Digitale Praktische Abschlussprüfung

Fachbereich gem. APO BK C1: Wirtschaft und Verwaltung
Bildungsgang gem. APO-BK nach Anlage: Kaufmännische Assistenten (m, w, d)
Fachbereich: Betriebswirtschaftslehre
Fachlicher Schwerpunkt: Informationsverarbeitung

Situation

Die Künstleragentur „Berühmt und beschäftigt“ vermittelt Auftritte an verschiedenen Veranstaltungsorten an die bei ihr unter Vertrag stehenden Künstlerinnen und Künstlern. Diese gehören verschiedenen Kategorien (Sänger, Musiker, Schauspieler, …) an. Die Verwaltung über eine Karteikartensammlung ist aufgrund der Vielzahl an Künstlerinnen und Künstlern und deren Auftritten an ihre Grenzen gestoßen. Als kaufmännischer Assistent in der Agentur werden Sie damit beauftragt, die Künstlerverwaltung im Rahmen eines Agenturprojekts zu digitalisieren. In Frage kommen ein Java-Programm und die Realisation der Verwaltung im Rahmen einer Datenbank. Einige Vorarbeiten wurden bereits geleistet. Es gibt bereits ein unvollständiges Java-Projekt und eine Tabellenstruktur. (Dateien im Prüfungsordner)

A1 Geschäftsprozessmodellierung

Aufgabe: eEPK-Diagramm erstellen (20 Punkte)

Erstellen Sie aufgrund der folgenden Beschreibung ein erweitertes, ereignisgesteuertes Prozesskettendiagramm zum Prozess „Veranstalteranfrage beantworten“.

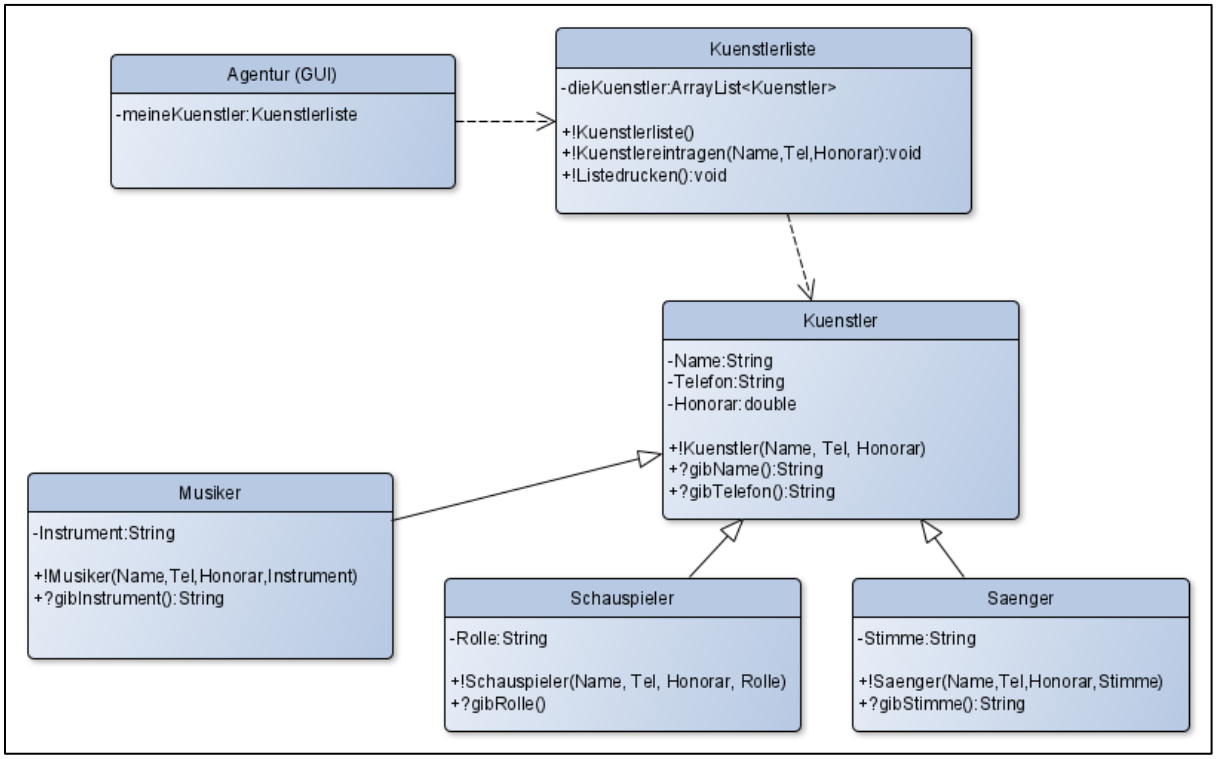
Beschreibung:

Normalerweise melden sich Veranstalter (Konzertveranstalter, Filmstudios, …) bei der Künstleragentur telefonisch. Die Anforderungen des Veranstalters werden durch das Agenturbüro notiert. Anschließend sichtet das Agenturbüro die „Künstlerkartei“ und stellt eine Liste mit allen in Frage kommenden Künstlerinnen und Künstlern auf. Danach prüft die Agenturleitung die Kalender der gelisteten Künstlerinnen und Künstler. Diejenigen, die zum angefragten Zeitraum verfügbar sind, werden in eine „Telefonliste“ aufgenommen und von der Agenturleitung kontaktiert. In diesem Telefonat wird geklärt, ob ein Interesse an der Veranstaltung/dem Auftritt besteht. Nur Künstlerinnen und Künstler, die interessiert sind, kommen auf eine „Vorschlagsliste“. Diese Vorschlagsliste wird dem Veranstalter durch das Agenturbüro zur Verfügung gestellt und eine Kopie der Liste im Agenturbüro abgeheftet. Danach ist der Prozess abgeschlossen.

