

## Aufgabe 01:

### eclipse kennenlernen, Wiederholung Array / Assoziation:

Die Installation und Verwendung von eclipse wurde ja kurz vorgezeigt, ich möchte an dieser Stelle aber auch die Seite [www.javabuch.de](http://www.javabuch.de) empfehlen. Man kann hier (leider nur bis zur 7. Ausgabe) ein sehr gutes kompaktes Buch gratis im HTML-Format herunterladen, das zu allen im heurigen Jahr vorkommenden Themen entsprechende Kapitel enthält. Für heute interessant: **Kapitel 3.2 Eclipse**

### Bitte beachten:

Um ein Programm mittels **eclipse** zu starten, muss im Gegensatz zu **BlueJ** eine Start- oder Testklasse mit einer *main*-Methode existieren: `public static void main(String[] args) { ... }`

Ein Inspektor-Fenster wie bei BlueJ gibt es nicht mehr. Werte von Variablen können in eclipse zur Laufzeit im Debugging-Modus betrachtet werden (Breakpoint setzen, Debugging starten,...)

### **Übungsbeispiel:**

Für alle im Anschluss beschriebene Klassen gilt:

- Konstruktoren, set- und get-Methoden **soweit erforderlich** bzw. entsprechend der Aufgabe realisieren. Überprüfungen finden nicht direkt im Konstruktor statt, sondern es werden geeignete set-Methoden aufgerufen.
- Die set-Methoden führen Plausibilitätsüberprüfungen durch und verwenden im Fehlerfall (voerst noch) einen sinnvoll gewählten Default-Wert.
- Alle Klassen und Methoden sind **Java-Doc**-tauglich **kommentiert**.
- Alle Methoden werden **nachweislich getestet**. D.h. es werden alle Methoden in der *main()*-Methode aufgerufen und auch ungültige Werte für die Parameter übergeben. Natürlich können auch mehrere Testmethoden verwendet und in der *main()* aufgerufen werden. Nach erfolgreichem Test können einzelne Aufrufe gerne auskommentiert werden, aber bitte nicht löschen!

Unbedingt auch mittels Debugger in eclipse den Inhalt von Attributen, Parameter und Arrays überprüfen, eine Suche nach einer freien Position verfolgen,... um auch dieses Hilfsmittel kennenzulernen.

### **Das fertige eclipse-Projekt ist abzugeben – inklusive sinnvoller Testklasse!**

Dazu:

- **lokal** einen Abgabeordner „**Aufgabe\_01**“ erstellen,
- aus eurem eclipse-Projekt den Quelltextunterordner („**src**“) in diesen lokalen Ordner kopieren,
- ihn **abschließend in euren Abgabebordner kopieren**

**Keinesfalls den ganzen Workspace kopieren bzw. abgeben!**

Zu implementieren sind die Klassen **Bild** und **Galerie**:

Galerie
name : String bilder : Bild [] ...
setName(name: String) aufhaengen(bild: Bild) : boolean berechneGesamtVerkaufswert(): float zaehleTeurer(wert: float) : int entferneBilderKuenstler (kuenstler: String): int  toString(): String

Bild
kuenstler : String titel : String ekPreis : float selten : boolean
set... : get... : berechneVerkaufswert(): float  toString(): String print(): void

Hinweise zu den Methoden:

#### Bild:

- Konstruktor: übernimmt den Namen des Künstlers, den Titel und den Einkaufs-Preis des Bildes
- getter/setter alle
- setKuenstler soll so implementiert werden, dass der Künstler nur ein einziges Mal gesetzt und nachher nie mehr verändert werden kann!
- berechneVerkaufswert Ist das Bild selten, wird als Verkaufswert der Einkaufspreis verdoppelt , sonst mit 1.25 multipliziert.
- toString gibt mit kurzem, erklärendem Text die Information über das Bild zurück (inklusive Verkaufswert, Text, ob das Bild "**selten**" ist oder "**nicht selten**"!)

#### Galerie:

- Konstruktor: übernimmt den Namen der Galerie
- aufhaengen(bild) "übernimmt" ein Bild in die Galerie, aber nur, wenn sein Verkaufswert über 100.000,- liegt.
- berechneGesamtVerkaufswert() liefert den Gesamtverkaufswert aller in der Galerie befindlichen Bilder zurück.
- zaehleTeurer(wert) liefert die Anzahl aller Bilder zurück, deren Verkaufswert höher als der übergebene Wert ist.
- entferneBilderKuenstler(kuenstler) "entfernt" alle Bilder des übergebenen Künstlers aus der Galerie und gibt zurück, wieviele Bilder entfernt wurden.
- toString() liefert einen Text zurück, der eine "Überschrift" mit dem Namen der Galerie sowie die Informationen aller vorhandenen Bilder enthält.
- print() gibt einen Text auf die Konsole aus, der eine "Überschrift" mit dem Namen der Galerie sowie die Informationen aller vorhandenen Bilder enthält.