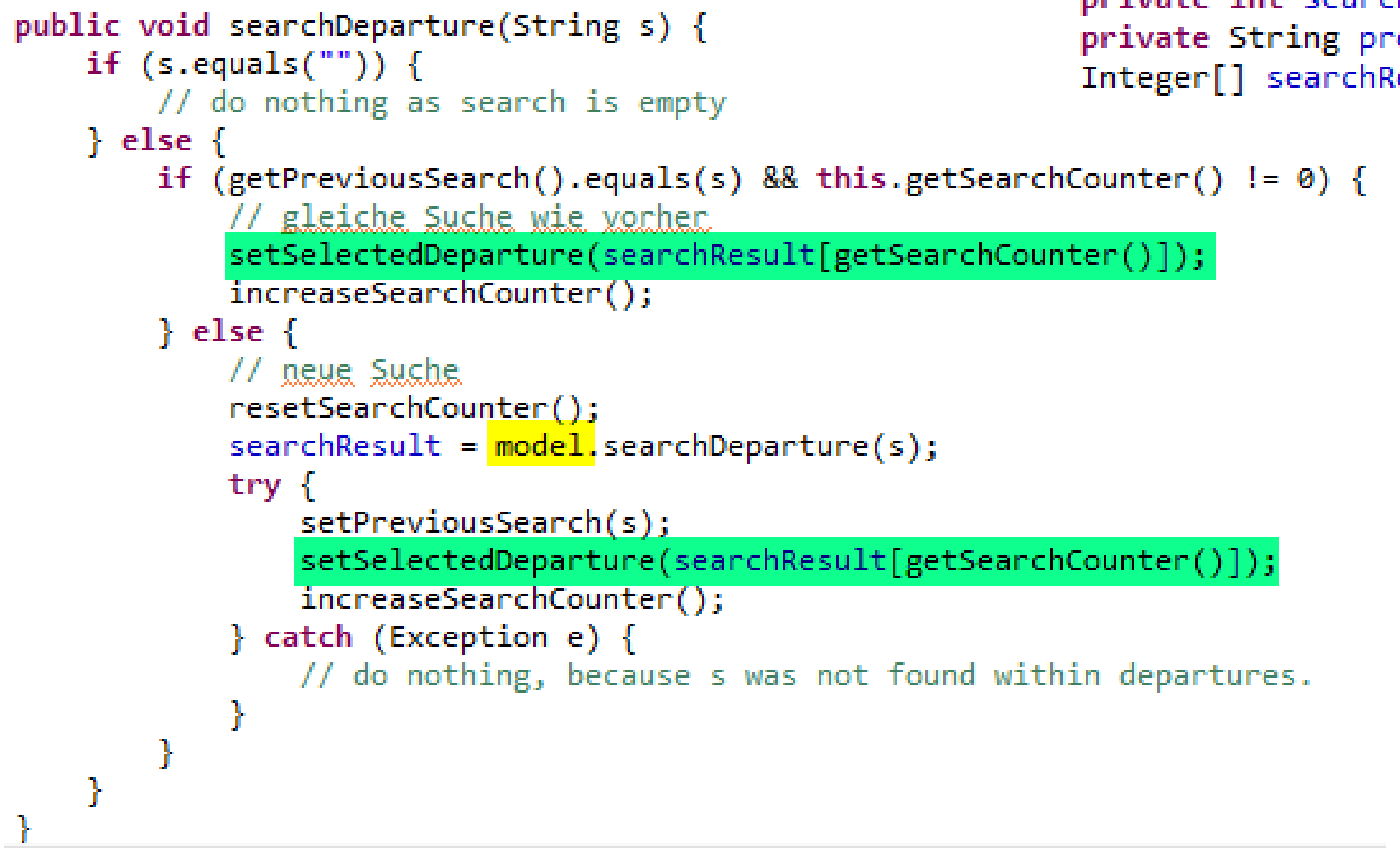


```
search.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        controller.searchDeparture(search.getText());
    }
});
```