A2 Objektorientierte Programmierung

1. Aufgabe: UML-Diagramm erläutern (15 Punkte)

Für das Agenturprojekt liegt das u.a. verkürzte Klassendiagramm vor. Erläutern Sie das abgebildete Klassendiagramm (Attribute, Methoden, Beziehungen) ausführlich.



2. Aufgabe Quelltext vervollständigen/ergänzen (30 Punkte)

Ihnen liegt ein unvollständiges BlueJ-Projekt „KuenstlerAgentur“ vor. Die Abbildung 1 zeigt das BlueJ-Projektfenster und die in Teilen bereits fertige grafische Benutzeroberfläche.

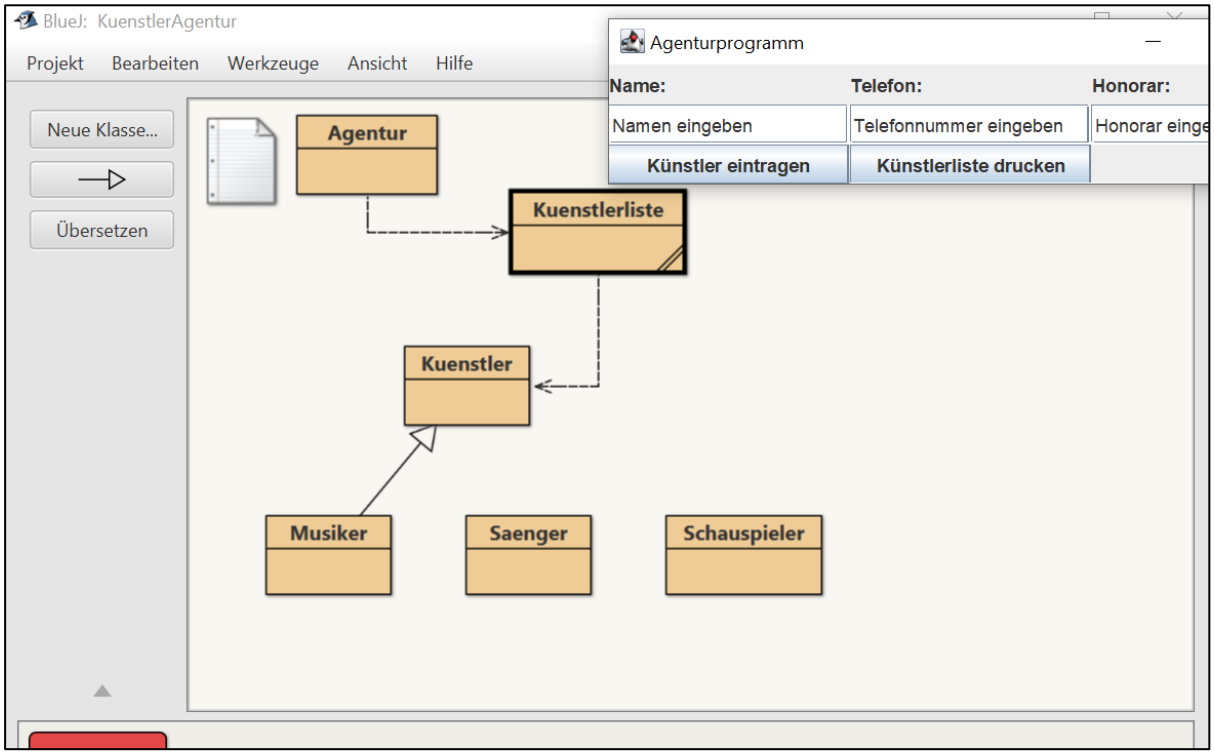


Abbildung 1: Das unvollständige Projekt

Vervollständigen Sie das Projekt „KünstlerAgentur“ um die im Folgenden angegebenen Funktionalitäten/Anweisungen. Orientieren Sie sich dabei an der Abbildung 2 unterhalb der Aufgaben. Die Abbildung 2 zeigt das Programm, nachdem ein Künstler erfasst und die Künstlerliste gedruckt wurde.

a) Erweitern Sie die Klasse „Kuenstler“ um eine Methode(Anfrage) „gibAgenturanteil()“. Die Agentur erhält 10 % des Honorars für die erfolgreiche Vermittlung eines Künstlers. Die Methode soll diesen Betrag ermitteln. (3 Punkte)

b) Erstellen Sie die Beziehungen zwischen der Oberklasse „Künstler“ und den Unterklassen „Sänger“ und „Schauspieler“. Die speziellen Attribute sollen über den Konstruktor mit Werten belegt werden (siehe Klassendiagramm). (6 Punkte)

c) Erstellen Sie den Quelltext für die Methode „Listedrucken()“ in der Klasse „Kuenstlerliste“. Die Methode soll die Daten der gespeicherten Künstler (Name, Telefonnummer und Honorar) auf dem Bildschirm/der Konsole ausgeben. (6 Punkte)

d) Realisieren Sie die Ereignisverarbeitung für die beiden Schaltflächen in der Klasse „Agentur“

   a. Ein Klicken auf den Knopf „Künstler eintragen“ soll dazu führen, dass ein neuer Künstler mit den Daten aus den Textfeldern (Name, Telefon, Honorar) der Künstlerliste hinzugefügt wird. (6 Punkte)

   b. Ein Klicken auf den Knopf „Liste drucken“ soll die aktuell gespeicherten Künstler auf der Konsole/dem Bildschirm ausgeben.(3 Punkte)

e) Ergänzen Sie die Klasse „Künstlerliste“ um drei Einzelanweisungen. Es sollen ein Musiker, ein Schauspieler und ein Sänger der Liste hinzugefügt werden. (6 Punkte)

