

Bereich: Datenstrukturen (1)**Ziehung der Lottozahlen****Package:** `de.dhbwka.java.exercise.collections`**Klasse:** `Lottery`**Aufgabenstellung:**

Schreiben Sie ein Programm, das die Ziehung der Lottozahlen „6 aus 49 mit Zusatzzahl“ nachbildet. Verwenden Sie dabei eine Datenstruktur des Collection-Frameworks für die Speicherung der Zahlen.

Das Programm soll wiederholt eine ganzzahlige Zufallszahl aus dem Zahlenbereich 1 bis 49 generieren und jeweils der Collection hinzufügen (insgesamt sieben Zahlen [inkl. Zusatzzahl] in der Collection; keine Duplikate!).

Es soll anschließend die in der Collection gespeicherten Zahlen auf den Bildschirm (Konsolenfenster) ausgeben.

Die gezogenen Zahlen sollen aufsteigend sortiert ausgegeben werden, die Zusatzzahl separat, z.B.:

2 6 13 27 28 43 Zusatzzahl: 33

Hinweis:

Die ersten sechs gezogenen Zahlen sollen aufsteigend sortiert ausgegeben werden. Die siebte ist die Zusatzzahl und wird (unabhängig von ihrem Wert) hinten angestellt.

Frage:

Welches Interface des Collection-Frameworks und welche Implementierung dieses Interfaces eignen sich für diese Aufgabenstellung am besten?

Bereich: Datenstrukturen (1)**Bücherei****Package:** de.dhbwka.java.exercise.collections**Klasse:** Library**Aufgabenstellung:**

Schreiben Sie eine graphische Java-Applikation, die es erlaubt Bücherdaten einer Bücherei zu archivieren!



Als Archiv soll dazu eine Textdatei dienen.

Über die Oberfläche soll der Autor, Titel, Erscheinungsjahr und Verlag des betreffenden Buches eingegeben werden können (siehe auch „Beispiel für geschachteltes Layout“ im Foliensatz „Grafische Benutzeroberflächen mit Swing“). Nach der Eingabe eines Eintrages sollen diese vier Eigenschaften in ein File geschrieben werden. Zu Beginn der Applikation wird dieses File eingelesen.

Hinweis: Schreiben Sie eine Klasse `Book` und eine Klasse `Library`. Verwenden Sie eine geeignete `Collection` zum Speichern der Bücher!

Ihre Oberfläche sollte die folgende Funktionalität haben:

- Zu Beginn der Applikation einlesen eines Files, das die bisher existierenden Einträge enthält.
- Speichern von neuen Einträgen (im File), bereits vorhandene Einträge sollen nicht noch einmal gespeichert werden.
- Anzeige aller archivierten Einträge. Hier soll ausgewählt werden können ob die Einträge nach Autor, Titel, Erscheinungsjahr oder Verlag sortiert sind (siehe Beispiel oben).
- Jede sortierte Liste soll dabei in einem eigenen Pop-Up-Fenster dargestellt werden (siehe Beispiele).

Weitere Hinweise:

- Verwenden Sie das Interface `Comparator` zum Erzeugen geeigneter Comparatoren.
- Mit der Methode `split()` eines `String`-Objekts können Sie eine Zeichenkette unter Vorgabe eines Trenners in Teilzeichenketten zerlegen, siehe Javadoc. Die Rückgabe ist ein Feld der Teilzeichenketten `String[]`.

Beispiele für sortierte Ausgabe