Freitextsuche – Implementierung - Controller

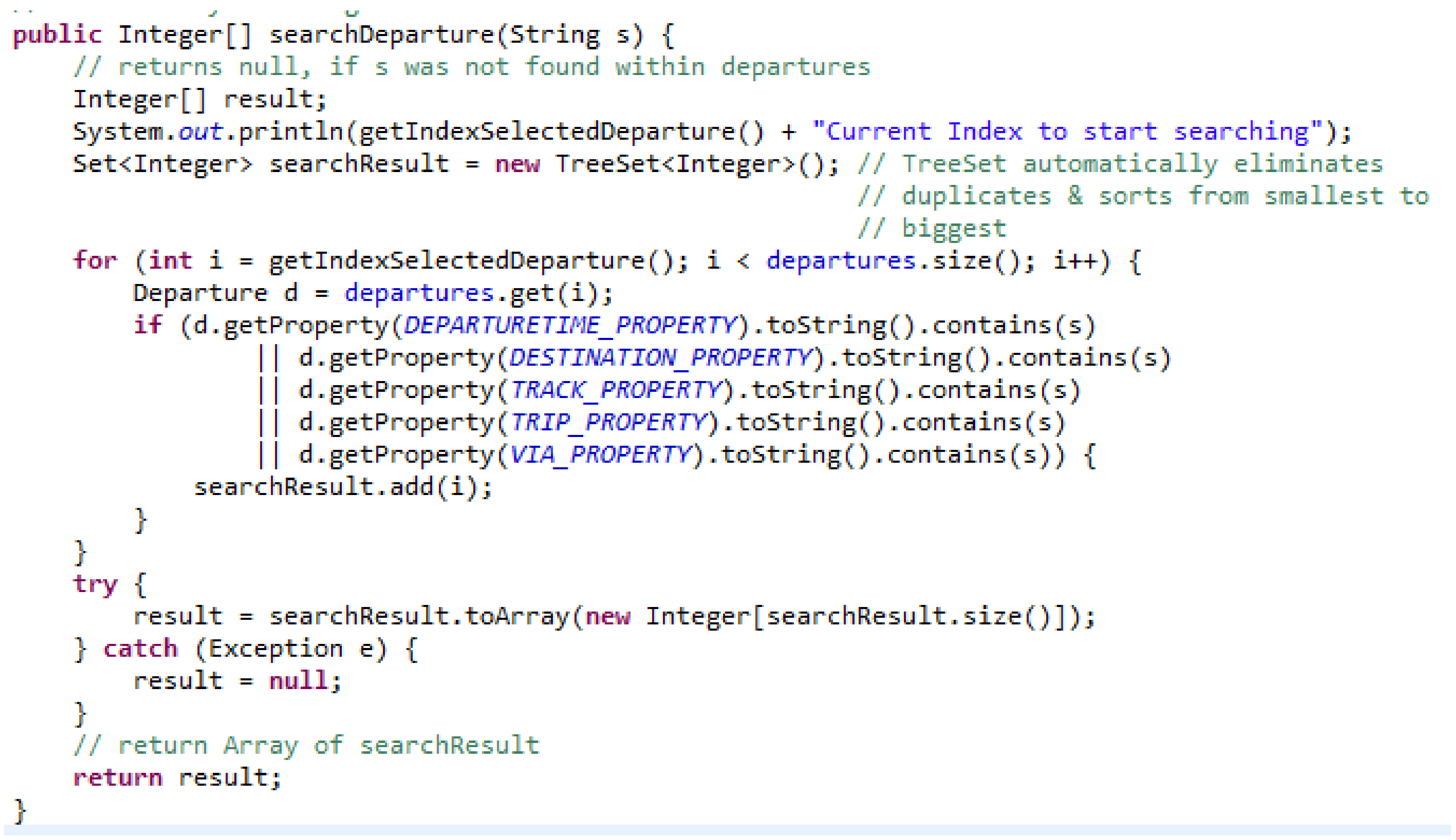


```java
private int searchCounter = 0;
private String previousSearch = "";
Integer[] searchResult;
```



