

Aufgabe für Studierende der Informatik (AI)

Umsetzung einer grafischen Oberfläche für das erstellte Programm aus Praktikum 1 mit Hilfe von Java Swing. Haben Sie Probleme bei der Implementierung, so sollten Sie Fragen in den Vorlesungsfolien nachschlagen und sich (erneut) die Screencasts anschauen. Weiterhin sind folgende Links sehr hilfreich:

- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>
- https://www.tutorialspoint.com/swing/swing_controls.htm
- <https://docs.oracle.com/javase/tutorial/uiswing/events/eventsandcomponents.html>

Die vollständige Realisierung kann am Ende wie folgt aussehen:

Maschine:	Erzeugnis:	Abhängigkeit:
Typ: Erzeuger Name: Apfelpflücker Kosten: 20.0	Name: Apfel Herstellungskosten: 0.5 Verkaufswert: 1.5	
Typ: Erzeuger Name: Bananenpflücker Kosten: 30.0	Name: Banane Herstellungskosten: 1.0 Verkaufswert: 3.0	

Abbildung 1 – InputPanel und MaschinenPanel: Anlegen eines Erzeugerentwurfs

Maschine:	Erzeugnis:	Abhängigkeit:
Typ: Erzeuger Name: Apfelpflücker Kosten: 20.0	Name: Apfel Herstellungskosten: 0.5 Verkaufswert: 1.5	
Typ: Erzeuger Name: Bananenpflücker Kosten: 30.0	Name: Banane Herstellungskosten: 1.0 Verkaufswert: 3.0	
Typ: Verwerter Name: Apfelpresse Kosten: 40.0	Name: Apfelsaft Herstellungskosten: 2.0 Verkaufswert: 5.0	Name: Apfel Herstellungskosten: 0.5 Verkaufswert: 1.5 Anzahl: 3

Abbildung 2 – InputPanel und MaschinenPanel: Anlegen eines Verwerterentwurfs

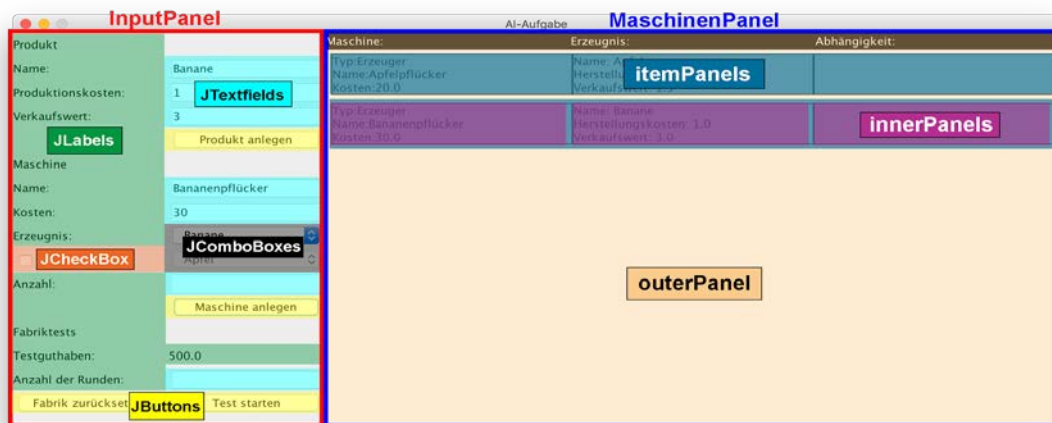


Abbildung 3 – InputPanel und MaschinenPanel: Anordnung und Zuordnung der Elemente

