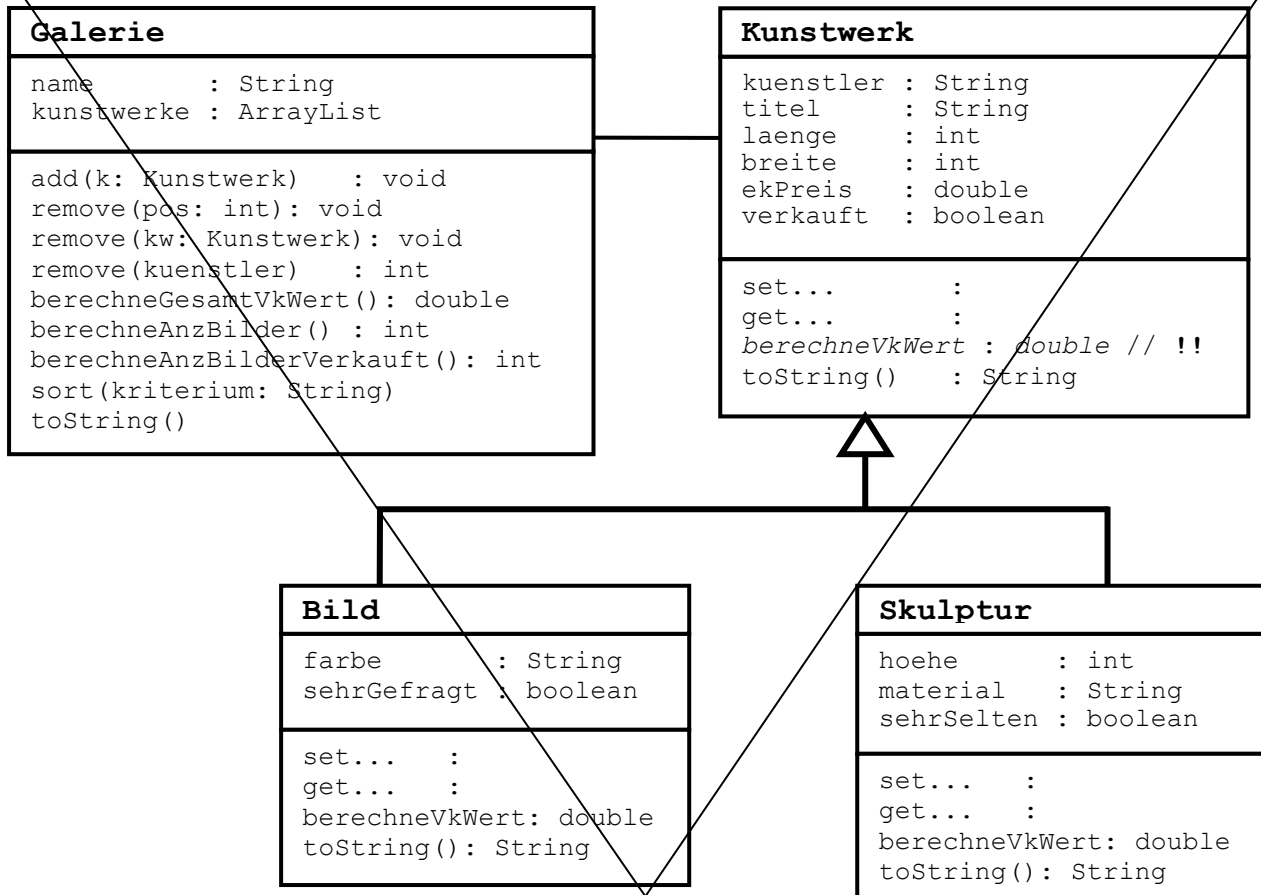


Implementieren Sie die im nachfolgenden Diagramm beschriebenen Klassen:

EX01



Die get- und set-Methoden sind im Diagramm nur angedeutet; sie sind jedoch obligatorisch!!