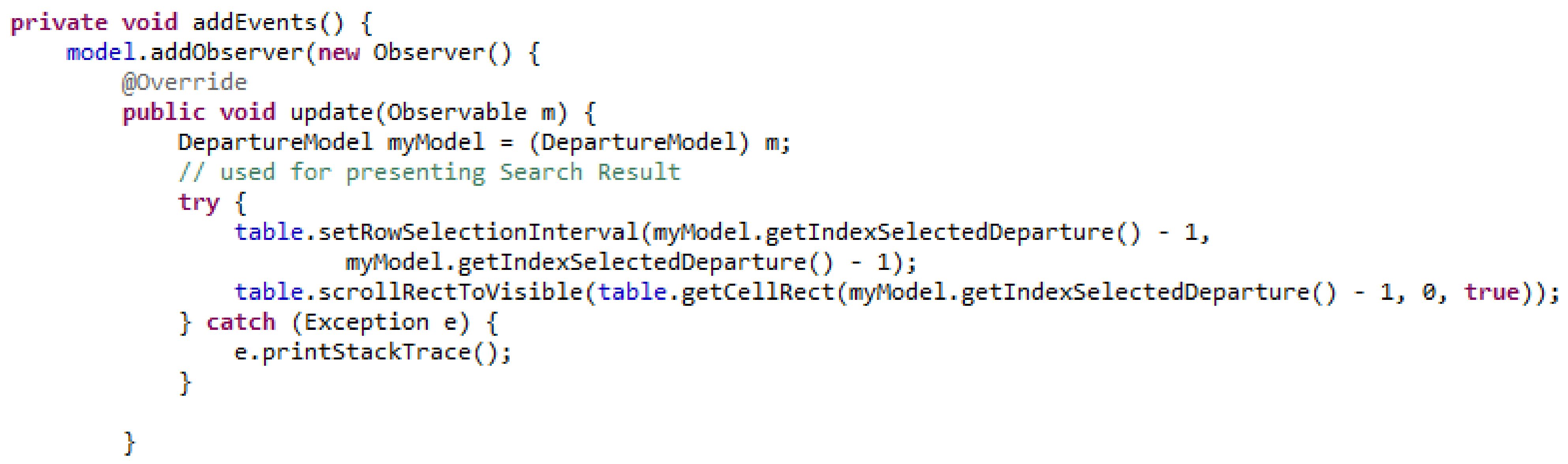
```java
public void searchDeparture(String s) {
    if (s.equals("")) {
        // do nothing as search is empty
    } else {
        if (getPreviousSearch().equals(s) && this.getSearchCounter() != 0) {
            // gleiche Suche wie vorher
            setSelectedDeparture(searchResult[getSearchCounter()]);
            increaseSearchCounter();
        } else {
            // neue Suche
            resetSearchCounter();
            searchResult = model.searchDeparture(s);
            try {
                setPreviousSearch(s);
                setSelectedDeparture(searchResult[getSearchCounter()]);
                increaseSearchCounter();
            } catch (Exception e) {
                // do nothing, because s was not found within departures.
            }
        }
    }
}
```

Freitextsuche – Implementierung - Model



```java
public Integer[] searchDeparture(String s) {
    // returns null, if s was not found within departures
    Integer[] result;
    System.out.println(getIndexSelectedDeparture() + "Current Index to start searching");
    Set<Integer> searchResult = new TreeSet<Integer>(); // TreeSet automatically eliminates
                                                         // duplicates & sorts from smallest to
                                                         // biggest
    for (int i = getIndexSelectedDeparture(); i < departures.size(); i++) {
        Departure d = departures.get(i);
        if (d.getProperty(DEPARTURETIME_PROPERTY).toString().contains(s)
                || d.getProperty(DESTINATION_PROPERTY).toString().contains(s)
                || d.getProperty(TRACK_PROPERTY).toString().contains(s)
                || d.getProperty(TRIP_PROPERTY).toString().contains(s)
                || d.getProperty(VIA_PROPERTY).toString().contains(s)) {
            searchResult.add(i);
        }
    }
    try {
        result = searchResult.toArray(new Integer[searchResult.size()]);
    } catch (Exception e) {
        result = null;
    }
    // return Array of searchResult
    return result;
}
```

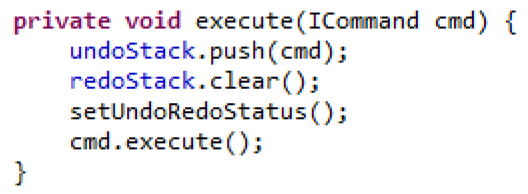
Freitextsuche – Implementierung – update Table View



```java
private void addEvents() {
    model.addObserver(new Observer() {
        @Override
        public void update(Observable m) {
            DepartureModel myModel = (DepartureModel) m;
            // used for presenting Search Result
            try {
                table.setRowSelectionInterval(myModel.getIndexSelectedDeparture() - 1,
                        myModel.getIndexSelectedDeparture() - 1);
                table.scrollRectToVisible(table.getCellRect(myModel.getIndexSelectedDeparture() - 1, 0, true));
            } catch (Exception e) {
                e.printStackTrace();
            }

        }
```