Musiker:	Bach, 0231-4711, 1500, Orgel,
Sänger:	Caruso, 0231-0815, 750,Tenor,
Schauspieler:	Arnold, 0231-911, 2000, Actionheld

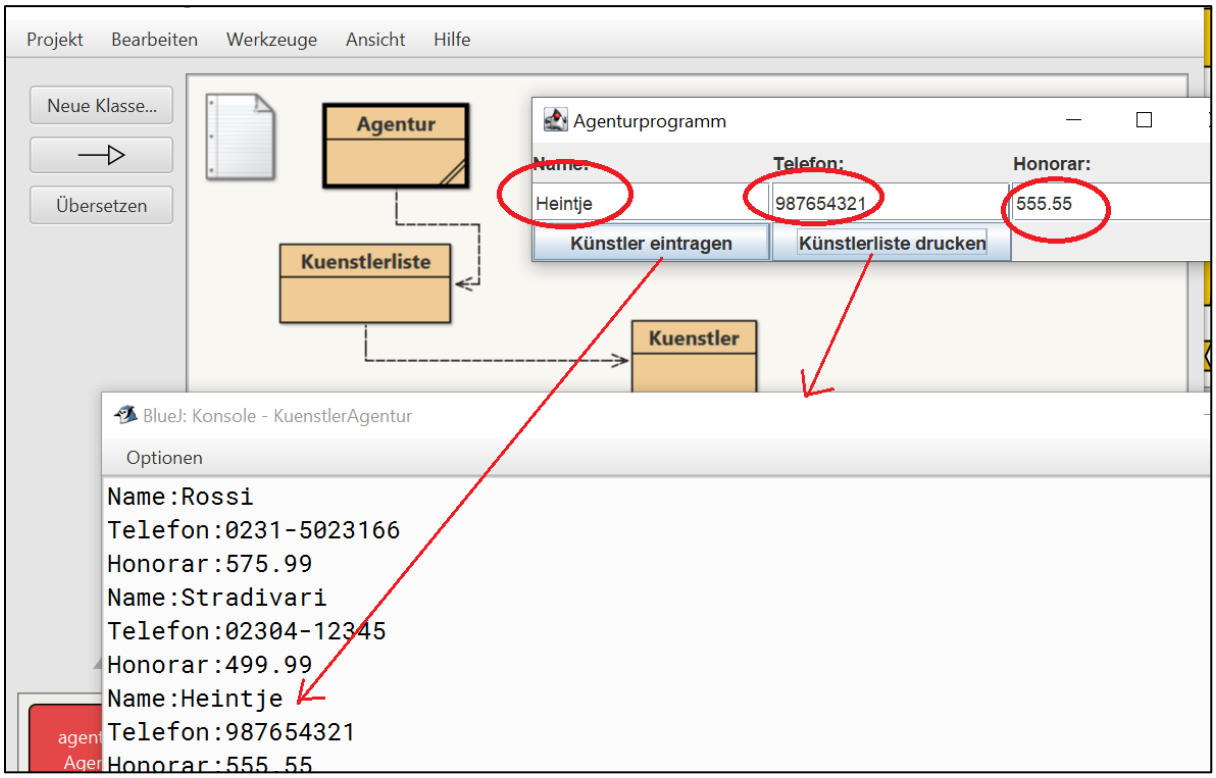


Abbildung 2: Das vervollständigte Projekt nach dem Drucken der Künstlerliste

A3 Relationale Datenbanken

1. Aufgabenstellung SQL-Statements (15 Punkte)

Die Künstleragentur hat für das Softwareprojekt die Engagements ihrer Künstler in einer Tabelle „Auftritte“ zusammengestellt.

Ausschnitt aus der Tabelle „Auftritte“ (liegt als XLS-Tabelle vor):

	A	B	C	D	E	F	G
1	Name	Telefon	Kategorie	Tageshonorar	EngagiertVon	EngagiertBis	Engagiert bei
2	Caruso, Celine	0231-0815	Sänger	750	02.03.2022	05.03.2022	Opernhaus Dortmund
3	Bach, Bertram	0231-4711	Musiker	1500	15.02.2022	16.02.2022	Erzbistum Köln
4	Caruso, Celine	0231-0815	Sänger	750	13.04.2022	20.04.2022	Opernhaus Dortmund
5	Arnold Armstrong	0231-911	Schauspieler	2000	15.01.2022	15.02.2022	Rakete Filmstudios
6	Arnold Armstrong	0231-911	Schauspieler	2000	04.03.2022	28.03.2022	Rakete Filmstudios
7	Amati, Anna	0231-112103	Musiker	2500	15.04.2022	15.05.2022	Scala Mailand

Abbildung 3: Die Tabelle „Auftritte“

Erläutern/Übersetzen Sie die folgenden SQL-Statements unter der Annahme, dass sich die o. a. Tabelle „Auftritte“ in der Datenbank „Agentur“ befindet.
Stellen Sie das Ergebnis für jedes Statement (sofern möglich) als Tabelle dar.