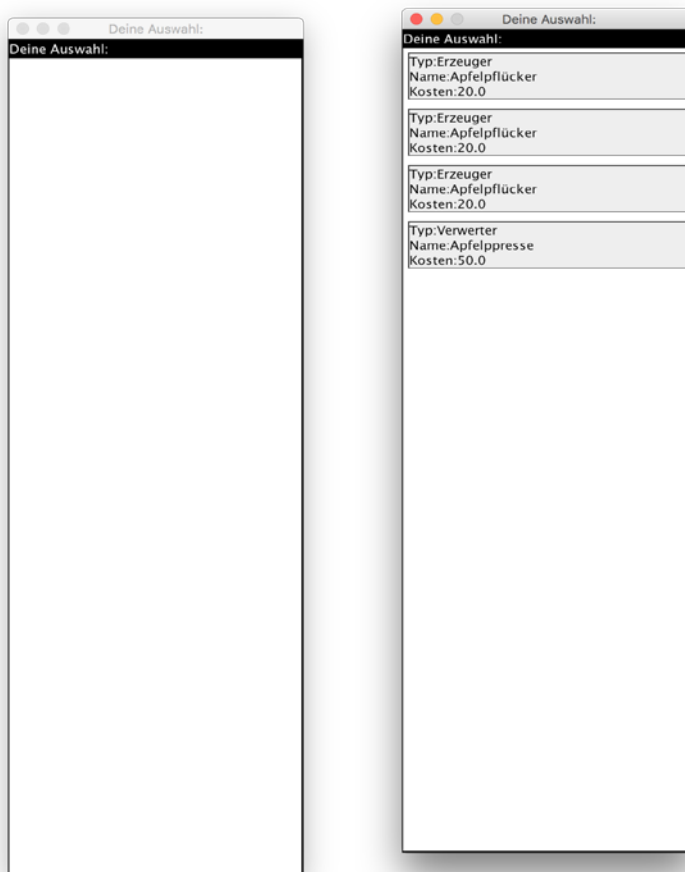


Abbildung 4 – AuswahlPanel: Keine Maschinen hinzugefügt (links) und Maschinen hinzugefügt (rechts)

Pflichtaufgabe 2:

In dieser Aufgabe wird die Funktionalität des Programms leicht geändert. Zunächst gilt es die Produkte anzulegen, die dann später als Erzeugnis oder Abhängigkeit einer Maschine auswählbar sind. Anstatt Maschinen direkt der Fabrik hinzuzufügen, werden zunächst Prototypen (bzw. Mustervorlagen) dieser Maschinen zwischengespeichert und in einer Liste angezeigt (Abbildung 1, 2 und 3). Aus dieser Liste können anschließend die gewünschten Maschinen ausgewählt und der Fabrik hinzugefügt werden (Abbildung 4). Abschließend wird über das InputPanel (Abbildung 1, 2 und 3 linker Abschnitt) ein Testdurchlauf gestartet.

a. Passen Sie die Datenstruktur **Fabrik** auf den gewünschten Programmablauf an.

- Erweitern Sie die Klasse um folgende Methoden:
 - `getMaschinen()`: Getter-Methode für die `ArrayList` der Maschinen.
 - `resetFabrik()`: Diese Methode setzt den Wert des Testguthabens auf den Wert des Firmenguthabens zurück und entfernt alle Maschinen aus der `ArrayList`.
- Folgende Methoden müssen zudem verändert werden:
 - Konstruktor: Setzen Sie bereits im Konstruktor den Wert des Testguthabens auf den Wert des übergebenen Guthabens.
 - `fuegeMaschineHinzu(Maschine maschine)`: Eine Maschine wird gekauft. Reduzieren Sie hier zusätzlich das Testguthaben um die Kosten der übergebenen Maschine.
 - `entferneMaschine(int index)`: Eine Maschine wird verkauft. Erhöhen Sie das Testguthaben um die Kosten der Maschine, bevor Sie diese entfernen.
 - `firmaTesten(int rundenanzahl)`: Da das Testguthaben nun bereits im Konstruktor beachtet wird sollten Sie hier zunächst die Zeile entfernen, in welcher das Testguthaben auf den Wert des Fabrikguthabens gesetzt wird. Weiterhin sollten nun die Zeilen entfernt werden, welche die Kosten der Maschinen von dem Testguthaben abziehen. Dies wird bereits in der `fuegeMaschineHinzu()`-Methode getätigt.

b. Überarbeiten Sie die Klasse **Maschine** und ihre Unterklassen.

- Fügen Sie der Klasse **Maschine** folgende Eigenschaft hinzu:
 - `String typ`
→ Diese wird benötigt, um eine Maschine in der Anzeige beschreiben zu können.
- Im Konstruktor der **Maschine**, des **Erzeugers** und des **Verwerter**:
 - Setzen Sie jeweils die Variable `typ` auf einen sinnvollen und aussagekräftigen Wert (für die Maschine z.B. "Maschine", für den Erzeuger "Erzeuger" etc.)
- Ergänzen Sie die Klassen um folgende Funktionen:
 - **Maschine** um eine Getter-Methode für die Variable `typ`.
 - **Erzeuger** um eine Getter-Methode für das Erzeugnis (vom Typ Produkt).
 - **Verwerter** um eine Getter-Methode für die Abhängigkeit (vom Typ Produkt) und eine Getter-Methode für die Anzahl der Abhängigkeit.

c. Erstellen Sie die Datenstruktur **PanelManager**, in der das Programm-Fenster geöffnet und alle beinhalteten **JPanel** verwaltet werden. Zudem ist diese Klasse für die Kommunikation zwischen den eingebetteten **JPanel**, sowie für die Kommunikation zwischen den **JPanel** und der Fabrik zuständig.

