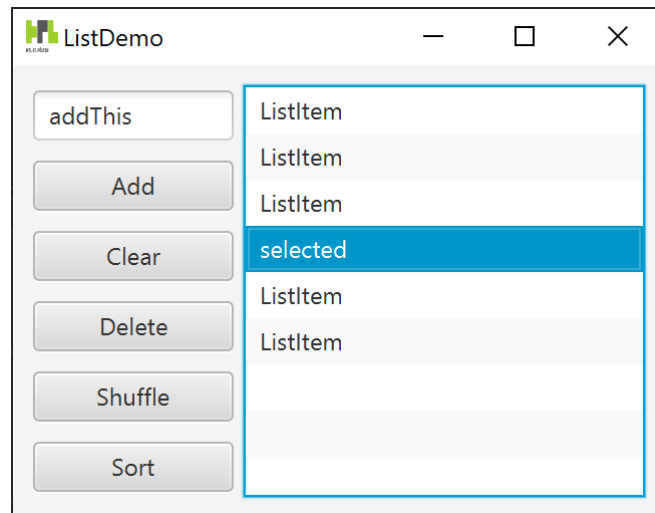


## 1 ListDemo

Schreibe eine Java FX Applikation, welche sich an folgendem Design orientiert:

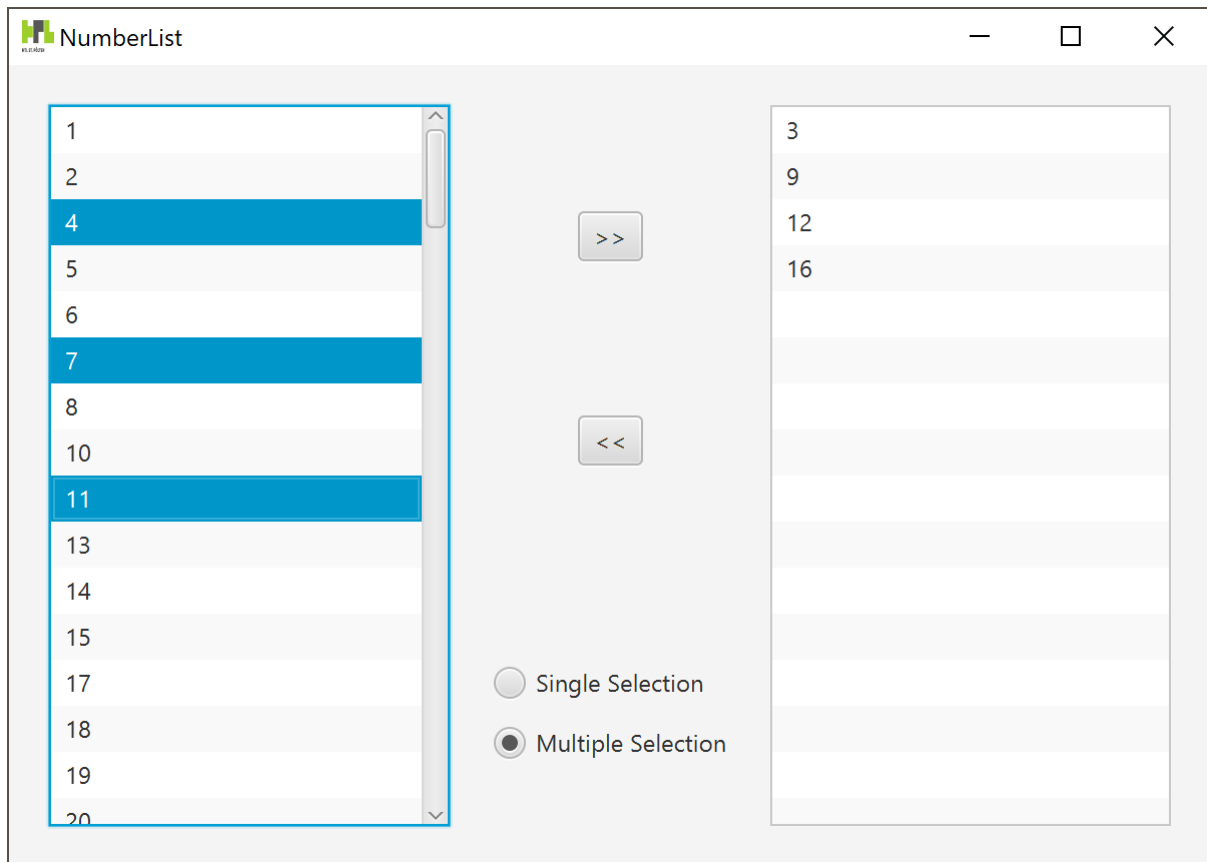


Funktionalität:

- **Enter** auf dem Textfeld und der Button **Add** fügen den Inhalt des Textfelds in die Liste ein. Nach dem Einfügen wird das Textfeld geleert.
- Leere Eingaben sind nicht zu übernehmen.
- **Clear** leert die Liste.
- **Delete** löscht den aktuell selektierten Eintrag.
- **Shuffle** mischt die Liste zufällig(`Collections.shuffle()`).
- **Sort** sortiert die Liste alphabetisch.