a) SELECT Name, Telefon FROM Auftritte WHERE Kategorie = Schauspieler ORDER BY Name DESC (3 Punkte)

b) SELECT Kategorie, AVG(Tageshonorar) AS Durchschnitt FROM Auftritte GROUP BY Kategorie (3 Punkte)

c) SELECT Name, ([EngagiertBis]-[EngagiertVon]+1)*[Tageshonorar]*0.1 AS Agenturanteil FROM Auftritte (3 Punkte)

d) SELECT Kategorie, COUNT (DISTINCT Name) FROM Auftritte (4 Punkte)

e) DELETE * FROM Auftritte (1 Punkt)

f) DROP DATABASE Agentur (1 Punkt)

2. Aufgabenstellung Normalisierung / ER-Modell (24 Punkte)

Die Daten der Tabelle „Auftritte“ sind nicht normalisiert. Nicht normalisierte Datenstrukturen führen zu Problemen bei der Datenverarbeitung, die durch den Prozess der Normalisierung vermieden werden können.

a) Erläutern Sie allgemein zwei Vorteile der Normalisierung von Datenstrukturen. (4 Punkte)

b) Beschreiben Sie an einem Beispiel, was man unter einer Löschanomalie versteht. (4 Punkte)

c) Normalisieren Sie die Daten der Tabelle „Auftritte“ und erstellen Sie ein vollständiges ER-Modell. (16 Punkte)

Unterrichtsvoraussetzungen
Übersicht über die Themen und Unterrichtsgegenstände der Halbjahre 12/I bis 13/II

Kurshalbjahr	Kursthemen/Unterthemen
12/I	**Anwendungsentwicklung** • Eigenschaften/Attribute und Methoden/Funktionen • Kontrollstrukturen (Verzweigung, Schleife/n) • Sammlungen **Grundlagen Relationale Datenbanken am Beispiel MS Access** • Datenstrukturen, Datenbanken, Tabellen • Attribute und Feldeigenschaften • Entity-Relationship-Modell, Schlüsselattribute • Normalisierung von Tabellen • SQL-Syntax
12/II	**Effiziente Erstellung und Nutzung relationaler Datenbanken mit XAMPP** • Generierung von Tabellen • Import/Export von SQL-Dateien • SQL-Auswahl- und Aktionsabfragen • Datenbankmanagement mit MySQL **Anwendungsentwicklung** • Wiederholung Kontrollstrukturen (Logische Operatoren, Mehrfachverzweigungen, Schleifen, Arrays, Funktionen) • Grafische Benutzeroberflächen • Vererbung
13/I	**Anwendungsentwicklung unter Einbezug von Datenbanken** **Geschäftsprozessmodelle als Bestandteil von Pflichtenheften** • Analyse, Darstellung und Gestaltung von Geschäftsprozessen • Abbildung ereignisgesteuerter Prozessketten **ER-Modelle als Bestandteil von Pflichtenheften** • Analyse, Darstellung und Gestaltung von ER-Modellen aufgrund von Interview, Selbstaufschreibung oder Dokumentenanalyse • Anlegen von Datenstrukturen auf einem MySQL-Server • SQL-Abfragesyntax **Anwendungsentwicklung** • Wiederholung Kontrollstrukturen und grafische Benutzeroberflächen • Wiederholung Vererbung • Verbindungsaufbau zum MySQL-Server • Abfangen von Ausnahmen • Einbettung von SQL-Syntax in Programme • Auslesen und Darstellen von SQL-Abfrageergebnissen in grafischen Benutzeroberflächen
13/II	**Prüfungsvorbereitung**

Konkrete unterrichtliche Voraussetzungen für die Bearbeitung des Vorschlags/der Aufgabe(n)

Die Aufgabenstellung zum Thema Geschäftsprozesse bildet die Modellierung eines Geschäftsprozesses, das spiralcurricular im Unterricht behandelt wird, ab, einschließlich der Unterscheidung von Kern- und Serviceprozessen. Prozesskettendiagramme verknüpfen betriebswirtschaftliche Aspekte aus dem Bereich der Ablauforganisation mit informationstechnischen Aspekten der Wirtschaftsinformatik/Anwendungsentwicklung, um betriebliche Prozesse it-technisch zu unterstützen. Die Analyse und Modellierung von Geschäftsprozessen wird im Rahmen des Phasenmodells der Anwendungsentwicklung in Form von eEPKs in der Phase Analyse als Dokumentationswerkzeug eingesetzt. Aus eEPKS lassen sich z.B. DV-Funktionen und notwendige Bildschirmmasken/Formulare ableiten, die im Rahmen des modellierten Geschäftsprozesses von den Anwendern benötigt werden, weshalb sie auch Bestandteil eines Pflichtenhefts sein können.

Die meisten kaufmännischen Anwendungen basieren auf einer relationalen Datenbank. Die Schüler haben im Unterricht die Handhabung der Datenbankmanagementsysteme MS-Access und MySQL-Datenbanken kennengelernt und darauf basierend kleinere Datenbankprojekte erstellt. Der Einsatz von Datenbanken erfordert eine angemessene Definition von Datenstrukturen, um Daten redundanzfrei abzulegen. Insofern ist der Prozess der Normalisierung von Daten Standard bei der Definition von Datenstrukturen (Datenbank und Datenbanktabellen). Die Standardabfragesprache SQL wurde auf Basis eines XAMPP-Systems mit einem MySQL-Server thematisiert und praktisch angewendet.

Üblicherweise arbeiten Benutzer nicht direkt an einem Datenbankserver, bzw. direkt in Datenbanktabellen. Üblicherweise ist eine Benutzeroberfläche/Anwenderschnittstelle zwischengeschaltet, um die zu bearbeitenden Daten in einer benutzerorientierten Form darzustellen. In MS-Access gibt es zu diesem Zweck sogenannte Formulare.

