Wicket - FORMs

DBMS 2013 Fall

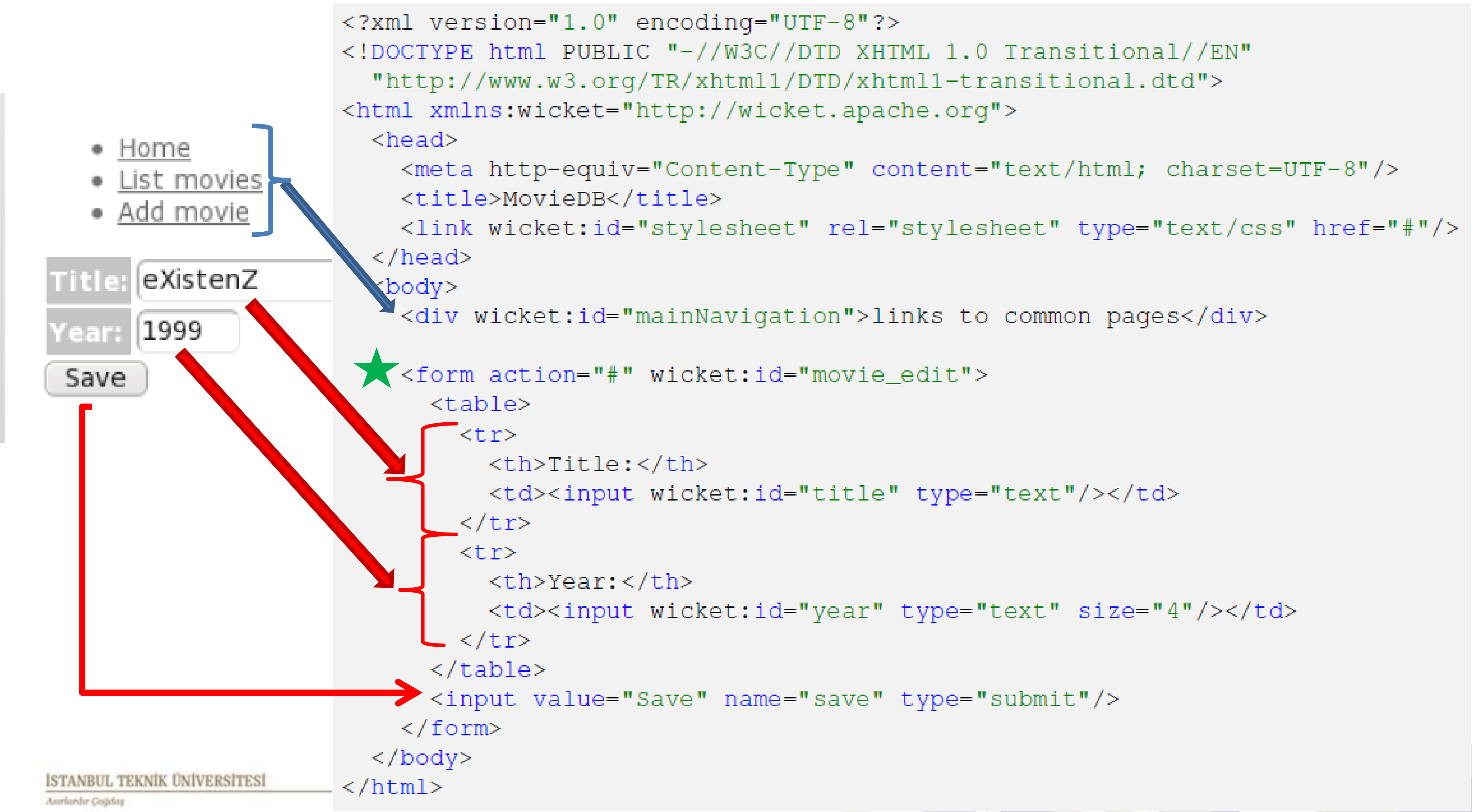
TAs:Mahiye Uluyağmur Öztürk , Nagehan İlhan

1: Form & Text Boxes

- We want to create a page that we can edit movies.
- We'll add a line to header to give a Link to this page
- Changes to this page should update the main application
  - We should see the updated movie list in the movie list page
- Page design includes some text box elements

1: Form & Text Boxes

- New Wicket Page! Html Source:



- Home
- List movies
- Add movie

Title:	eXistenZ
Year:	1999

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns:wicket="http://wicket.apache.org">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>MovieDB</title>
    <link wicket:id="stylesheet" rel="stylesheet" type="text/css" href="#"/>
  </head>
  <body>
    <div wicket:id="mainNavigation">links to common pages</div>