## 2 NumberList

Schreibe eine Java FX Applikation, welche sich an folgendem Design orientiert:

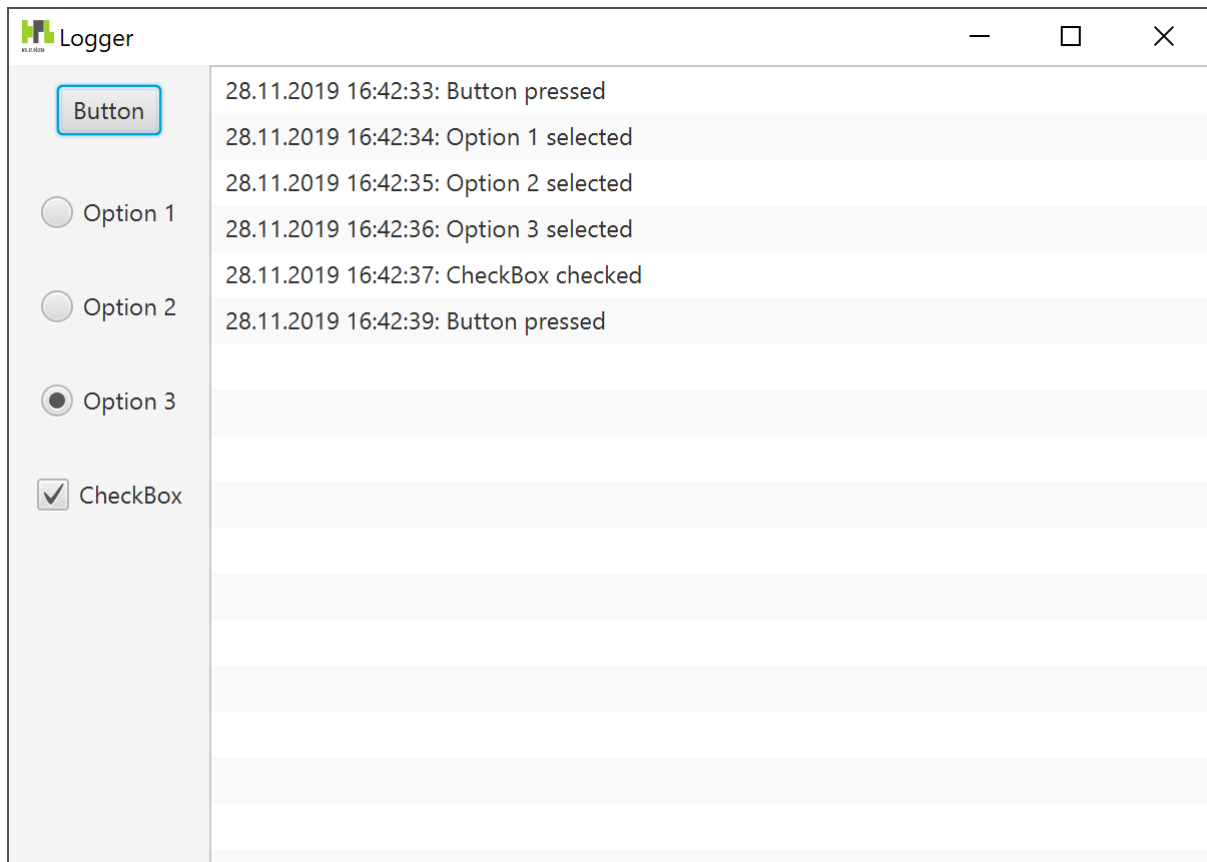


Funktionalität:

- Zu Beginn ist die linke Liste mit Integern im Bereich  $[0, 100]$  gefüllt.
- Die Buttons verschieben selektierte Elemente von der einen Liste in die andere.
- Die RadioButtons ermöglichen das Auswählen von mehreren Elementen.
- Die Listen sollen sortiert sein, daher erfolgt das Verschieben
  - eines einzelnen Elements so, dass mittels `binarySearch` die richtige Position gefunden wird und das Element dort in der Zielliste eingetragen wird.
  - mehrerer Elemente so, dass die Elemente verschoben werden und danach die Zielliste sortiert wird.

### 3 Logger

Schreibe eine Java FX Applikation, welche sich an folgendem Design orientiert:



Funktionalität:

- Ein Klick auf ein Bedienelement trägt einen entsprechenden Eintrag ins Log ein.
- Verwende für die Logeinträge eine Klasse ähnlich

```
public class LogEntry {  
    private static final DateTimeFormatter formatter =  
        DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm:ss");  
    private final String description;  
    private final LocalDateTime timestamp;
```

- "Sehr gute" Lösungen verwenden maximal 3 Action-Handler.