

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

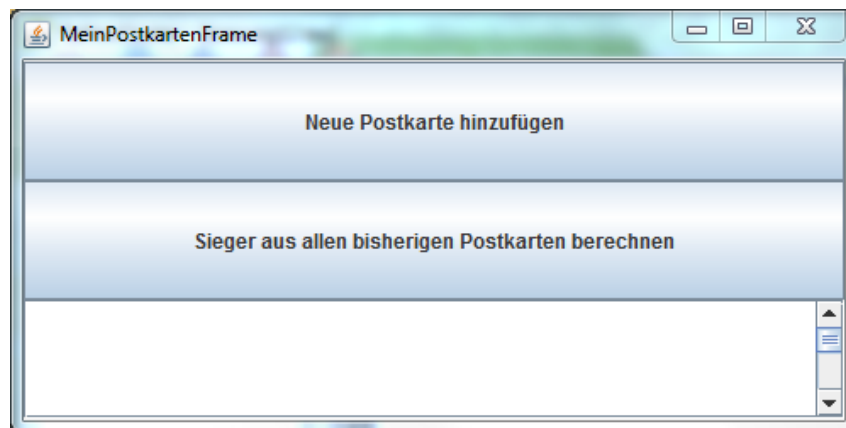
### Aufgabe 13-1

### Ballonwettbewerb

Präsenz

In einer Stadt werden mehrere tausend Ballons mit Absenderpostkarten losgelassen. Bis zu einem gewissen Stichtag kommt ein Teil der Postkarten mit Angabe des Fundortes zurück. Sieger ist, wessen Ballon am weitesten geflogen ist (in km), wobei bei gleicher Entfernung mehrere Sieger möglich sind.

- Schreiben Sie eine Klasse `Postkarte`, die eine Absenderpostkarte wie oben beschrieben repräsentiert. Die Klasse soll Attribute für den Namen des Teilnehmers und die zurückgelegte Entfernung (als Ganzzahl vom Typ `int`) haben. Fügen Sie einen Konstruktor zur Initialisierung der Attribute sowie "Getter" für beide Attribute hinzu.
- Implementieren Sie nun eine verkettete Liste zur Speicherung von Postkarten nach dem Beispiel in der Vorlesung. Nennen Sie die Klasse, die Ihre verkettete Liste darstellt, `MeinPostkartenListe` und die Klasse für Elemente in dieser Liste `MeinPostkartenListenelement`.
- Die grafische Benutzeroberfläche für die Verwaltung des Ballonwettbewerbs soll wie folgt aussehen:



Es soll zwei Buttons mit der oben genannten Aufschrift geben. Darunter soll ein Ausgabebereich platziert werden.

Schreiben Sie eine Klasse `MeinPostkartenFrame`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll und das Fenster erzeugt. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `MeinPostkartenFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `MeinPostkartenFrame` abspeichern.

Erweitern Sie Ihre Klasse `MeinPostkartenFrame` um eine Ereignisbehandlung für die beiden Buttons. Wird der Button "Neue Postkarte hinzufügen" gedrückt, soll der Benutzer zunächst mit Hilfe der Klasse `JOptionPane` nach dem Namen des Teilnehmers und der zurückgelegten Entfernung gefragt werden. Anschließend soll aus diesen Angaben ein Objekt vom Typ `Postkarte` erstellt werden und in einer Liste vom Typ `MeinPostkartenListe` gespeichert werden. Darüber soll der Benutzer im Ausgabebereich informiert werden.

Wird der Button "Sieger aus allen bisherigen Postkarten berechnen" gedrückt, sollen in der Liste aller bisher eingegebenen Postkarten alle Postkarten gesucht werden, die die weiteste

Entfernung erreicht haben. Der Benutzer soll anschließend darüber informiert werden, welche Postkarten die weiteste Entfernung erreicht haben. Beachten Sie, dass dies auch mehrere Postkarten sein können.

*Hinweis:* In Ihrer Klasse `MeinPostkartenFrame` werden Sie ein Attribut brauchen, um die Liste von Postkarten speichern zu können (Modell der GUI).

- d) Zur Lösung der Aufgabe sollen nun die generischen Klassen `LinkedList<E>` und `Iterator<E>` der Java-Bibliothek verwendet werden. Nehmen Sie die Benutzeroberfläche aus Aufgabe c) zur Grundlage und schreiben Sie eine neue Klasse `PostkartenFrame`. Das Aussehen der Benutzeroberfläche sowie die Funktionsweise der Buttons soll nicht verändert werden, allerdings soll anstelle eine Liste vom Typ `MeinePostkartenListe` nun eine Liste vom Typ `LinkedList<Postkarte>` verwendet werden, um alle Postkarten zu speichern. Ebenso soll zum Durchlaufen einer Liste stets ein Objekt der Klasse `Iterator<Postkarte>` benutzt werden. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `PostkartenFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `PostkartenFrame` abspeichern.