    <form action="#" wicket:id="movie_edit">
      <table>
        <tr>
          <th>Title:</th>
          <td><input wicket:id="title" type="text"/></td>
        </tr>
        <tr>
          <th>Year:</th>
          <td><input wicket:id="year" type="text" size="4"/></td>
        </tr>
      </table>
      <input value="Save" name="save" type="submit"/>
    </form>
  </body>
</html>
```

1: Form & Text Boxes

★ Form Object? Java Class (*MovieEditForm*):

- is derived from Wicket Form class

import org.apache.wicket.markup.html.form.Form;

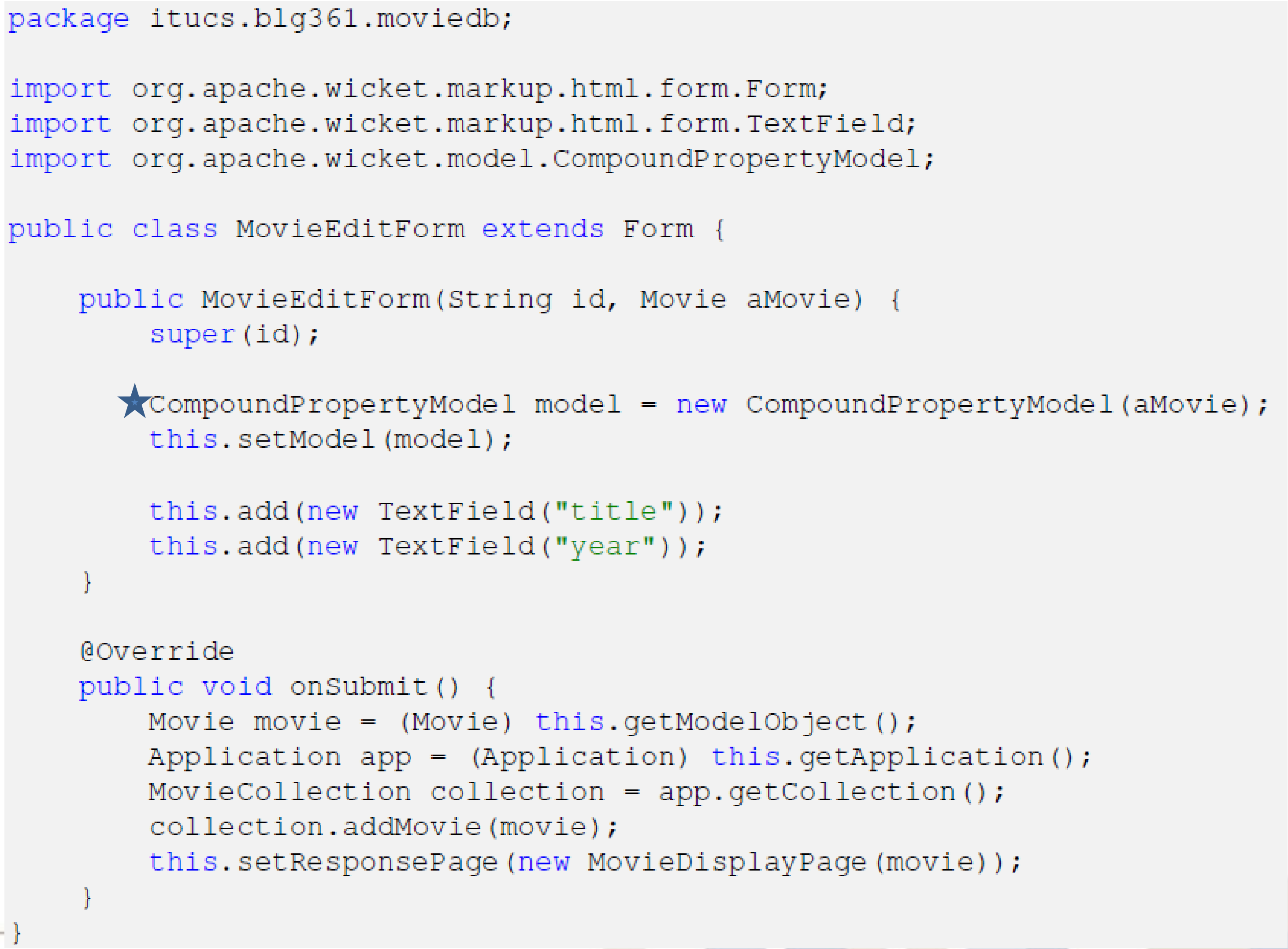
- A new java class
- has to provide components included in the Html file
- «Title» and «Year» fields (Text Field)

import org.apache.wicket.markup.html.form.TextField;

- «onSubmit()», should be overridden in a way that it should add new movie into the collection

1: Form & Text Boxes

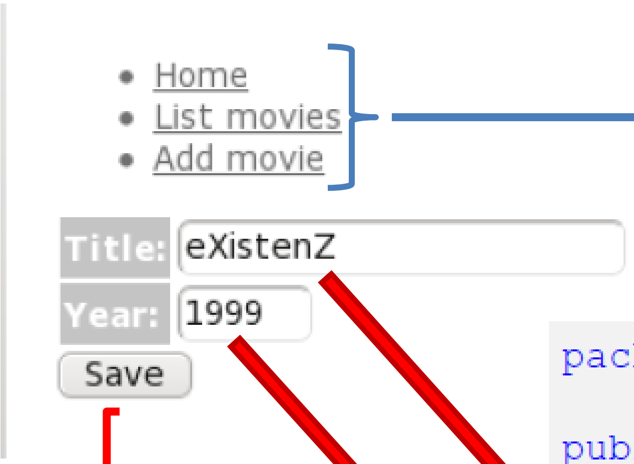
★ Form Object? Java Class (*MovieEditForm*):