- Dafür benötigen Sie mindestens diese `private` Eigenschaften, welche spätestens im Konstruktor instanziiert werden müssen:
 - `ArrayList<Produkt> produktEntwuerfe`:
→ Hier werden später die Produktentwürfe gesammelt.
 - `ArrayList<Maschine> maschinenEntwuerfe`:
→ Hier werden später die erstellten Maschinenentwürfe gesammelt.
 - `Fabrik fabrik`

- Diese Klasse benötigt folgende Methoden:
 - `void addOrReplaceProduktentwurf(Produkt produkt):` Diese Methode nimmt ein Produkt entgegen und fügt es der Liste von Produktentwürfen hinzu. Falls bereits ein Produkt mit dem Namen vorhanden ist, wird dieses lediglich ausgetauscht anstatt das neue Produkt hinzuzufügen.
 - `void addOrReplaceMaschinenentwurf(Maschine maschine):` Identisch zu der `addOrReplaceProduktentwurf`, lediglich mit der Maschinenliste.
 - `ArrayList<Maschine> getMaschinenentwürfe():`
→ Gibt die Liste der Maschinenentwürfe zurück.
 - `ArrayList<Produkt> getProduktentwürfe():`
→ Gibt die Liste der Produktentwürfe zurück.
- Um den Panel (`InputPanel`, `MaschinenPanel` und `AuswahlPanel`), welche in einem späteren Schritt realisiert werden, die Kommunikation mit der Fabrik zu ermöglichen, werden folgende Methoden benötigt:
 - `void addMaschineToFabrik(int index):`
→ Diese Methode fügt den Maschinenentwurf an der Position `index` der Fabrik hinzu.
 - `void removeMaschineFromFabrik(int index)`
→ Entfernt die Maschine an der Position `index` aus der Fabrik (hier bezieht sich der Index auf die Position in der `ArrayList` der Fabrik und nicht die position in den Maschinenentwürfen).
 - `void resetFabrik()`
→ greift momentan lediglich auf die `resetFabrik()`-Methode der Fabrik zu.
 - `ArrayList<Maschine> getAktiveMaschinen()`
→ Gibt eine Liste der Maschinen zurück, die bereits der Firma hinzugefügt wurden.
 - `double getTestguthaben()`
→ Gibt das momentane Testguthaben der Fabrik zurück.

Diese Methoden werden später für die Aktualisierung der Fenster benötigt.

d. Erstellen Sie die Datenstruktur `InputPanel`, die für die Erstellung der Produktentwürfe und der Maschinenentwürfe sowie den Start der Testläufe zuständig ist (siehe Abbildungen 1- 3).

- Diese Klasse soll von der Klasse `JPanel` erben und benötigt folgende Elemente:
 - `JLabel` für:
 - Name des Produktes,
 - Produktkosten,
 - Verkaufswert des Produktes,
 - Name der Maschine,
 - Kosten der Maschine,
 - Erzeugnis der Maschine,
 - Anzahl der Bedingungen,
 - Beschreibung des Testguthabens,
 - Wert (höhe) des Testguthabens,
 - Anzahl der Testrunden
 - "Überschriften" für Produkt, Maschine und Fabriktests (optional)
 - `JTextField` für:
 - Name des Produkts,
 - Kosten des Produkts,
 - Verkaufswert des Produkts,
 - Name der Maschine,
 - Kosten der Maschine,
 - Anzahl der Bedingungen,
 - Anzahl der Testrunden