Das Testen der Klassen hat so zu erfolgen, daß sowohl beim Instanzieren der Klassen als auch beim Aufruf diverser Methoden mit Parametern die korrekte Behandlung sämtlicher erdenklicher Fehler überprüft werden muß!! Sie müssen daher für das Testen einen zeitlichen Aufwand von ca. 25-30 % der gesamten Implementierungszeit veranschlagen!

Die Behandlung von Fehlern hat ausnahmslos mit Hilfe von Exceptions zu erfolgen.

Aufgabe0:

Erstellen Sie eine Klasse **GalerieException**, mit deren Hilfe sämtliche Problem-Situationen behandelt werden können.

Aufgabe1:

Kunstwerk Konstruktor: übernimmt Parameter für Künstler, Titel u. Einkaufspreis

getter/setter: alle

Hinweise: 1.) der Künstler darf im Nachhinein nicht mehr veränderbar sein!!)

2.) da die (plausible) Länge und Breite eines **Bildes** sich von jenen einer **Skulptur** unterscheiden, müssen die Plausibilitätsprüfungen in der jeweiligen Klasse durchgeführt werden.

berechneVkWert(): wird hier nicht implementiert; es muß aber sichergestellt werden, daß ableitende Klassen diese Methode implementieren.

toString(): liefert Information über Künstler, Titel, Wert und Verfügbarkeit (fixer Text ist grau unterlegt!!).

Form: kuenstler " - " titel " - " berechneWert " - " text_ob_verkauft

Beispiele: Schiele - Krumau - 1200000 - verfügbar
Brunner - Horizonte - 1100 - verkauft

Aufgabe2:

Bild

Konstruktor: übernimmt Parameter für Länge und Breite
getter/setter: alle (max. Länge: 500 cm, Breite: 300)
berechneVkWert(): ist das Bild z.Zt. sehr gefragt, ergibt sich der Wert aus Einkaufs-Preis + 50%, sonst + 25%.
toString(): liefert zusätzlich Info über die Maße des Bildes (Form s.u.)

Beispiele: Schiele - Krumau - 1200000 - verfügbar - Maße: 120x95
Brunner - Horizonte - 1100 - verkauft - Maße: 180x90

Aufgabe2b: **TestBild** zum ausführlichen Testen der Klasse und ihrer Methoden

Aufgabe3:

Skulptur

Konstruktor: übernimmt Parameter für Länge, Breite, Höhe und Material
getter/setter: alle (max. Länge: 200, Breite: 200, Höhe: 1000)
berechneVkWert(): sind Skulpturen dieses Künstler selten, wird der Einkaufs-Preis verdoppelt; andernfalls + 50% .
toString(): liefert zus. Info über Material (Form siehe unten)

Beispiele: Laura Browne - Your Face in my Hand - 1500.0 - verfügbar - Material: Ton
Michelangelo - David - 15000000.0 - verfügbar - Material: Marmor

Aufgabe3b: **TestSkulptur** zum ausführlichen Testen

Aufgabe4:

Galerie

Konstruktor: übernimmt d. Namen und instanziert die ArrayList für mindestens 100 Kunstwerke.
getter/setter: alle
add(kw) fügt ein Kunstwerk d. Galerie hinzu.
remove(pos) entfernt das Kunstwerk an der Stelle **pos** aus der Galerie.
remove(kw) entfernt das angegebene Kunstwerk aus der Galerie.
remove(kuenstler) entfernt alle Kunstwerke dieses Künstlers.
(Der Erfolg dieser 3 Methoden ist zu kontrollieren!!!)
berechneGesamtVkWert() liefert den Gesamtwert aller Kunstwerke der Galerie.
berechneAnzBilder() liefert die Gesamtanzahl aller Bilder der Galerie.
berechneAnzVerkauft() liefert die Anzahl aller verkauften Bilder.
sort(kriterium) Sortiert die Kunstwerke nach dem übergebenen Kriterium (z.Zt. nur "Künstler" oder "Wert").
toString() Form siehe unten (verpflichtend!):

```
Galerie NewArts
-----
Waldmüller - Mädchen mit Kuh - 12500.0 - verkauft - Maße: 100x130
Auguste Rodin - Wasserspiel - 150000.0 - verfügbar - Material: Eisen u. Stahl
Schiele - Krumau - 125000.0 - verfügbar - Maße: 80x100
Michelangelo - David - 15000000.0 - verfügbar - Material: Marmor
Schiele - Frau - 125000.0 - verfügbar - Maße: 80x90
Bruno Gironcoli - Weltuhr - 1500000.0 - verfügbar - Material: Stahl u. Messing
Laura Browne - Your Face in my Hand - 1500.0 - verfügbar - Material: Ton
```

