

## Projektblatt P11 ( 16 P)

**Abgabe:** Donnerstag 28. November 2018, 12h

Entpacken Sie zunächst die Archiv-Datei `vorgaben-p11.zip`, in der sich die Rahmendateien für die zu lösenden Aufgaben befinden. Ergänzen Sie die Dateien durch Ihre Lösungen gemäß der Aufgabenstellung unten. Der hinzuzufügende Java-Sourcecode sollte **syntaktisch richtig** und **vollständig formatiert** sein. Alle Dateien sollten am Ende fehlerfrei übersetzt werden können.

Verpacken Sie die von Ihnen bearbeiteten `.java` Dateien für Ihre Abgabe in einem ZIP-Archiv mit dem Namen `IhrNachname.IhrVorname.P11.zip`, welches Sie auf Ilias hochladen.

Führen Sie dazu in dem Verzeichnis, in dem Sie die Dateien bearbeitet haben, folgenden Befehl auf der Kommandozeile aus:

```
zip IhrNachname.IhrVorname.P11.zip *.java
```

Nutzen Sie beim Bearbeiten der Aufgaben nur die Klassen und Methoden aus der Java API, welche explizit durch den Aufgabentext erlaubt sind.

## Aufgabe 1: Listener

10 Punkte

In dieser Aufgabe sollen Sie das Schiebepuzzle Programm vom letzten Projektblatt durch einen Listener so ergänzen, dass dieses interaktiv gespielt werden kann. Hierzu sollen Sie eine Listener-Klasse entwickeln, die auf Klicks auf die Puzzleteile reagiert. Da die Puzzleteile Buttons darstellen (`ZahlButton` ist abgeleitet von der Klasse  `JButton`), reicht hier ein  `ActionListener`. Die Klassen  `Schiebepuzzle`,  `PuzzlePanel` und  `ZahlButton`, die die Oberfläche definieren, sind bereits vorgegeben.

- (a) Legen Sie zunächst für den Listener in einer separaten Datei eine neue Klasse an, die das  `ActionListener` Interface implementiert. Statten Sie diese wie folgt aus:
- Definieren Sie ein Attribut  `parent` vom Typ  `PuzzlePanel`, die eine Referenz auf das Objekt darstellen soll, das den Listener erzeugt. Schreiben Sie dann einen Konstruktor, der einen passenden Parameter besitzt, mit dem dieses Attribut initialisiert wird. Über dieses Attribut gelangen Sie später an Informationen über den Spielzustand und können diesen ggf. anpassen.
  - Implementieren Sie die einzige Methode des  `ActionListener` Interfaces wie folgt:
    - Ermitteln Sie die  `ZahlButton` Instanz, die das Event ausgelöst hat.
    - Falls diese nicht das Leerfeld ist, ermitteln Sie mit Hilfe des Attributs  `parent`, ob diese  `ZahlButton` Instanz ein Nachbar des Leerfelds ist (siehe hierzu die Methoden der Klasse  `PuzzlePanel`). Ist dies der Fall, tauschen Sie den Inhalt der beiden Teile aus (mit der entsprechenden Methode der Klasse  `ZahlButton`) und aktualisieren Sie das  `parent` Objekt mit dessen  `update` Methode.
    - Prüfen Sie anschließend mit Hilfe der Instanz-Methode  `fertig` aus der Klasse  `PuzzlePanel`, ob das Spielfeld sich im Endzustand (Teile sortiert und Leerfeld links oben bzw. rechts unten) befindet. Ist dies der Fall, zeigen Sie dem Nutzer mit Hilfe einer der statischen Methoden aus der Klasse  `JOptionPane` an, dass das Spiel beendet ist und wieviele Züge der Spieler hierfür benötigt hat. Die Anzahl erhalten Sie wiederum

mit Hilfe der `parent` Instanz. Fragen Sie den User zugleich, ob er das Spiel beenden möchte oder nochmals spielen will. Im ersten Fall beenden Sie das Programm mit der Anweisung `System.exit(0)`. Im zweiten Fall reinitialisieren Sie das Spiel mit Hilfe der Instanzmethode `neuesSpiel` aus der Klasse `PuzzlePanel`. Das Bild unten zeigt Ihnen ein Beispiel dafür, wie die Anzeige der Dialogbox am Ende aussehen könnte.

- Importieren Sie alle Klassen aus der JDK einzeln (nicht mit einer Wildcard).

(b) Ergänzen Sie nun noch den Konstruktor der Klasse `PuzzlePanel` wie folgt:

- Generieren Sie eine Instanz der neuen Listener-Klasse. Übergeben Sie dem Konstruktor der Listener-Klasse dabei die aktuelle `PuzzlePanel` Instanz als Parameter.
- Registrieren Sie die Listener-Instanz dann bei allen Buttons.



## Aufgabe 2: Listener

6 Punkte

In dieser Aufgabe sollen Sie die GUI aus der Aufgabe 2 des letzten Projektblatts so ergänzen, das bei der Auswahl eines Jahres oder einer Kategorie bzw. beim Selektieren oder Deselektieren der Checkbox jeweils die Ausgabe in dem Textbereich entsprechend aktualisiert wird.

Ergänzen Sie hierzu die Klasse `OscarGUI` wie im Folgenden beschrieben:

- (a) Implementieren Sie zunächst die Methode `updateText`: Diese soll das in der Combobox `year` selektierte Jahr, die in der Combobox `category` selektierte Kategorie bestimmen sowie überprüfen, ob die Checkbox `winnersOnly` selektiert ist oder nicht. Anschließend soll die Klassenmethode `getNomineesInYear` aus der Klasse `FilterUtils` mit den entsprechenden Parameterwerten aufgerufen werden. Das resultierende Feld soll dann an die Methode `createText` übergeben und der Ergebnisstring dieser Methode verwendet werden, um den Text des Textbereichs `results` zu aktualisieren.
- (b) Ergänzen Sie die Klasse `OscarGUI` nun durch eine innere Klasse, die das Interface `ActionListener` implementiert. Implementieren Sie die einzige Methode des Interface so, dass die Instanzmethode `updateText` (aus der Klasse `OscarGUI`) aufgerufen wird.
- (c) Implementieren Sie nun noch die Methode `createListener`, die im Konstruktor der Klasse (nach der Methode `createComponents`) aufgerufen wird. In dieser Methode müssen Sie zunächst eine Instanz der in Aufgabenteil (b) entwickelten inneren Klasse generieren und diese dann bei den beiden Comboboxen und der Checkbox als Listener registrieren.

Testen Sie Ihre Implementierung, indem Sie das Programm starten und verschiedene Werte in den Comboboxen auswählen und die Checkbox verwenden um alle Nominierten oder nur die Gewinner anzuzeigen.