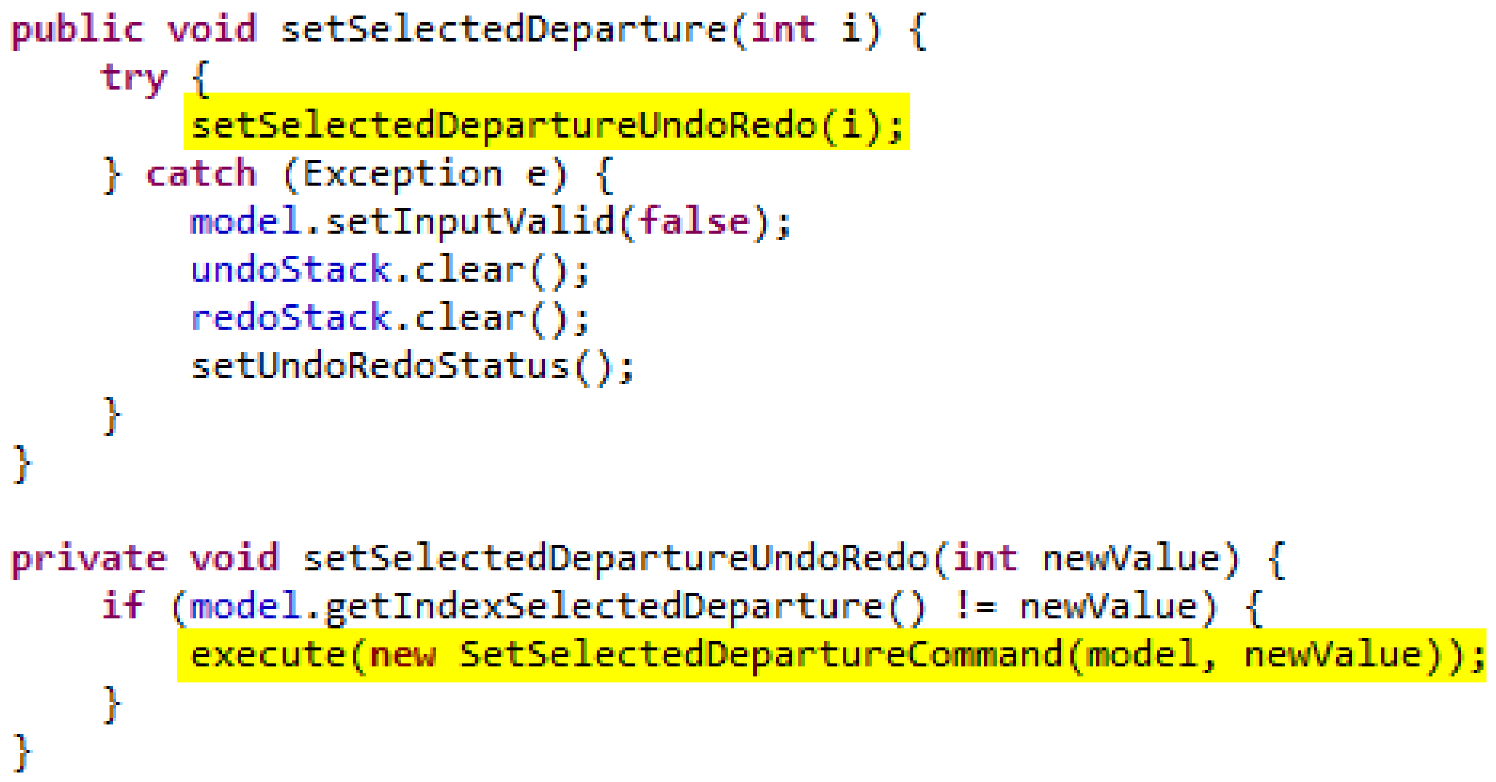
Undo/Redo

Demo



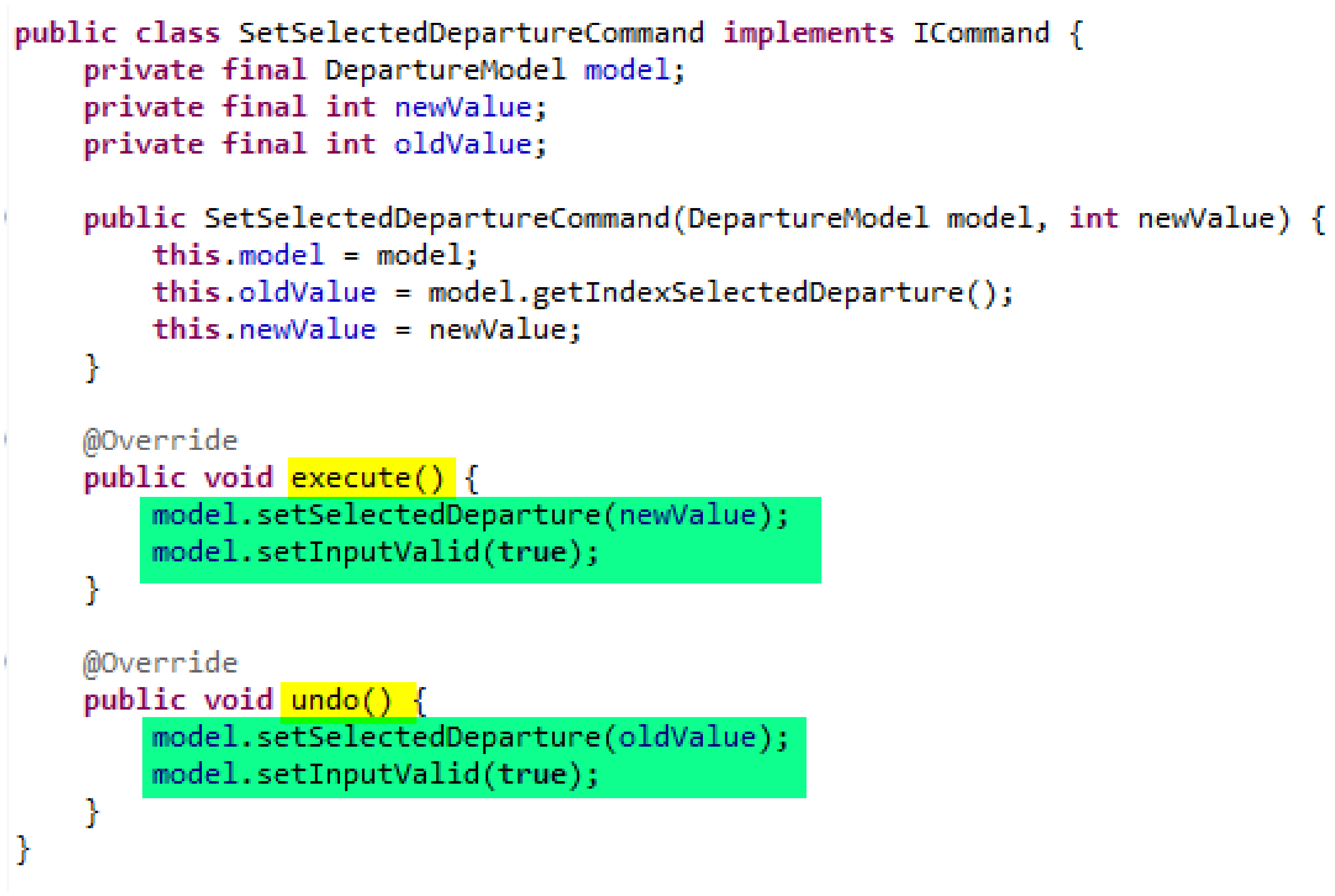
```
private void execute(ICommand cmd) {
    undoStack.push(cmd);
    redoStack.clear();
    setUndoRedoStatus();
    cmd.execute();
}
```