```
package itucs.blg361.moviedb;

import org.apache.wicket.markup.html.form.Form;
import org.apache.wicket.markup.html.form.TextField;
import org.apache.wicket.model.CompoundPropertyModel;

public class MovieEditForm extends Form {

    public MovieEditForm(String id, Movie aMovie) {
        super(id);

       ★CompoundPropertyModel model = new CompoundPropertyModel(aMovie);
        this.setModel(model);

        this.add(new TextField("title"));
        this.add(new TextField("year"));
    }

    @Override
    public void onSubmit() {
        Movie movie = (Movie) this.getModelObject();
        Application app = (Application) this.getApplication();
        MovieCollection collection = app.getCollection();
        collection.addMovie(movie);
        this.setResponsePage(new MovieDisplayPage(movie));
    }
}
```

1: Form & Text Boxes

- New Java File!



- Home
- List movies
- Add movie

Title: eXistenZ

Year: 1999

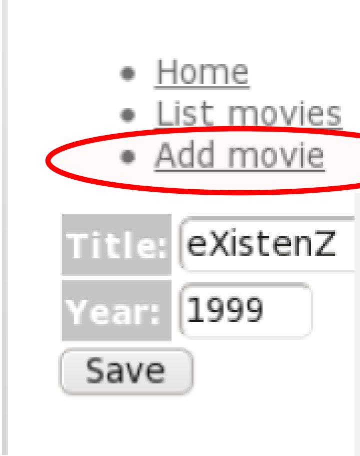
Save

```
package itucs.blg361.moviedb;

public final class MovieEditPage extends BasePage {

    public MovieEditPage(Movie aMovie) {
        this.add(new MovieEditForm("movie_edit", aMovie));
    }
}
```

Just one wicket component: Form (label is movie_edit)

1: Form & Text Boxes

Add link to *MovieEditPage* to the «Header



- Home
- List movies
- Add movie

Title: eXistenZ

Year: 1999

«HeaderPanel» should be modified:

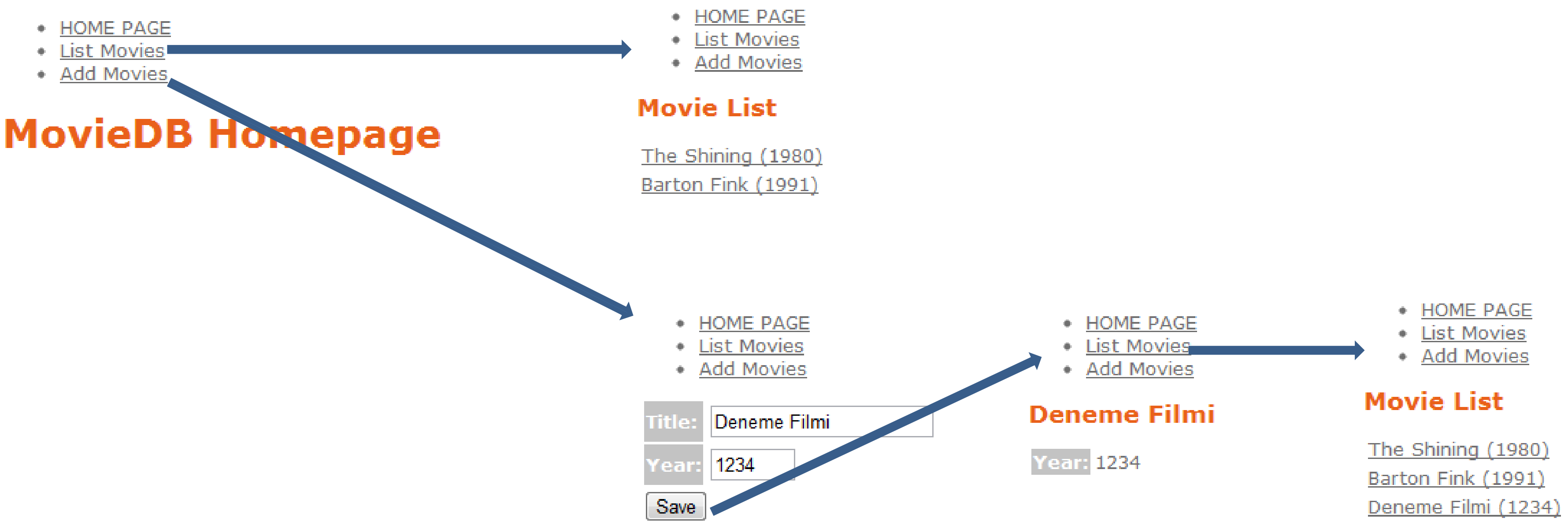
```
<body>
  <wicket:panel>
    <ul>
      <li><a href="#" wicket:id="home">Home</a></li>
      <li><a href="#" wicket:id="list_movies">List movies</a></li>
      <li><a href="#" wicket:id="add_movie">Add movie</a></li>
    </ul>
  </wicket:panel>
</body>
```