MySQL bietet eine solche Sichtenfunktion nicht. In der Praxis werden entsprechende Benutzeroberflächen entweder programmiert oder man benutzt z.B. die Formularfunktion von MS-Access, in dem man eine lokale Datenbank mit den Tabellen auf dem MySQL-Server verknüpft und lokale Formulare für die Benutzer erstellt.

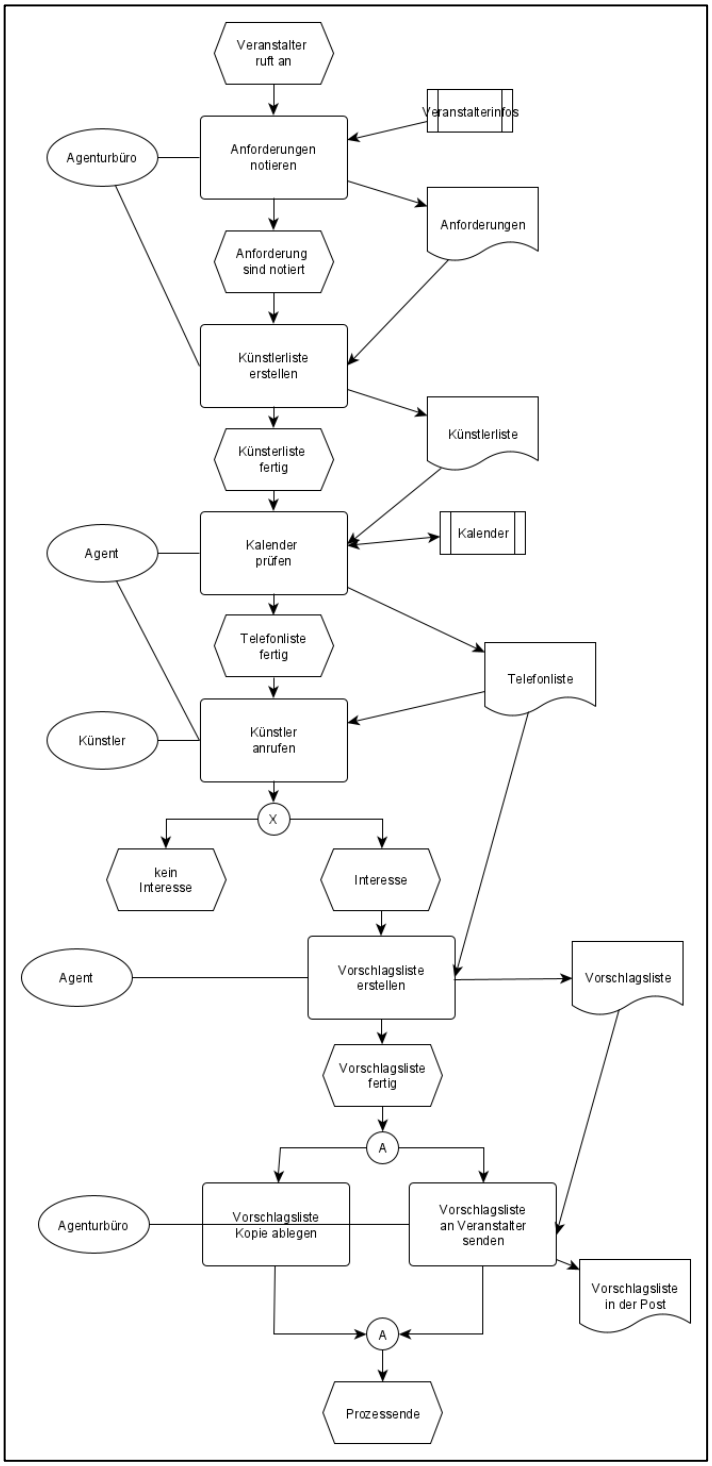
Java ist eine weit verbreitete Programmiersprache und eine Standardprogrammiersprache in den entsprechenden Studiengängen an Fachhochschulen. Insofern erscheint es sinnvoll, im Unterricht Grundlagen der Java-Programmierung, einschließlich der Möglichkeit, mit einem Java-Programm auch Datenbanken verwalten zu können, zu thematisieren. In der Jahrgangsstufe 13.1 wurde in kleineren Projekten der Zugriff auf eine Datenbank und die einfache Abfrage von Datenbankdaten mit Hilfe eines Java-Programms angewendet. Java Grundlagen waren Gegenstand des Unterrichts in den Jahrgangsstufen 12.1 und 12.2.

Erwartungshorizont

A1 Geschäftsprozessmodellierung

Aufgabe eEPK-Diagramm erstellen (20 Punkte)
Softwaretool: z. B. yED, grafischer Editor (OpenSource) (6.2.3 Fachbereichsspezifische Software einsetzen), *(6.3.3 Prozessdarstellung)*

Erstellen Sie aufgrund der folgenden Beschreibung ein Prozesskettendiagramm zum Prozess „Veranstalteranfrage beantworten“ *(6.2.4 Prozesse visualisieren)*.



Normalerweise melden sich Veranstalter (Konzertveranstalter, Filmstudios, …) bei der Künstleragentur telefonisch.

Die Anforderungen des Veranstalters werden durch das Agenturbüro notiert.

Anschließend sichtet das Agenturbüro die „Künstlerkartei“ und stellt eine Liste mit allen in Frage kommenden Künstlerinnen und Künstlern auf.