Undo/Redo– Implementierung – Controller



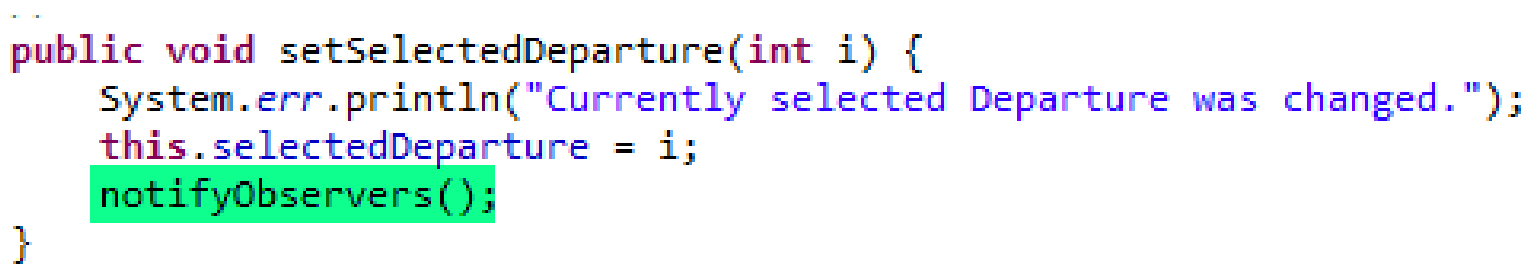
```
public void setSelectedDeparture(int i) {
    try {
        setSelectedDepartureUndoRedo(i);
    } catch (Exception e) {
        model.setInputValid(false);
        undoStack.clear();
        redoStack.clear();
        setUndoRedoStatus();
    }
}

private void setSelectedDepartureUndoRedo(int newValue) {
    if (model.getIndexSelectedDeparture() != newValue) {
        execute(new SetSelectedDepartureCommand(model, newValue));
    }
}
```

Undo/Redo– Implementierung – Command



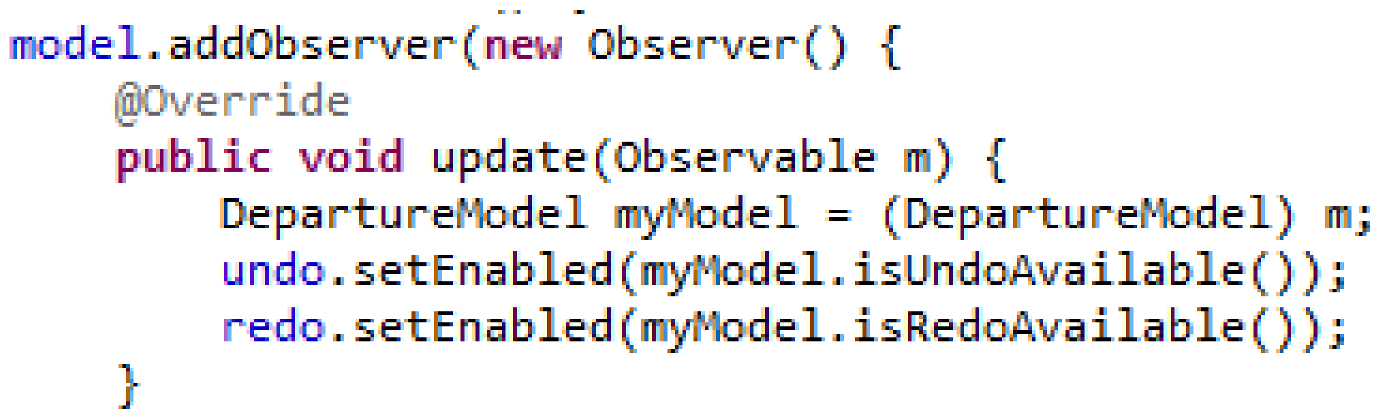
```java
public class SetSelectedDepartureCommand implements ICommand {
    private final DepartureModel model;
    private final int newValue;
    private final int oldValue;

    public SetSelectedDepartureCommand(DepartureModel model, int newValue) {
        this.model = model;
        this.oldValue = model.getIndexSelectedDeparture();
        this.newValue = newValue;
    }

    @Override
    public void execute() {
        model.setSelectedDeparture(newValue);
        model.setInputValid(true);
    }

    @Override
    public void undo() {
        model.setSelectedDeparture(oldValue);
        model.setInputValid(true);
    }
}
```

Undo/Redo– Implementierung – Model



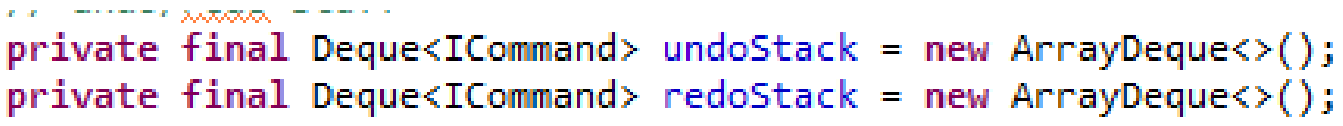
```
..
public void setSelectedDeparture(int i) {
    System.err.println("Currently selected Departure was changed.");
    this.selectedDeparture = i;
    notifyObservers();
}
```