```
Link movieAddLink = new Link("add_movie") {

    @Override
    public void onClick() {
        Movie movie = new Movie("");
        this.setResponsePage(new MovieEditPage(movie));
    }
};
this.add(movieAddLink);
```

adding Link to HeaderPanel class constructor

1: Form & Text Boxes

Sample RUN:



- HOME PAGE
- List Movies
- Add Movies

MovieDB Homepage

- HOME PAGE
- List Movies
- Add Movies

Movie List

The Shining (1980)

Barton Fink (1991)

- HOME PAGE
- List Movies
- Add Movies

Title:	Deneme Filmi
Year:	1234

- HOME PAGE
- List Movies
- Add Movies

Deneme Filmi

Year: 1234

- HOME PAGE
- List Movies
- Add Movies

Movie List

The Shining (1980)

Barton Fink (1991)

Deneme Filmi (1234)

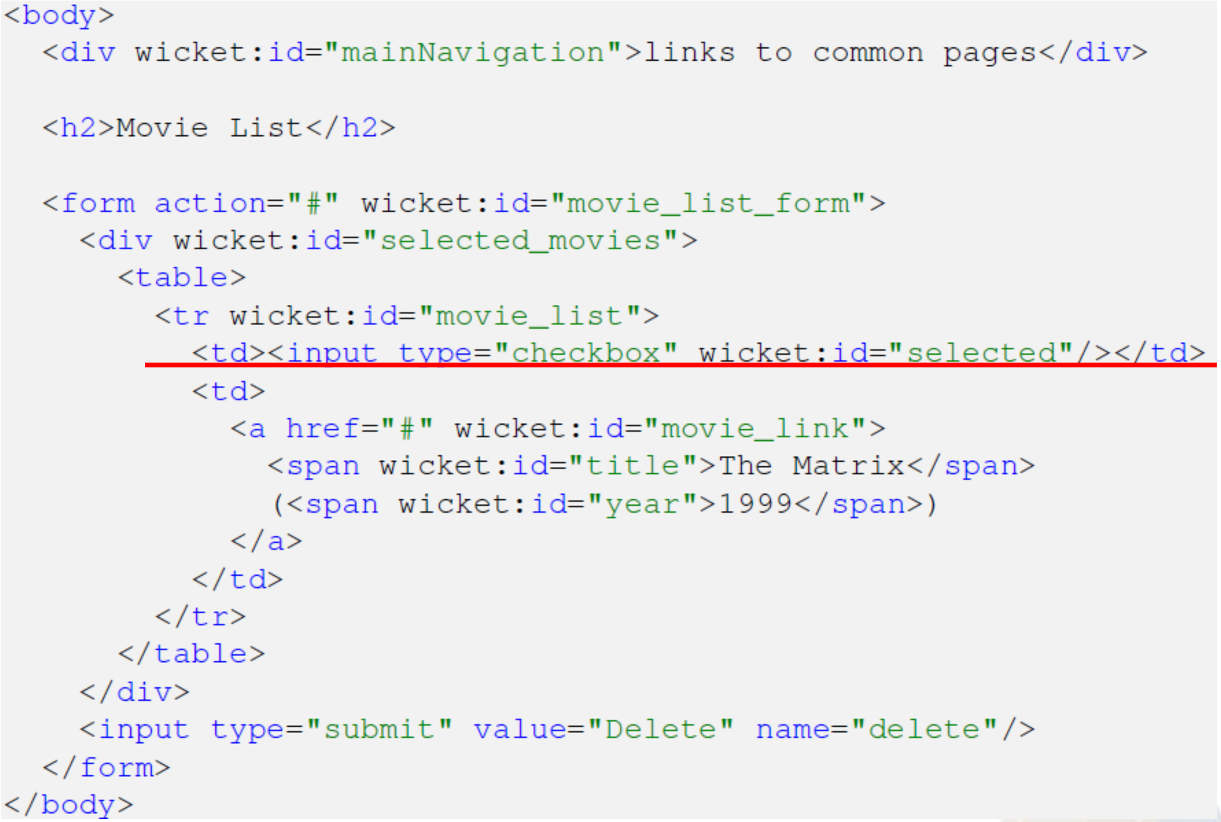
2: Check Boxes

- We want to delete any movie

1. we need to put check boxes next to every movie
2. we need to place a delete button to the bottom of the page
3. we need to delete selected movies when the button is clicked

2: Check Boxes

1: «MovieListPage.html» adding checkbox :



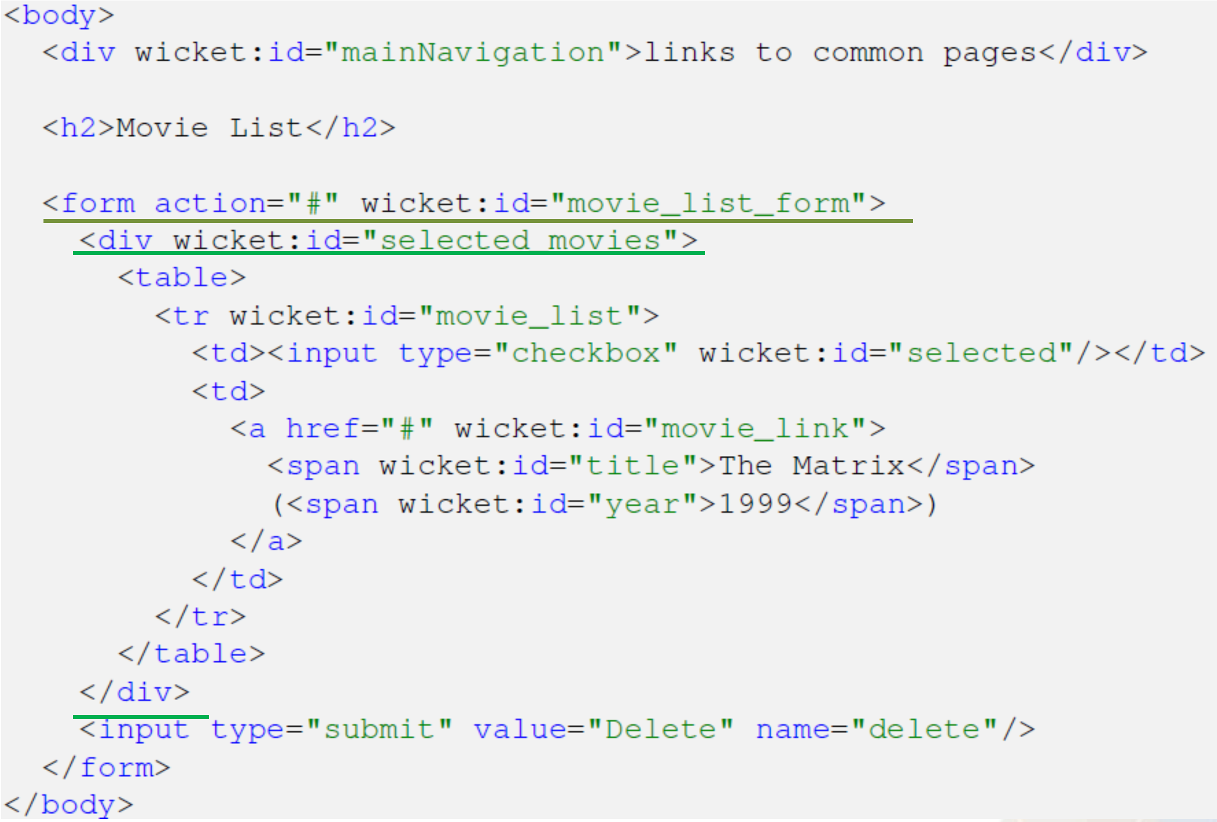
```
<body>
  <div wicket:id="mainNavigation">links to common pages</div>