Danach prüft die Agenturleitung die Kalender der gelisteten Künstlerinnen und Künstler.

Diejenigen, die zum angefragten Zeitraum verfügbar sind, werden in eine „Telefonliste“ aufgenommen und von der Agenturleitung kontaktiert. In diesem Telefonat wird geklärt, ob ein Interesse an der Veranstaltung/dem Auftritt besteht.

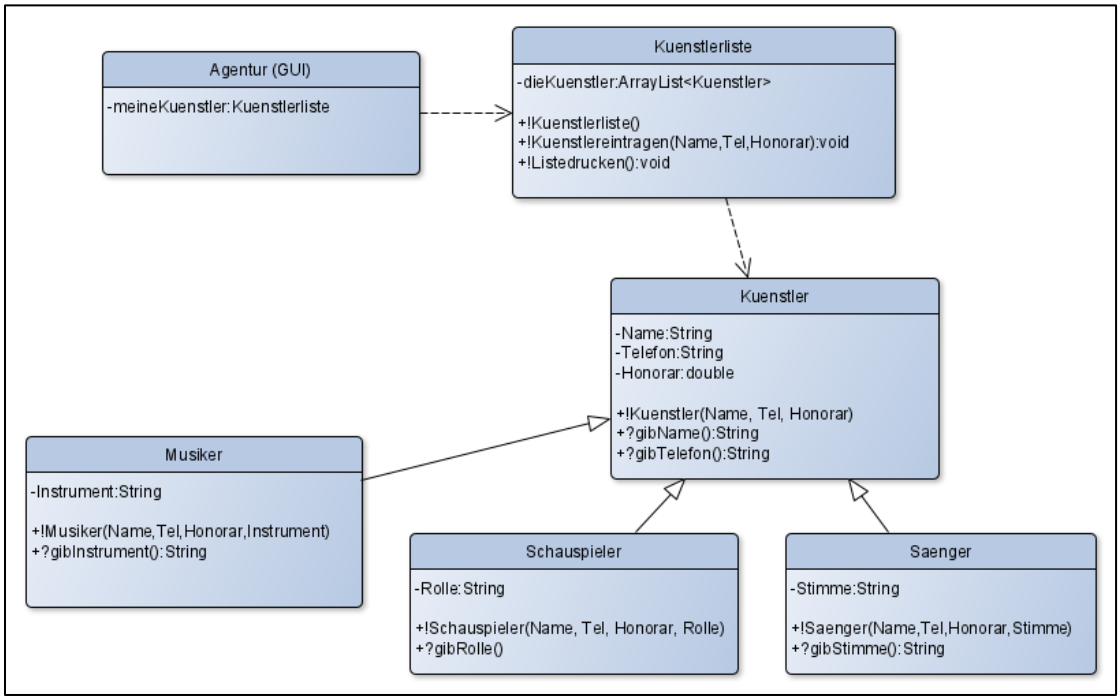
Nur Künstler, die interessiert sind, kommen auf eine „Vorschlagsliste“.

Diese Vorschlagsliste wird dem Veranstalter durch das Agenturbüro zur Verfügung gestellt und eine Kopie der Liste im Agenturbüro abgeheftet.

Danach ist der Prozess abgeschlossen.

A2 Objektorientierte Programmierung

Aufgabe UML-Diagramm erläutern *(6.3.3 Prozessdarstellung)* **(15 Punkte)**
Softwaretool: z. B. ArgoUML → Diagrammdatei kommentieren, alternativ Textverarbeitung *(6.2.3 Fachbereichsspezifische Software einsetzen)*
Für das Agenturprojekt liegt das untenstehende verkürzte Klassendiagramm vor. Erläutern Sie das abgebildete Klassendiagramm (Attribute, Methoden, Beziehungen) ausführlich *(4.2.2 Daten aufbereiten, strukturieren, dokumentieren und **interpretieren**)*. *(6.1.1 Digitale Repräsentation von Informationen und Daten in automatisierten Prozessen bewerten)*



- *Das Diagramm zeigt insgesamt 6 Klassen. Die Klasse Agentur verfügt über ein privates Attribut „meineKuenstler“ vom Typ Künstlerliste. Dadurch ergibt sich eine Verwendungsbeziehung zur Klasse Künstlerliste.*
- *Die Klasse Künstlerliste hat ein privates Attribut „dieKuenstler“ vom Typ ArrayList. Die ArrayList sammelt Objekte vom Typ „Kuenstler“. Daraus folgt die Verwendungsbeziehung zur Klasse „Kuenstler“.*
- *Die Klasse „Kuenstler“ hat drei private Attribute: Name und Telefon vom Typ String sowie Honorar vom Typ double.*
- *Die Klassen Musiker, Schauspieler und Saenger erben von der Oberklasse Kuenstler, d. h. Musiker, Saenger und Schauspieler sind Kuenstler (IS A).*
- *Die Unterklassen haben jeweils ein spezielles Attribut (Musiker → Instrument, Typ String, Schauspieler → Rolle, Typ String, Saenger → Stimme, Typ String).*
- *Die Klasse Kuenstler verfügt über drei Methoden. Der Konstruktor erwartet drei Parameter (Name, Telefon, Honorar). Weiterhin verfügt die Klasse Kuenstler über zwei sondierende Methoden, gibName() und gibTelefon().*
- *Die Konstruktormethoden der Unterklassen Musiker, Schauspieler und Saenger erwarten dieselben Parameter wie der Konstruktor der Oberklasse, ergänzt um einen Parameter für das jeweilige spezielle Attribut.*
- *Die Klasse Kuenstlerliste hat einen Konstruktor und zwei Methoden vom Typ Auftrag (void).*
- *Die Methode Kuenstlereintragen erwartet drei Parameter, die mit denen des Konstruktors der Klasse Kuenstler übereinstimmen. Zusätzlich gibt es eine Methode, um eine Kuenstlerliste zu drucken.*