Aufgabe4b: **TestGalerie** zum ausführlichen Testen

Datei-Handling:

%

FILES:

Aufgabe5:

Erstellen Sie zur Klasse **Galerie** eine Methode **saveKunstwerke**, mit deren Hilfe alle Informationen über die Kunstwerke serialisiert in eine Datei beliebigen Namens (z.B. "D:\scratch\kunstwerke.ser") geschrieben werden.

Aufgabe5b: \ **TestSave** zum Testen

Aufgabe 6:

Erstellen Sie zur Klasse **Galerie** eine Methode **loadKunstwerke**, mit deren Hilfe Informationen für Kunstwerke aus einer Datei beliebigen Namens gelesen werden. Diese sollen in die Collection importiert werden.

Aufgabe 6b: ~~TestLoad~~ zum Testen

Aufgabe7:

Erstellen Sie zur Klasse **Galerie** eine Methode **exportKunstwerke**, mit deren Hilfe alle Informationen über die Kunstwerke in eine Datei namens "Kunstwerke.dat" geschrieben werden. Die Form der Ausgabe und die Anordnung der Daten (Fix-Length!) ist dabei so zu wählen, daß diese Daten über die Methode **importKunstwerke** wieder eingelesen und weiterverarbeitet werden können.

Aufbau einer Zeile (siehe auch unten *Hilfs-Raster*):

| | Stellen | |
|------------------------|---------|----------------------|
| Kunstwerk-Art: | 0 - 9 | |
| Künstler: | 10 - 29 | |
| Titel: | 30 - 49 | |
| Wert: | 50 - 59 | |
| Länge: | 60 - 65 | |
| Breite: | 66 - 71 | |
| sehrGefragt/sehrSelten | 73 | (j n) |
| Höhe: | 75 - 79 | (nur bei Skulptur) |
| Material: | 80 - .. | (nur bei Skulptur) |

Beispiele:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------------------|-------------|----------------------|-------------|-------------|-------------|-------------|-------------|----------------------|
| | 01234567890 | 12345678901 | 23456789012 | 34567890123 | 45678901234 | 56789012345 | 67890123456 | 78901234567 | 890123456789 |
| Bild | Waldmüller | | Mädchen mit Kuh | | 10000 | 100 | 130 | j | |
| Skulptur | Auguste Rodin | | Wasserspiel | | 100000 | 5000 | 5000 | n | 2000 Eisen u. Stahl |
| Bild | Schiele | | Krumau | | 100000 | 80 | 100 | j | |
| Skulptur | Michelangelo | | David | | 1000000 | 100 | 100 | n | 350 Marmor |
| Bild | Schiele | | Frau | | 100000 | 80 | 90 | j | |
| Skulptur | Bruno Gironcoli | | Weltuhr | | 1000000 | 500 | 700 | n | 350 Stahl u. Messing |
| Skulptur | Laura Browne | | Your Face in my Hand | | 1000 | 10 | 30 | j | 10 Ton |
| Bild | Brunner | | Horizonte | | 1000 | 130 | 70 | n | |
| Skulptur | Alberto Giacometti | | Der Wagen | | 10000 | 120 | 60 | j | 90 Bronze |
| Bild | Klimt | | Judith | | 50000 | 70 | 80 | j | |
| Skulptur | Auguste Rodin | | Bürger von Calais | | 170000 | 500 | 100 | n | 250 Bronze |
| Bild | Frank | | Tango | | 500 | 60 | 80 | n | |