  <h2>Movie List</h2>

  <form action="#" wicket:id="movie_list_form">
    <div wicket:id="selected_movies">
      <table>
        <tr wicket:id="movie_list">
          <td><input type="checkbox" wicket:id="selected"/></td>
          <td>
            <a href="#" wicket:id="movie_link">
              <span wicket:id="title">The Matrix</span>
              (<span wicket:id="year">1999</span>)
            </a>
          </td>
        </tr>
      </table>
    </div>
    <input type="submit" value="Delete" name="delete"/>
  </form>
</body>
```

2: Check Boxes

1: «MovieListPage.html» adding checkbox :

to group selected checkboxes use div wicket component!

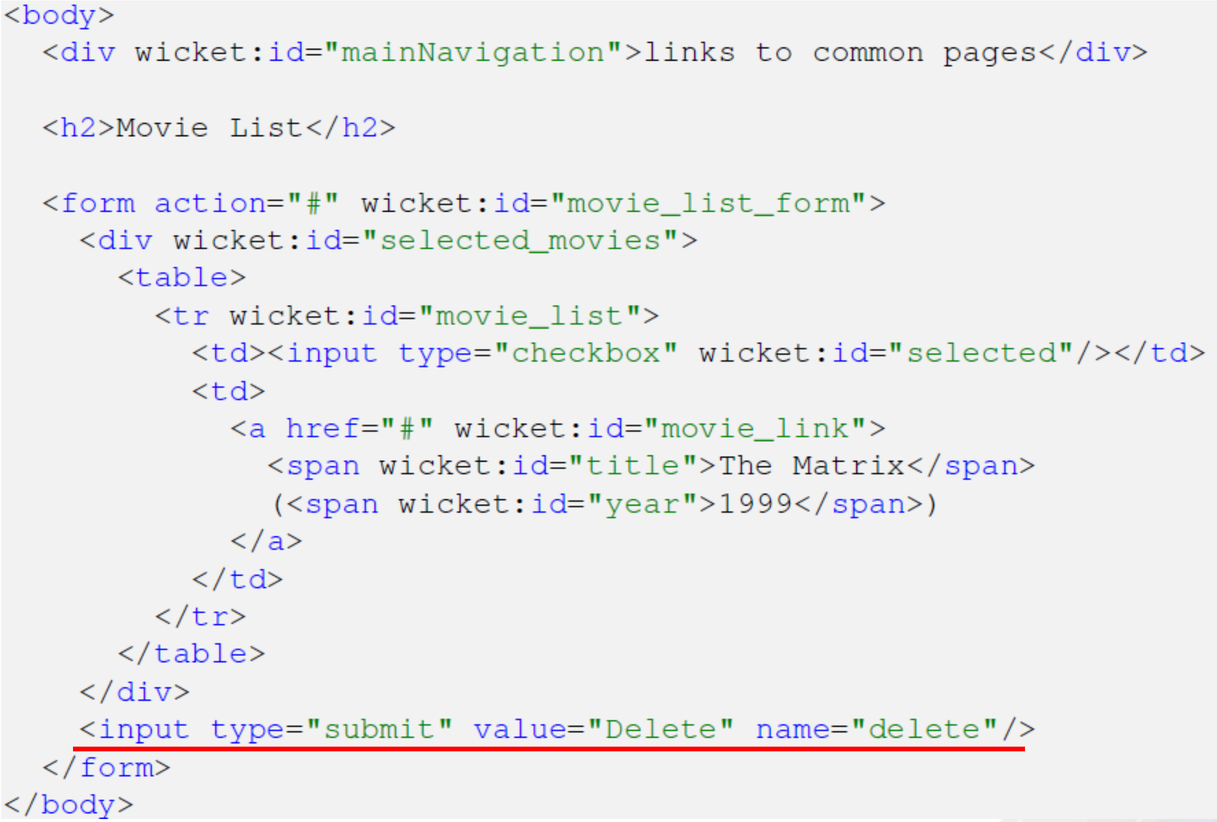
Checkboxes should be in a form element