Aufgabe Quelltext vervollständigen/ergänzen (30 Punkte)
Softwaretool: Java Programmierumgebung, z. B. BlueJ oder NetBeans …
(7.3.8 Programmierwerkzeuge)

Vervollständigen Sie das Projekt „KünstlerAgentur“ um die im Folgenden angegebenen Funktionalitäten/Anweisungen *(7.2.4 Programmiertools anwenden)*. Orientieren Sie sich dabei an der Abbildung 2 unterhalb der Aufgaben. Abbildung 2 zeigt das Programm, nachdem ein Künstler erfasst und die Künstlerliste gedruckt wurde.

a) Erweitern Sie die Klasse „Kuenstler“ um eine Methode(Anfrage) „gibAgenturanteil()“. Die Agentur erhält 10% des Honorars für die erfolgreiche Vermittlung eines Künstlers. Die Methode soll diesen Betrag ermitteln. *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (3 Punkte)

```
public double gibAgenturanteil()
   {
      return honorar*0.1;
   }
```

b) Erstellen Sie die Beziehungen zwischen der Oberklasse Künstler und den Unterklassen Sänger und Schauspieler. Die speziellen Attribute sollen über den Konstruktor mit Werten belegt werden (siehe Klassendiagramm). *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (6 Punkte)

```
public class Saenger extends Kuenstler
{
   public String Stimme;
   public Saenger(String pName, String pTel, double pHonorar, String pStimme)
   {
      super(pName, pTel, pHonorar);
      Stimme = pStimme;
   }
}
```

a) Erstellen Sie den Quelltext für die Methode „Listedrucken()“ in der Klasse „Kuenstlerliste. Die Methode soll die Daten der gespeicherten Künstler (Name, Telefonnummer und Honorar) auf dem Bildschirm/der Konsole ausgeben. *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (7.3.2 Verzweigungen,Bedingungen, Schleifen,Operatoren)
(6 Punkte)

```
public void Listedrucken()
   {
      int i =0;
      while(i < dieKuenstler.size())
      {
         System.out.println("Name:" + dieKuenstler.get(i).gibName());
         System.out.println("Telefon:" + dieKuenstler.get(i).gibTelefon());
         System.out.println("Honorar:" + dieKuenstler.get(i).gibHonorar());
         i++;
      }
   }
```

b) Realisieren Sie die Ereignisverarbeitung für die beiden Schaltflächen in der Klasse „Agentur“

a. Ein Klicken auf den Knopf „Künstler eintragen“ soll dazu führen, dass ein neuer Künstler mit den Daten aus den Textfeldern (Name, Telefon, Honorar) der Künstlerliste hinzugefügt wird. *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (6 Punkte)

```
public void actionPerformed(ActionEvent click)
    {
       String n = name.getText();
       String t = telefon.getText();
       double h = Double.parseDouble(honorar.getText());
       meineKuenstler.kuenstlerHinzufuegen(n,t,h);
    }
```

b. Ein Klicken auf den Knopf „Liste drucken“ soll die aktuell gespeicherten Künstler auf der Konsole/dem Bildschirm ausgeben. *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (3 Punkte)

```
public void actionPerformed(ActionEvent click)
    {
       meineKuenstler.Listedrucken();
    }
```

c) Ergänzen Sie die Klasse „Künstlerliste“ um drei Einzelanweisungen. Es sollen ein Musiker, ein Schauspieler und ein Sänger der Liste hinzugefügt werden. *(7.2.2. Algorithmen entwickeln) (7.2.4 Programmiertools anwenden)* (7.3.4 Programm, Methode, Klasse) (6P)

Musiker:	Bach, 0231-4711, 1500, Orgel
Sänger:	Caruso, 0231-0815, 750, Tenor
Schauspieler:	Arnold, 0231-911, 2000, Actionheld

```
dieKuenstler.add(new Musiker("Bach","0231-4711",1500,"Orgel"));
dieKuenstler.add(new Saenger("Caruso","0231-0815",750,"Tenor"));
dieKuenstler.add(new Schauspieler("Arnold","0231-911",2500,"Actionheld"));
```

A3 Relationale Datenbanken

1. Aufgabenstellung **SQL-Statements** *(4.3.3 Datenbankmanagementsysteme)* **(15 Punkte)**

Die Künstleragentur verwaltet die Engagements ihrer Künstler in einer Tabelle „Auftritte“.

Ausschnitt aus der Tabelle „Auftritte“ (liegt als XLS-Tabelle vor):

	A	B	C	D	E	F	G
1	Name	Telefon	Kategorie	Tageshonorar	EngagiertVon	EngagiertBis	Engagiert bei
2	Caruso, Celine	0231-0815	Sänger	750	02.03.2022	05.03.2022	Opernhaus Dortmund
3	Bach, Bertram	0231-4711	Musiker	1500	15.02.2022	16.02.2022	Erzbistum Köln
4	Caruso, Celine	0231-0815	Sänger	750	13.04.2022	20.04.2022	Opernhaus Dortmund
5	Arnold Armstrong	0231-911	Schauspieler	2000	15.01.2022	15.02.2022	Rakete Filmstudios
6	Arnold Armstrong	0231-911	Schauspieler	2000	04.03.2022	28.03.2022	Rakete Filmstudios
7	Amati, Anna	0231-112103	Musiker	2500	15.04.2022	15.05.2022	Scala Mailand