- 2 x JComboBox mit dem Typ String für die Erzeugnisliste und die Bedingungsliste.
 - Eine JCheckBox, um auszuwählen, dass eine Abhängigkeit von einem anderen Produkt benötigt wird. In diesem Fall soll also ein Verwerter erstellt werden.
 - JButton, um
 - ein Produkt anzulegen,
 - eine Maschine anzulegen,
 - die Fabrik zurückzusetzen und
 - den Fabriktest zu starten.
 - Zudem wird die private Eigenschaft PanelManager panelManager benötigt. Diese muss im Konstruktor dieser Klasse übergeben werden.
- Um die Benutzeroberfläche in folgenden Schritten testen zu können, müssen Sie den PanelManager um die private Eigenschaft InputPanel inputPanel erweitern. Fügen Sie dieser Klasse zudem die Methode void start() hinzu, welche ein JFrame erstellt und das InputPanel auf dessen linken Seite einbettet. Setzen Sie die bevorzugte Größe zunächst auf 1000*500.
- Bauen Sie das InputPanel ähnlich dem Beispiel (Abb.1-3) auf. Verwenden Sie dabei ein geeignetes Layout (z.B. BoxLayout, GridLayout, etc.). Achten Sie darauf, dass Sie in der Klasse App zunächst einen PanelManager anlegen und dessen start()-Methode aufrufen müssen bevor Sie ihr Fenster sehen.
- Fügen Sie dem JButton zum Anlegen eines Produkts einen ActionListener hinzu. Wird auf diesen gedrückt, so soll mithilfe der Textfeldeingaben (Produktname, Produktkosten, Verkaufskosten des Produkts) ein neues Produkt erstellt, sowie der Liste mit Produktentwürfen (im PanelManager) hinzugefügt werden.
- Definieren Sie im InputPanel eine Methode public void updateData(). Hier müssen die JComboBoxen entleert und mit den Namen aller Produktentwürfe neu befüllt werden. Nutzen Sie dafür die Methode addItem(String item). Anschließend soll das JLabel für den Betrag des Testguthabens auf das aktuelle Testguthaben gesetzt werden. Um die Implementierung zu nutzen müssen Sie in den Methoden des PanelManagers
 - addOrReplaceProdukt(Produkt produkt),
 - addMaschineToFabrik(int index),
 - removeMaschineFromFabrik(int index)
 jeweils die Methode updateData() aufrufen.
- Fügen Sie auch dem JButton zum Anlegen eines Maschinenentwurfs einen ActionListener hinzu. Unterscheiden Sie dabei zwei Fälle:
 - CheckBox ausgewählt: ein Verwerter wird erstellt
 - CheckBox NICHT ausgewählt: ein Erzeuger wird erstellt.
 Für die Erstellung benötigte Eigenschaften können aus den jeweiligen Textfeldern entnommen werden. Die benötigten Produkte können anhand des aktiven Items der JComboBoxen (der Index des aktiven Items) aus der Produktentwurfs-Liste des PanelManagers entnommen werden. Nachdem eine neue Instanz der jeweiligen Maschine erstellt wurde, muss diese der Maschinenentwurfs-Liste des PanelManagers hinzugefügt werden.
- Abschließend müssen die JButtons zum zurücksetzen der Fabrik und starten eines Testlaufes mit ActionListenern versehen werden. Rufen Sie hierzu die jeweilige Funktion des PanelManagers auf. Beachten Sie im Falle des Teststarts, dass die Anzahl der Testrunden aus dem dafür vorgesehenen Textfeld entnommen werden muss.
- e. Erstellen Sie eine Datenstruktur MaschinenPanel, welche von JPanel erbt und für die Auflistung der Maschinenentwürfe zuständig ist (Abbildungen 1, 2, 3 - rechte Seite). In dieser Klasse werden Sie vermehrt in Kontakt mit der Verschachtelung von JPanel-Objekten kommen. Orientieren Sie sich an Abbildung 3 und wählen Sie für jedes Panel ein geeignetes Layout.

- Für die Klasse `MaschinenPanel` werden folgende `private` Eigenschaften benötigt:
 - `PanelManager panelManager`
 - Einen Konstruktor dem der `PanelManager` übergeben wird.
 - Erstellen Sie die folgende Methode um Ihre geänderten Werte zu aktualisieren:
 - `updateData()`
→ Hier sollten zunächst alle Elemente innerhalb des `MaschinenPanels` entfernt werden.
 - Legen Sie nun das `outerPanel` (siehe Abbildung 3) an, welches für die Sammlung der einzelnen `itemPanel` zuständig ist.
 - Holen Sie sich die Liste der Maschinenentwürfe und iterieren Sie über diese. Für jeden Maschinenentwurf müssen die folgenden drei `JPanel` zur Visualisierung der Maschineninformationen angelegt werden (in Abbildung 3 als `InnerPanels` bezeichnet):
 - Maschinendaten (Typ, Name, Kosten),
 - Erzeugnisdaten (Name, Herstellungskosten, Verkaufswert) und
 - Abhängigkeitsdaten (Name, Herstellungskosten, Verkaufswert, Anzahl).