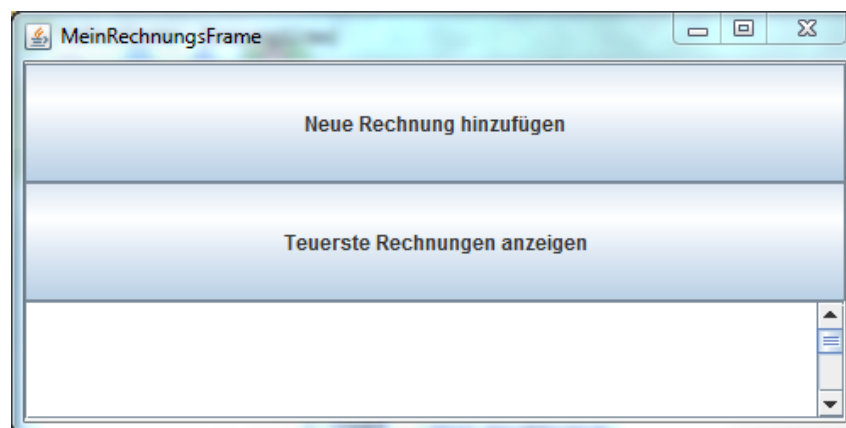
### Aufgabe 13-2

### Rechnungsverwaltung

*Hausaufgabe*

Eine Firma möchte beliebig viele Rechnungen erfassen, die sie an Kunden ausgestellt hat. Für jede einzelne Rechnung sollen die Rechnungsnummer und der Rechnungsbetrag angegeben werden. Die Firma möchte außerdem ihre besten Kunden herausfinden, indem sie den/die Kunden mit dem höchsten Rechnungsbetrag ermittelt.

- a) Schreiben Sie eine Klasse `Rechnung`, die eine Rechnung wie oben beschrieben repräsentiert. Die Klasse soll Attribute für die Rechnungsnummer und den Rechnungsbetrag (als Ganzzahl vom Typ `int`) haben. Fügen Sie einen Konstruktor zur Initialisierung der Attribute sowie "Getter" für beide Attribute hinzu.
- b) Implementieren Sie nun eine verkettete Liste zur Speicherung von Rechnungen nach dem Beispiel in der Vorlesung. Nennen Sie die Klasse, die Ihre verkettete Liste darstellt, `MeineRechnungsListe` und die Klasse für Elemente in dieser Liste `MeinRechnungsListenelement`.
- c) Die grafische Benutzeroberfläche für die Rechnungsverwaltung soll wie folgt aussehen:



Es soll zwei Buttons mit der oben genannten Aufschrift geben. Darunter soll ein Ausgabebereich platziert werden.

Schreiben Sie eine Klasse `MeinRechnungsFrame`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll und das Fenster erzeugt. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `MeinRechnungsFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `MeinRechnungsFrame` abspeichern.

Erweitern Sie Ihre Klasse `MeinRechnungsFrame` um eine Ereignisbehandlung für die beiden Buttons. Wird der Button "Neue Rechnung hinzufügen" gedrückt, soll der Benutzer zunächst

mit Hilfe der Klasse `JOptionPane` nach der Rechnungsnummer und dem Rechnungsbetrag gefragt werden. Anschließend soll aus diesen Angaben ein Objekt vom Typ `Rechnung` erstellt werden und in einer Liste vom Typ `MeineRechnungsListe` gespeichert werden. Darüber soll der Benutzer im Ausgabebereich informiert werden.

Wird der Button "Teuerste Rechnungen anzeigen" gedrückt, sollen in der Liste aller bisher eingegebenen Rechnungen alle Rechnungen gesucht werden, die den höchsten Rechnungsbetrag aufweisen. Der Benutzer soll anschließend darüber informiert werden, welche Rechnungen den größten Betrag aufweisen. Beachten Sie, dass dies auch mehrere Rechnungen sein können.

*Hinweis:* In Ihrer Klasse `MeinRechnungsFrame` werden Sie ein Attribut brauchen, um die Liste von Rechnungen speichern zu können (Modell der GUI).

- d) Zur Lösung der Aufgabe sollen nun die generischen Klassen `LinkedList<E>` und `Iterator<E>` der Java-Bibliothek verwendet werden. Nehmen Sie die Benutzeroberfläche aus Aufgabe c) zur Grundlage und schreiben Sie eine neue Klasse `RechnungsFrame`. Das Aussehen der Benutzeroberfläche sowie die Funktionsweise der Buttons soll nicht verändert werden, allerdings soll anstelle eine Liste vom Typ `MeineRechnungsListe` nun eine Liste vom Typ `LinkedList<Rechnung>` verwendet werden, um alle Rechnungen zu speichern. Ebenso soll zum Durchlaufen einer Liste stets ein Objekt vom Typ `Iterator<Rechnung>` benutzt werden. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `RechnungsFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `RechnungsFrame` abspeichern.

### Aufgabe 13-3

### Klausurvorbereitung 6 und 9 ECTS

### Hausaufgabe

Überlegen und formulieren Sie Fragen zu Themen aus der Vorlesung, bei denen Sie noch Probleme haben und senden Sie diese per Mail an Philipp Wendler ([philipp.wendler@lmu.de](mailto:philipp.wendler@lmu.de)). Die am meisten gestellten Fragen werden in der Zentralübung am 06.02.2019 behandelt.

*Besprechung der Präsenzaufgaben in den Übungen am 25.01.2019 und 28.01.2019. Abgabe der Hausaufgaben bis Montag, 04.02.2019, 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 nur die entsprechenden `.java`-Dateien ab. Wir benötigen **nicht** Ihre `.class`-Dateien.
- Geben Sie Java-Code nur in `.java`-Dateien ab. Java-Code in Bildern, PDF-Dokumenten und Text-Dateien wird nicht korrigiert.