```
<body>
  <div wicket:id="mainNavigation">links to common pages</div>

  <h2>Movie List</h2>

  <form action="#" wicket:id="movie_list_form">
   <div wicket:id="selected_movies">
    <table>
      <tr wicket:id="movie_list">
        <td><input type="checkbox" wicket:id="selected"/></td>
        <td>
          <a href="#" wicket:id="movie_link">
            <span wicket:id="title">The Matrix</span>
            (<span wicket:id="year">1999</span>)
          </a>
        </td>
      </tr>
    </table>
   </div>
   <input type="submit" value="Delete" name="delete"/>
  </form>
</body>
```

2: Check Boxes

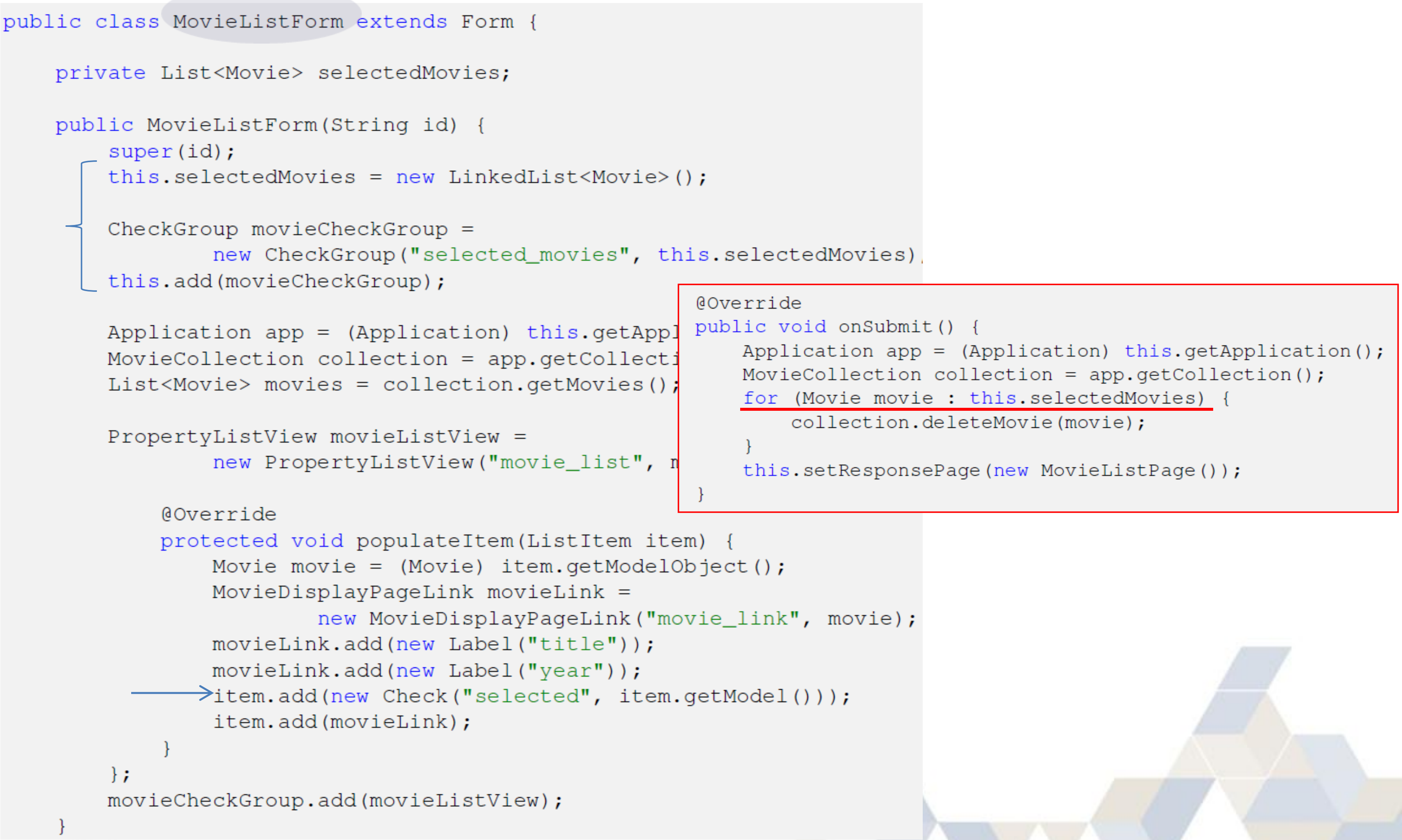
2: «MovieListPage.html» adding delete button:



```
<body>
  <div wicket:id="mainNavigation">links to common pages</div>

  <h2>Movie List</h2>

  <form action="#" wicket:id="movie_list_form">
    <div wicket:id="selected_movies">
      <table>
        <tr wicket:id="movie_list">
          <td><input type="checkbox" wicket:id="selected"/></td>
          <td>
            <a href="#" wicket:id="movie_link">
              <span wicket:id="title">The Matrix</span>
              (<span wicket:id="year">1999</span>)
            </a>
          </td>
        </tr>
      </table>
    </div>
    <input type="submit" value="Delete" name="delete"/>
  </form>
</body>
```

2: Check Boxes

3: When the button is clicked, selected movies should be deleted

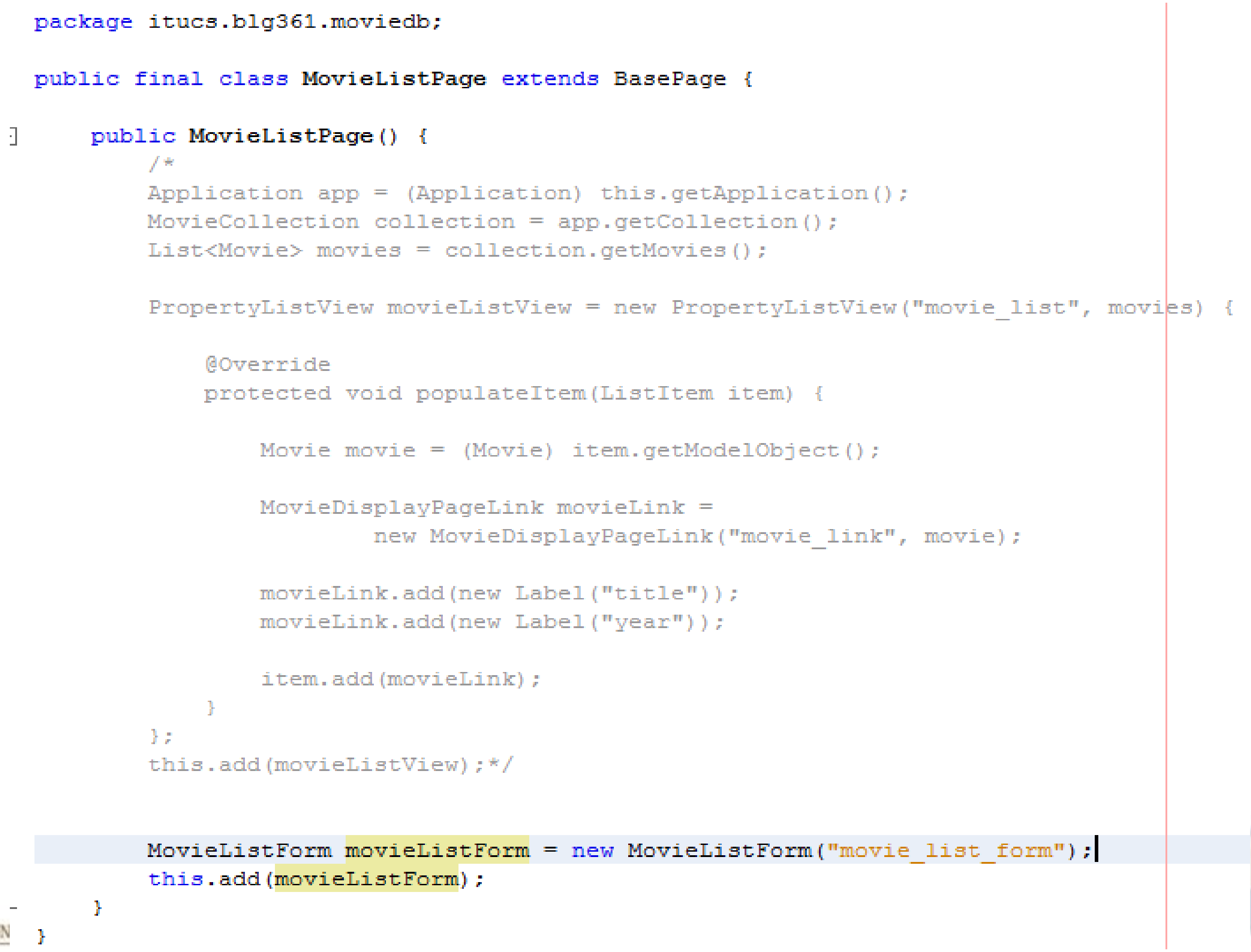


