

## Aufgabenstellung für das *Einführungsbeispiel* der LVA “Objektorientiertes Programmieren“

**Abgabeschluss für den Source Code:      Mittwoch 25.03.2018 um 23:55 Uhr**

**WICHTIG:**

Wird das Beispiel nicht fristgerecht abgegeben, haben Sie in Folge nicht mehr die Möglichkeit, die LVA im laufenden Semester erfolgreich abzuschließen!

### **Allgemeines**

Die Beispiele müssen eigenständig ausgearbeitet werden.

### **Umsetzung und Abgabe**

- Geben Sie Ihrem Eclipse-Projekt einen eindeutigen Namen. Halten Sie sich dabei an folgende Benennungsvorschrift:

**Eclipse-Projekt:**      *matrNr\_Einfuehrungsbeispiel\_Nachname*

**Beispiel:**              *00123456\_Einfuehrungsbeispiel\_Mustermann*

**ACHTUNG:** Wenn Sie sich nicht an die Benennungsvorschrift halten, kann Ihre Abgabe nicht gewertet werden!

- Name und Matrikelnummer muss zu Beginn jeder Klasse in einem Kommentarblock angeführt werden!
- Exportieren Sie das Projekt (inklusive aller für Eclipse notwendigen Dateien, z.B.: .classpath, .project) als ZIP-Datei. Eine Anleitung dazu finden Sie in TUWEL.

### **Kontakt:**

Inhaltliche Fragen:      [Diskussionsforum](#) in TUWEL

Organisatorische Fragen: [oop@ict.tuwien.ac.at](mailto:oop@ict.tuwien.ac.at)

## Aufgabe: Hardware-Platine

### Ziel:

Vertraut machen mit Java und der Entwicklungsumgebung Eclipse.  
Erwerb konkreter Erfahrung und Kompetenz in der Umsetzung folgender Konzepte:  
Klassen, Abstrakte Klassen, Objekte, Objekterzeugung, Vererbung.

### Einleitung:

In dieser Aufgabenstellung wird ein (stark vereinfachtes) Platinen-Design Programm entwickelt. Dabei können unterschiedliche Hardwarekomponenten, beispielsweise Kondensatoren oder Widerstände, auf einer Platine platziert und mittels Leiterbahnen verbunden werden.

### HardwareComponent

Die Klasse *HardwareComponent* ist eine **abstrakte** Superklasse konkreter Hardwarekomponenten.

Eine Hardwarekomponente hat eine Bezeichnung (Attribut: *id* vom Typ *String*) und einen Preis (Attribut: *price* vom Typ *float*). Diese privaten Attribute müssen über den Konstruktor gesetzt werden. Zusätzlich können sie über entsprechende Zugriffsmethoden (z.B.: *getPrice()/setPrice(float new\_price)*) manipuliert und abgefragt werden.

### Capacitor

Die Klasse *Capacitor* ist eine **Subklasse** von *HardwareComponent*. Diese konkrete Hardwarekomponente verfügt über ein zusätzliches privates Attribut *capacitorValue* (vom Typ *float*), welches den Wert des Kondensators abbildet. Dieser Wert kann sowohl über den Konstruktor als auch über eine entsprechende Zugriffsmethode gesetzt werden. Zusätzlich kann der Wert über eine Methode ausgelesen werden.

### Resistor

Die Klasse *Resistor* ist eine **Subklasse** von *HardwareComponent*. Diese konkrete Hardwarekomponente verfügt über ein zusätzliches privates Attribut *resistorValue* (vom Typ *float*), welches den Wert des Widerstands abbildet. Dieser Wert kann sowohl über den Konstruktor als auch über eine entsprechende Zugriffsmethode gesetzt werden. Zusätzlich kann der Wert über eine Methode ausgelesen werden.