Achtung: Abhängigkeitsdaten werden nur gefüllt, wenn die Maschine vom Typ des Verwerfers ist.
 - Erstellen Sie ein weiteres `Panel`, in welchem Sie die drei `innerPanel` einbetten (in Abbildung 3 als `itemPanels` bezeichnet). Fügen Sie diesem `itemPanel` zudem einen `MouseListener` hinzu, welcher nach einem Mausklick den momentanen Maschinenentwurf der Fabrik hinzufügt. Fügen Sie dieses `itemPanel` dem `outerPanel` hinzu.
Nachdem für jeden Maschinenentwurf ein `itemPanel` angelegt und dem `outerPanel` hinzugefügt wurde, kann dieses `outerPanel` im *Norden* des `MaschinenPanels` hinzugefügt werden.
Rufen Sie aus anschließend die Methode `revalidate()` des `MaschinenPanels` auf, um die Darstellung durch Swing zu aktualisieren.
Optional kann noch ein Tabellenkopf hinzugefügt werden.
 - Ergänzen Sie den `PanelManager` um die `private` Eigenschaft
 - `MaschinenPanel maschinenPanel`
 - und fügen Sie diese in der `start()`-Methode dem Frame auf der rechten Seite hinzu.
 - Verändern Sie im `PanelManager` zudem die Methode `addOrReplaceMaschinenentwurf(...)`, so dass nach jeder Veränderung die `updateData()`-Methode und ggf. noch `repaint()` des `MaschinenPanels` aufgerufen wird.
 - Bei richtiger Implementierung sollte nun nach dem Hinzufügen eines Maschinenentwurfs dieser wie in Abbildungen 1 und 2 angezeigt werden. Weiterhin können diese Entwürfe bereits der Fabrik hinzugefügt und ein Testlauf gestartet werden. Funktioniert dies wie gewollt, so dürfte nun die in Praktikum 1 implementierte Ausgabe auf der Konsole erscheinen.
- f. In Aufgabe e) haben Sie das `MaschinenPanel` realisiert, welches zur Auflistung der Maschinenentwürfe genutzt wird.
Implementieren Sie die Datenstruktur `AuswahlPanel`, in welcher die Maschinen aufgelistet werden, welche bereits der Fabrik hinzugefügt wurden. Da sich diese Klassen sehr ähneln, kann der Großteil des Codes übernommen werden und in dem folgenden Abschnitt werden vorwiegend die Änderungen genannt.
- Die Klasse `AuswahlPanel` soll von der Klasse `JPanel` erben und mindestens folgende `private` Eigenschaft besitzen:
 - `PanelManager panelManager` welche im Konstruktor übergeben wird.
 - Lege Sie auch hier auf dieselbe Weise eine Methode `updateData()` an.

- Kopieren Sie zunächst den Code aus dem `MaschinenPanel` und bauen Sie folgende Veränderungen ein:
 - Anstatt die Liste der Maschinenentwürfe abzufragen wird hier die Liste der (in der Fabrik) aktiven Maschinen benötigt.
 - Es wird lediglich ein `innerPanel` benötigt für die Maschineninformationen (Typ, Name, Kosten).
 - Der `MouseListener` des `itemPanels` wird ausgetauscht durch folgende Funktionalität: Wird auf das `JPanel` gedrückt, so soll die entsprechende Maschine entfernt werden.
- Ergänzen Sie ein letztes Mal den `PanelManager`:
 - Der `PanelManager` benötigt nun die private Eigenschaft `AuswahlPanel` `AuswahlPanel`, welche spätestens im Konstruktor instanziiert werden muss.
 - In folgenden Methoden muss zudem die `updateData()`-Methode des `AuswahlPanel`-Objekts aufgerufen werden:
 - `addMaschineToFabrik(int index)`
 - `removeMaschineFromFabrik(int index)`
 - `resetFabrik()`
 - Erweitern Sie zudem die `start()`-Methode, sodass ein zweites Fenster erstellt und in diesem das `AuswahlPanel` eingebettet wird. Bei korrekter Implementierung sollte die Anwendung nun vollständig genutzt werden können. Der Aufbau der Fabrik kann vollständig über die grafische Oberfläche getätigt werden, die Ausgabe eines Testlaufs wird jedoch weiterhin nur auf der Konsole ausgegeben.

VIEL ERFOLG!