```
public class MovieListForm extends Form {

    private List<Movie> selectedMovies;

    public MovieListForm(String id) {
        super(id);
        this.selectedMovies = new LinkedList<Movie>();

        CheckGroup movieCheckGroup =
                new CheckGroup("selected_movies", this.selectedMovies),
        this.add(movieCheckGroup);

        Application app = (Application) this.getAppl
        MovieCollection collection = app.getCollecti
        List<Movie> movies = collection.getMovies();

        PropertyListView movieListView =
                new PropertyListView("movie_list", m

            @Override
            protected void populateItem(ListItem item) {
                Movie movie = (Movie) item.getModelObject();
                MovieDisplayPageLink movieLink =
                        new MovieDisplayPageLink("movie_link", movie);
                movieLink.add(new Label("title"));
                movieLink.add(new Label("year"));
                item.add(new Check("selected", item.getModel()));
                item.add(movieLink);
            }
        };
        movieCheckGroup.add(movieListView);
    }
```

```
@Override
public void onSubmit() {
    Application app = (Application) this.getApplication();
    MovieCollection collection = app.getCollection();
    for (Movie movie : this.selectedMovies) {
        collection.deleteMovie(movie);
    }
    this.setResponsePage(new MovieListPage());
}
```

2: Check Boxes

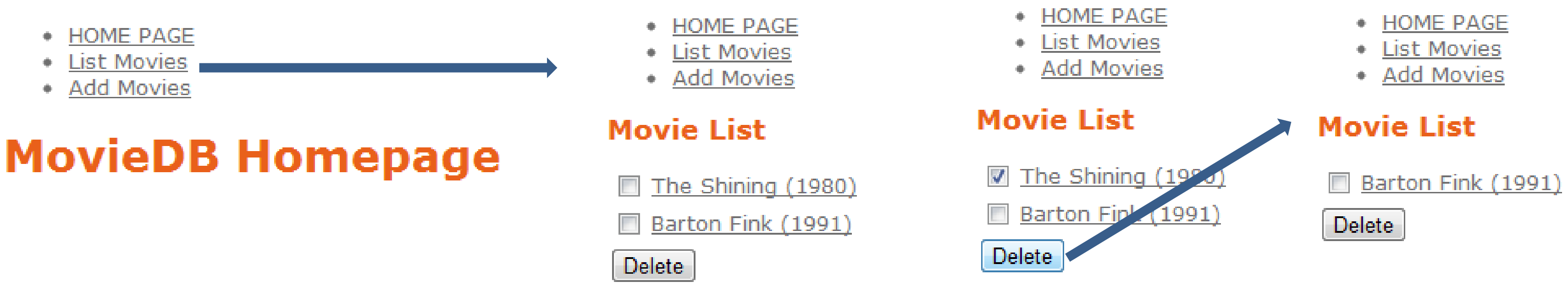
3: MovieListForm.java is ready, what about MoveListPage?



```
package itucs.blg361.moviedb;

public final class MovieListPage extends BasePage {

    public MovieListPage() {
        /*
        Application app = (Application) this.getApplication();
        MovieCollection collection = app.getCollection();
        List<Movie> movies = collection.getMovies();

        PropertyListView movieListView = new PropertyListView("movie_list", movies) {

            @Override
            protected void populateItem(ListItem item) {

                Movie movie = (Movie) item.getModelObject();

                MovieDisplayPageLink movieLink =
                        new MovieDisplayPageLink("movie_link", movie);

                movieLink.add(new Label("title"));
                movieLink.add(new Label("year"));

                item.add(movieLink);
            }
        };
        this.add(movieListView);*/


        MovieListForm movieListForm = new MovieListForm("movie_list_form");
        this.add(movieListForm);
    }
}
```

2: Check Boxes

SAMPLE RUN:



- HOME PAGE
- List Movies
- Add Movies

MovieDB Homepage

- HOME PAGE
- List Movies
- Add Movies

Movie List

□ The Shining (1980)

□ Barton Fink (1991)

Delete

- HOME PAGE
- List Movies
- Add Movies

Movie List

☑ The Shining (1980)

□ Barton Fink (1991)

Delete

- HOME PAGE
- List Movies
- Add Movies

Movie List

□ Barton Fink (1991)

Delete