

Ein Startup im Einzelhandel benötigt ein einfaches Inventarmanagement Tool. Mit dem Tool sollen sie Produkte erstellt, Produkte nach Name (aufsteigend) sortieren, Produkte nach Preis (absteigend) sortieren, und Produkte exportieren können. Gewünscht ist es auch, eine exportierte Datei wieder zu importieren (nice to have). Ein GUI Entwurf wird in Figure 1 gezeigt.

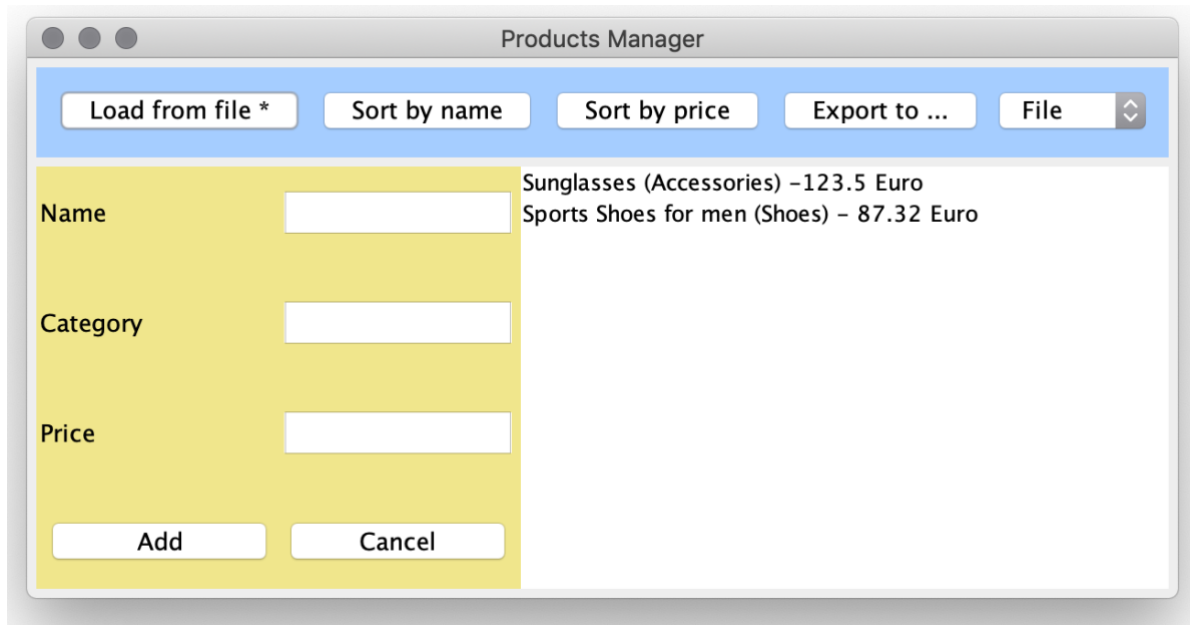


Figure 1: GUI Entwurf

Teil 1 (blau):

1. Load from file: hier wird JFileChooser verwendet um die Quelldatei auszuwählen. Die Einträge in der Datei werden dann in der JList (Teil 3) gezeigt werden
2. Sort by name: die Einträge in der Liste (JList Teil 3) sollen nach Name aufsteigend sortiert werden
3. Sort by price: die Einträge in der Liste (JList Teil 3) sollen nach Preis absteigend sortiert werden
4. Export to: die Einträge in der Liste (ggf. nach der Sortierung) sollen in der Reinform exportiert werden. Das Tool bietet 3 Export Alternativen an: File, Email, und Fax. Es soll aber möglich, die Liste der Alternativen zu erweitern (💡 Design Patterns?). Wenn „File“ als Export Methode ausgewählt wird, werden die Einträge in einer Text Datei gespeichert. Mit JFileChooser soll der Nutzer eine Datei auswählen¹. Die andere Export Methoden sollen nur eine Message (mittels JOptionPane) zeigen, z.B., „Die Liste der Produkte wird per Email gesendet“.

Teil 2 (gelb)

1. Add: der Nutzer soll mit dem Teil Produkte erstellen können. Ein Produkt (💡 Klasse?) hat Name, Kategorie, und Preis. Mit dem „Add“ Knopf wird ein Objekt erstellt und zu der List (Teil 3) hinzugefügt.
2. Cancel: mit dem Knopf werden alle drei Felder zurückgesetzt.

Teil 3 (weiß)

Hier wird ein JList Element gezeigt, welches alle Produkte nach dem Format „Name (Kategorie) – Preis Euro“ (💡 toString() überschreiben?) auflistet.

¹ Notfalls kann der Dateipfad in dem Code hartkodiert werden

Hinweise

JFileChooser

JFileChooser kann mit dem Code gezeigt werden

```
JFileChooser chooser = new JFileChooser();
```

```
// Dateien öffnen
```

```
int returnVal = chooser.showOpenDialog(parent);
```

```
// ODER
```

```
// Dateien speichern
```

```
int returnVal = chooser.showSaveDialog(parent);
```

//ein Container oder null kann als parent eingegeben werden

```
if (returnVal == JFileChooser.APPROVE_OPTION) {
```

```
    File file = chooser.getSelectedFile();
```

```
    // von der Datei lesen oder in die Datei schreiben
```

```
}
```

JOptionPane

Eine Message kann mit JOptionPane gezeigt werden:

```
JOptionPane.showInternalMessageDialog(
```

```
    parent,
```

```
    "Die Liste der Produkte wird per Email gesendet",
```

```
    "Export",
```

```
    JOptionPane.INFORMATION_MESSAGE);
```

//ein Container oder null kann als parent eingegeben werden

Layouts

- GUI Elements können in einem Container (JPanel) gruppiert werden.
- BorderLayout, GridLayout, und FlowLayout können fürs Tool relevant sein