### CircuitPath

Verbindungen zwischen Hardwarekomponenten werden über die Klasse *CircuitPath* abgebildet und repräsentiert eine Leiterbahn auf der Platine. Die Referenzen auf die verbundenen Hardwarekomponenten werden in den privaten Attributen *hwComponent1* und *hwComponent2*, jeweils vom Typ *HardwareComponent*, abgebildet. Diese Referenzen werden direkt bei der Erstellung im Konstruktor festgelegt und dürfen in weiterer Folge nicht mehr verändert werden. Über entsprechende Zugriffsmethoden können die verbundenen Hardwarekomponenten eines *CircuitPath* abgefragt werden.

### PCB

Die Klasse *PCB* repräsentiert eine Platine. Auf dieser Platine können Hardwarekomponenten und Leiterbahnen platziert werden. Die auf der Platine platzierten Hardwarekomponenten werden über das private Attribut *hwComponents* (vom Typ *Collection<HardwareComponent>*) und die vorhandenen Leiterbahnen über das private Attribut *connections* (vom Typ *Collection<CircuitPath>*) abgebildet.

Mittels der Methode *placeComponent(HardwareComponent hw)* kann eine Hardwarekomponente auf der PCB platziert werden. Die übergebene Hardwarekomponente wird in dieser Methode zu der Liste der Hardwarekomponenten (*hwComponents*) hinzugefügt.

Die Methode *connectComponents(HardwareComponent hw1, HardwareComponent hw2)* verbindet zwei bereits auf der Platine befindlichen Komponenten miteinander. Es muss daher überprüft werden, ob die übergebenen Hardwarekomponenten bereits in der Liste *hwComponents* vorhanden sind (**Hinweis:** Iteration über alle Elemente dieser Liste, *equals*-Methode). Sind beide Hardwarekomponenten vorhanden, wird eine Leiterbahn zwischen diesen erstellt und zu der Liste der Leiterbahnen (*connections*) hinzugefügt. In diesem Fall wird *TRUE* zurückgegeben. Befindet sich eine der beiden Komponenten nicht auf der Platine, so wird keine Verbindung erstellt und *FALSE* zurückgegeben.

Zusätzlich existiert eine Methode *addConnection(CircuitPath connection)* über die eine Leiterbahn direkt hinzugefügt werden kann. Hierbei werden sowohl die Leiterbahn als auch die von der Leiterbahn aus referenzierten Hardwarekomponenten zu der Platine hinzugefügt. Hierbei muss nicht beachtet werden, ob eine Komponente bereits auf der Platine platziert ist.

Der Gesamtpreis der Platine (= Summe der Preise aller darauf befindlichen Hardwarekomponenten) kann über die Methode *calculatePrice* ermittelt werden. Der ermittelte Gesamtpreis wird als *float* zurückgegeben.

Informationen zur Platine können mittels der Methode *showConnectionDetails* auf der Konsole ausgegeben werden. Dabei werden Informationen über alle auf der Platine befindlichen Leiterbahnen und deren dadurch verbundenen Hardwarekomponenten ausgegeben. Zusätzlich wird der Gesamtpreis der Platine ausgegeben.

Die Form der Ausgabe kann selbst bestimmt werden. Beispielsweise könnte eine Verbindung in der Form

*hwComponent1* < --- Connected --- > *hwComponent2*

ausgegeben werden.

### **Ausführung:**

Um Ihr Programm ausführen zu können, benötigen Sie einen Einstiegspunkt. In Java ist dies eine *main*-Methode.

Erstellen Sie eine Klasse *Configurator* und implementieren Sie eine *main*-Methode. Erstellen Sie in dieser *main*-Methode zumindest ein Objekt der Klasse *PCB* und platzieren Sie mindestens 8 Hardwarekomponenten auf dieser. Fügen Sie zusätzlich mindestens fünf Leiterbahnen der Platine hinzu. Geben Sie abschließend Informationen über die Platine aus und ermitteln Sie deren Gesamtpreis. Geben Sie den Gesamtpreis auf der Konsole aus.

Überprüfen Sie die Funktionalität Ihres Programms zusätzlich in eigenem Ermessen.