Undo/Redo– Implementierung – ToolbarView

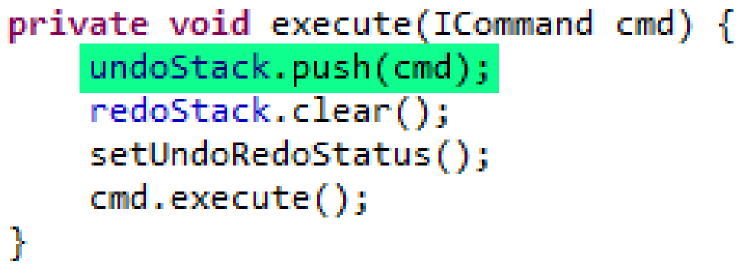


```java
model.addObserver(new Observer() {
    @Override
    public void update(Observable m) {
        DepartureModel myModel = (DepartureModel) m;
        undo.setEnabled(myModel.isUndoAvailable());
        redo.setEnabled(myModel.isRedoAvailable());
    }
```

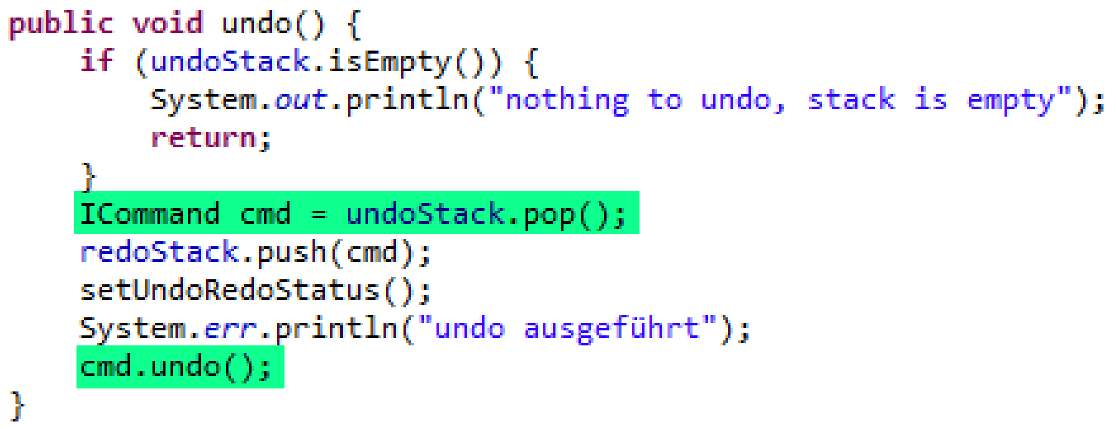
Undo/Redo – Implementierung - ButtonActions



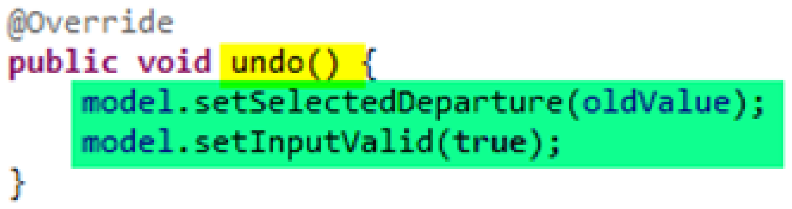
```
private final Deque<ICommand> undoStack = new ArrayDeque<>();
private final Deque<ICommand> redoStack = new ArrayDeque<>();
```



```
private void execute(ICommand cmd) {
    undoStack.push(cmd);
    redoStack.clear();
    setUndoRedoStatus();
    cmd.execute();
}
```



```
public void undo() {
    if (undoStack.isEmpty()) {
        System.out.println("nothing to undo, stack is empty");
        return;
    }
    ICommand cmd = undoStack.pop();
    redoStack.push(cmd);
    setUndoRedoStatus();
    System.err.println("undo ausgeführt");
    cmd.undo();
}
```



```
@Override
public void undo() {
    model.setSelectedDeparture(oldValue);
    model.setInputValid(true);
}
```