Aufgabe 7b: **TestExport**

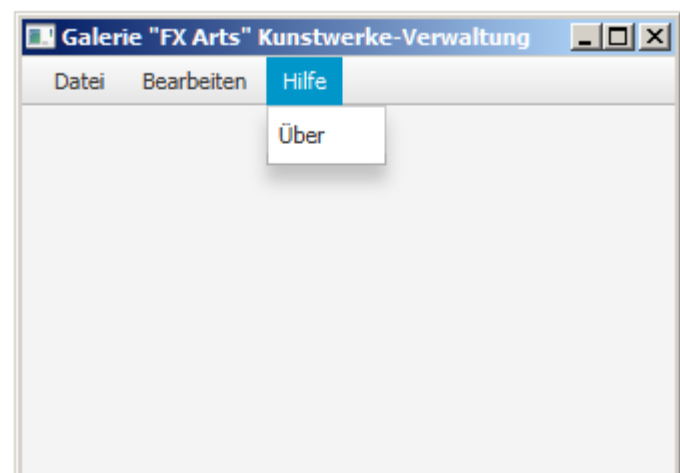
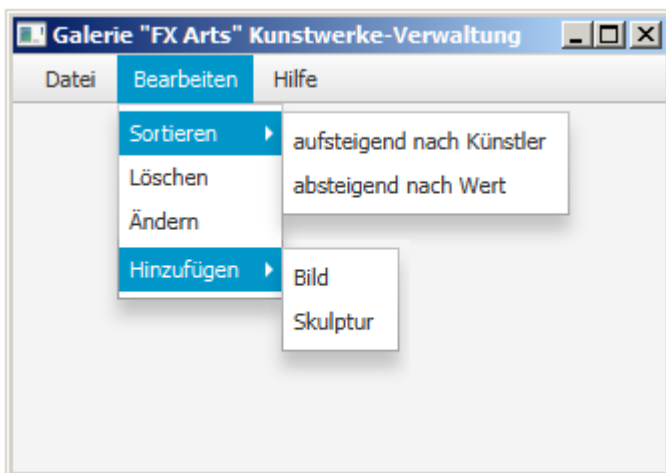
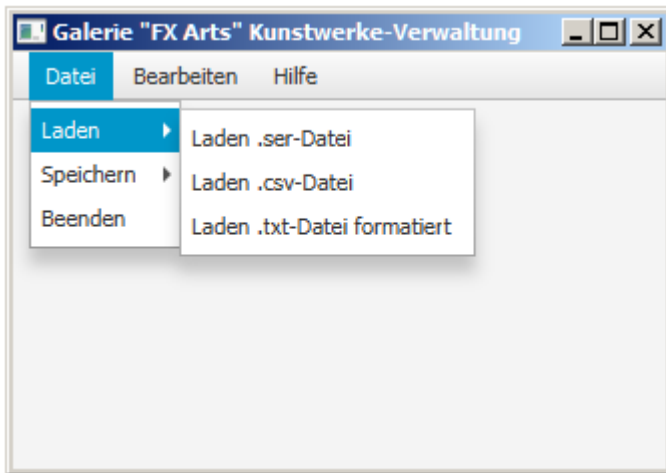
Aufgabe8:

Erstellen Sie zur Klasse **Galerie** eine Methode **importKunstwerke**. Mit deren Hilfe sollen Daten über Kunstwerke aus einer Datei namens "Kunstwerke.dat" gelesen werden; diese sollen in die "Sammlung" aufgenommen werden.

Aufgabe8b: / TestImport

Aufgabe 9:

Erstellen Sie für die Verwaltung der Kunstwerke der Galerie eine graphische Oberfläche (Größe: 500*200 Pixel), die folgendes Aussehen sowie Menü-Struktur aufweist (jedoch noch ohne jegliche Funktionalitäten!!):



Jene Menü-Einträge, die beim Neu-Erscheinen der GUI nicht sinnvoll ausgewählt werden können (wie z.B. "Speichern", "Ändern" usw.), sollen hierbei zwar angezeigt werden, vorerst aber nicht aktivierbar sein:

