

## Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung: Lösungsvorschlag

### Aufgabe 8-1

### AWT/Swing

### Präsenz

In dieser Aufgabe sollen Sie eine grafische Benutzeroberfläche implementieren, mit der eine Figur erzeugt, bewegt und gelöscht werden kann. Nehmen Sie für diese Aufgabe die Klasse **Figur** von Übungsblatt 6, Aufgabe 6-3, zur Grundlage.

*Hinweis:* Damit Ihr Programm korrekt auf die Klasse **Figur** und die Klasse **Point** (die von der Klasse **Figur** benötigt wird) zugreifen kann, laden Sie die beiden Klassen von der Webseite herunter und speichern Sie im gleichen Ordner wie Ihre neuen Klassen **FigurFrame** und **FigurFrameMain**.

a) Die grafische Benutzeroberfläche soll wie folgt aussehen:



Es soll drei Buttons zum Erzeugen, Löschen und Bewegen einer Figur geben, die in der ersten Zeile direkt nebeneinander angeordnet sind. In der zweiten Zeile soll ein Ausgabebereich platziert werden, in dem Rückmeldung über das Erzeugen, Löschen oder Bewegen einer Figur gegeben werden kann.

Schreiben Sie eine Klasse **FigurFrame**, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse **FigurFrameMain**, die Sie im gleichen Ordner wie Ihre Klasse **FigurFrame** abspeichern. Die Klasse **FigurFrameMain** soll ein Objekt vom Typ **FigurFrame** erzeugen und das Fenster sichtbar machen.

### Lösung:

Bei der Implementierung der grafischen Benutzeroberfläche geht man wie folgt vor:

Deklarieren Sie eine Klasse **FigurFrame**, die die Hauptklasse Ihrer grafischen Benutzeroberfläche wird und deshalb von der Klasse **JFrame** erbt. Ihre Klasse **FigurFrame** soll vier Attribute haben:

- je ein Attribut vom Klassentyp  **JButton**  für Buttons zum Erzeugen, Löschen und Bewegen einer Figur,
- ein Attribut vom Klassentyp  **JTextArea** , das als Ausgabebereich für die spätere Rückmeldung über die Erzeugung, Löschung und Bewegung der Figur dient,

Schreiben Sie einen Konstruktor, der den **FigurFrame** mit einem entsprechenden Titel und Größe (wir wählen hier 700x200 Pixel) initialisiert. In dem Konstruktor sollen weiterhin alle Attribute korrekt initialisiert werden. Alle Buttons sollen in einem **JPanel** gruppiert werden. Das Layout des **ContentPane** des **FigurFrames** wird auf ein **GridLayout** mit zwei Zeilen und einer Spalte initialisiert und das Button-JPanel sowie der Ausgabebereich darauf platziert. Fügen Sie abschließend noch ein, dass das Programm ordnungsgemäß beendet wird, falls der **FigurFrame** geschlossen wird. Benutzen Sie dazu die Methode **setDefaultCloseOperation**.

**Auf der Webseite finden Sie die Implementierung der Klassen **FigurFrame** und **FigurFrameMain**.**

- b) Deklarieren Sie in Ihrer Klasse **FigurFrame** ein Attribut **aktuelleFigur** mit Klassentyp **Figur**. Erweitern Sie Ihre Klasse **FigurFrame** um eine Ereignisbehandlung für den Button zur Erzeugung einer Figur. Wird der Button zur Erzeugung gedrückt, soll eine neue Figur erzeugt und im Attribut **aktuelleFigur** gespeichert werden. Ist in dem Attribut bereits eine Figur gespeichert, soll keine neue Figur erzeugt werden, sondern eine Fehlermeldung im Ausgabebereich angezeigt werden. Alle nötigen Informationen zur Erzeugung einer Figur sollen vom Benutzer mit Hilfe der Klasse **JOptionPane** abgefragt werden. Nach der Erzeugung einer Figur soll der Benutzer im Ausgabebereich darüber informiert werden, d.h. es soll ausgegeben werden, welches Objekt erzeugt wurde und welche Koordinaten dessen Mittelpunkt hat.

**Lösung:**

Bei der Implementierung dieser Aufgabe müssen Sie bedenken, dass für den Button zunächst der **FigurFrame** als **ActionListener** registriert werden muss. Der **FigurFrame** muss dann eine Methode **actionPerformed** implementieren, die überprüft, welcher Button gedrückt wurde. Falls der Button zur Erzeugung gedrückt wurde, wird eine neue Methode aufgerufen, die die Erzeugung einer Figur, wie oben beschrieben, vornimmt.

**Auf der Webseite finden Sie die Implementierung der Klassen **FigurFrame** und **FigurFrameMain**.**

- c) Erweitern Sie Ihre Klasse **FigurFrame** um eine Ereignisbehandlung für den Button zum Löschen einer Figur. Wird der Button zum Löschen gedrückt, soll die aktuell gespeicherte Figur gelöscht werden. Dazu müssen Sie lediglich dem Attribut **aktuelleFigur** für die aktuelle Figur den Wert **null** zuweisen. Nach dem Löschen der Figur soll der Benutzer im Ausgabebereich darüber informiert werden. Falls allerdings keine aktuelle Figur gespeichert war, soll nach dem Knopfdruck eine Fehlermeldung im Ausgabebereich angezeigt werden.

**Lösung:**

Bei der Implementierung dieser Aufgabe müssen Sie bedenken, dass für diesen Button auch der **FigurFrame** als **ActionListener** registriert werden muss und die Methode **actionPerformed** erweitert werden muss.

**Auf der Webseite finden Sie die Implementierung der Klassen **FigurFrame** und **FigurFrameMain**.**

- d) Erweitern Sie Ihre Klasse **FigurFrame** um eine Ereignisbehandlung für den Button zum Bewegen einer Figur. Wird der Button zum Bewegen einer Figur gedrückt, soll der Benutzer zunächst mit Hilfe der Klasse **JOptionPane** gefragt werden, um wieviel die Figur bewegt werden soll (d.h. um wieviel die x-Koordinate des Mittelpunkts und um wieviel die y-Koordinate des Mittelpunkts bewegt werden soll), und anschließend die gespeicherte Figur entsprechend bewegt werden. Nach dem Bewegen soll der Benutzer im Ausgabebereich darüber informiert werden, d.h. es soll ausgegeben werden, welche Koordinaten der Mittelpunkt der Figur nun

hat. Falls keine aktuelle Figur gespeichert ist, soll eine Fehlermeldung im Ausgabebereich angezeigt werden.

#### Lösung:

Bei der Implementierung dieser Aufgabe müssen Sie bedenken, dass für diesen Button auch der `FigurFrame` als `ActionListener` registriert werden muss und die Methode `actionPerformed` erweitert werden muss.

Auf der Webseite finden Sie die Implementierung der Klassen `FigurFrame` und `FigurFrameMain`.

### Aufgabe 8-2

### AWT/Swing

### Hausaufgabe

In dieser Aufgabe sollen Sie eine grafische Benutzeroberfläche implementieren, mit der ein Fahrzeug erzeugt, beschleunigt und gelöscht werden kann. Nehmen Sie für diese Aufgabe die Klasse `Fahrzeug` von Übungsblatt 6, Aufgabe 6-4, zur Grundlage.

*Hinweis:* Damit Ihr Programm korrekt auf die Klasse `Fahrzeug` und die Klasse `Point` (die von der Klasse `Fahrzeug` benötigt wird) zugreifen kann, laden Sie die beiden Klassen von der Webseite herunter und speichern Sie im gleichen Ordner wie Ihre neuen Klassen `FahrzeugFrame` und `FahrzeugFrameMain`.

- a) Die grafische Benutzeroberfläche soll wie folgt aussehen:



Es soll drei Buttons zum Erzeugen, Löschen und Beschleunigen eines Fahrzeugs geben, die in der ersten Zeile direkt nebeneinander angeordnet sind. In der zweiten Zeile soll ein Ausgabebereich platziert werden, in dem Rückmeldung über das Erzeugen, Löschen oder Beschleunigen eines Fahrzeugs gegeben werden kann.

Schreiben Sie eine Klasse `FahrzeugFrame`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `FahrzeugFrameMain`, die Sie im gleichen Ordner wie Ihre Klasse `FahrzeugFrame` abspeichern. Die Klasse `FahrzeugFrameMain` soll ein Objekt vom Typ `FahrzeugFrame` erzeugen und das Fenster sichtbar machen.

#### Lösung:

Bei der Implementierung der grafischen Benutzeroberfläche geht man wie folgt vor:

Deklarieren Sie eine Klasse `FahrzeugFrame`, die die Hauptklasse Ihrer grafischen Benutzeroberfläche wird und deshalb von der Klasse `JFrame` erbt. Ihre Klasse `FahrzeugFrame` soll vier Attribute haben:

- je ein Attribut vom Klassentyp `JButton` für Buttons zum Erzeugen, Löschen und Beschleunigen eines Fahrzeugs,
- ein Attribut vom Klassentyp `JTextArea`, das als Ausgabebereich für die spätere Rückmeldung über die Erzeugung, Löschung und Beschleunigung des Fahrzeugs dient,

Schreiben Sie einen Konstruktor, der den `FahrzeugFrame` mit einem entsprechenden Titel und Größe (wir wählen hier 700x200 Pixel) initialisiert. In dem Konstruktor sollen weiterhin alle Attribute korrekt initialisiert werden. Alle Buttons sollen in einem `JPanel` gruppiert werden. Das Layout des `ContentPane` des `FahrzeugFrames` wird auf ein `GridLayout` mit zwei Zeilen und einer Spalte initialisiert und das Button-`JPanel` sowie der Ausgabebereich darauf platziert. Fügen Sie abschließend noch ein, dass das Programm ordnungsgemäß beendet wird, falls der `FahrzeugFrame` geschlossen wird. Benutzen Sie dazu die Methode `setDefaultCloseOperation`.

**Auf der Webseite finden Sie die Implementierung der Klassen `FahrzeugFrame` und `FahrzeugFrameMain`.**

- b) Erweitern Sie Ihre Klasse `FahrzeugFrame` um Ereignisbehandlungen für die Buttons zur Erzeugung, zum Löschen und Beschleunigen eines Fahrzeug.

**Lösung:**

Auf der Webseite finden Sie die Implementierung der Klassen `FahrzeugFrame` und `FahrzeugFrameMain`.

*Besprechung der Präsenzaufgaben in den Übungen ab 07.12.2016. Abgabe der Hausaufgaben bis Mittwoch, 21.12.2016, 14:00 Uhr über UniworX (siehe Folien der ersten Zentralübung).*

- Erstellen Sie zu jeder Aufgabe Klassen mit den entsprechenden Namen, die in der Aufgabe gefordert sind.
- Geben Sie Ihren Code **NICHT** in einer `.pdf`-Datei oder `.class`-Datei ab, sondern in einer `.java`-Datei!
- Erstellen Sie bitte **keine** geschachtelten `.zip`-Archive!