Abbildung 3: Die Tabelle „Auftritte“

Erläutern/Übersetzen Sie die folgenden SQL-Statements unter der Annahme, dass sich die o. a. Tabelle „Auftritte“ in der Datenbank „Agentur“ befindet.
Stellen Sie das Ergebnis für jedes Statement (sofern möglich) als Tabelle dar. *(4.2.2 Daten aufbereiten, strukturieren, analysieren, visualisieren, dokumentieren und **interpretieren**) (4.3.4 Datenanalyse und -auswertung)*

a) SELECT Name, Telefon FROM Auftritte WHERE Kategorie = Schauspieler ORDER BY Name DESC (3 Punkte)
Die Abfrage listet die Attribute Name und Telefon für alle Auftritte mit der Kategorie Schauspieler und sortiert die Datensätze nach Name absteigend.

b) SELECT Kategorie, AVG(Tageshonorar) AS Durchschnitt FROM Auftritte GROUP BY Kategorie (3 Punkte)
Die Abfrage listet die Kategorie und das durchschnittliche Tageshonorar je Kategorie. Für den Ausdruck mit der Aggregatfunktion wird ein Alias verwendet.

c) SELECT Name, ([EngagiertBis]-[EngagiertVon]+1)*[Tageshonorar]*0.1 AS Agenturanteil FROM Auftritte (3 Punkte)
Die Abfrage zeigt den Künstlernamen und je Künstlername in einem berechneten Ausdruck den Agenturanteil von 10 % für die Dauer des Engagements.

d) SELECT Kategorie, COUNT (DISTINCT Name) FROM Auftritte (4 Punkte)
Die Abfrage listet die Kategorie und die Anzahl der verschiedenen Künstlernamen in dieser Kategorie.

e) DELETE * FROM Auftritte (1 Punkt)
Die Abfrage löscht alle Datensätze aus der Tabelle Auftritte.

f) DROP DATABASE Agentur (1 Punkt)
Die Anweisung löscht die Datenbank Agentur.

2. Aufgabenstellung Normalisierung / ER-Modell (24 Punkte)

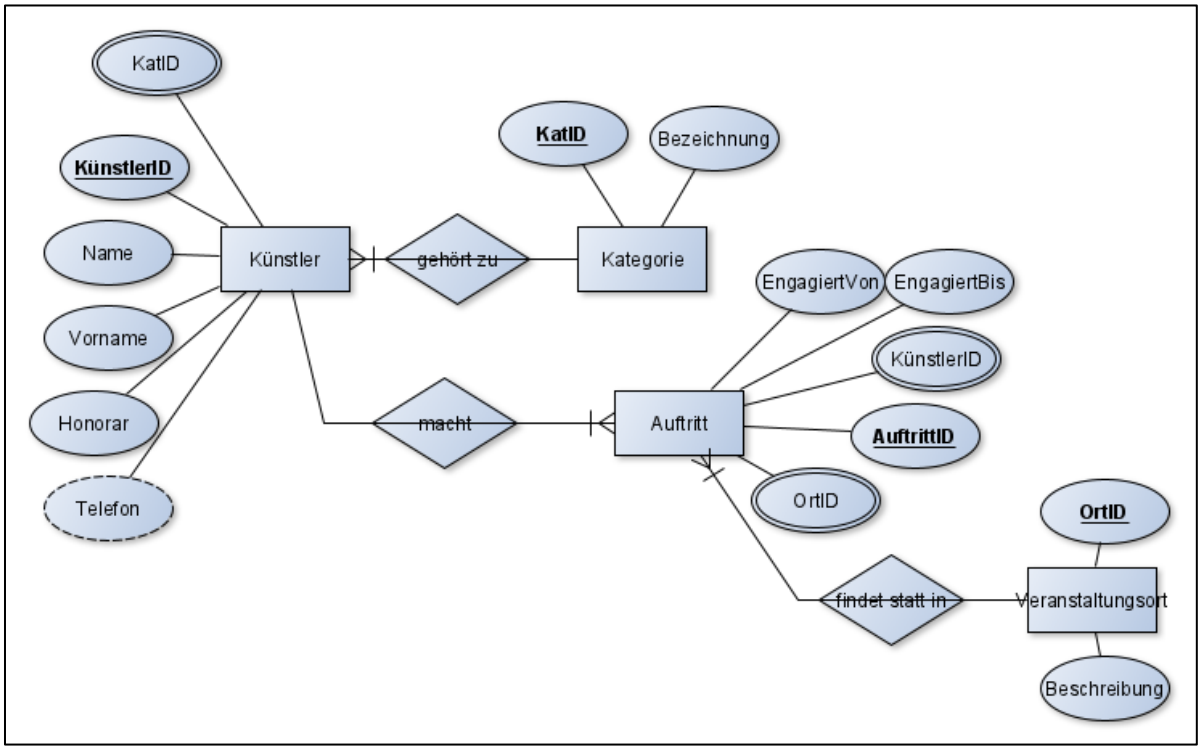
Die Daten der Tabelle „Auftritte“ sind nicht normalisiert. Nicht normalisierte Datenstrukturen führen zu Problemen bei der Datenverarbeitung, die durch den Prozess der Normalisierung vermieden werden können.

a) Erläutern Sie allgemein zwei Vorteile der Normalisierung von Datenstrukturen *(7.1.2 Softwarequalität bewerten).* (4.3.2 Daten und ihre Strukturierung) (4 Punkte)

Normalisierte Datenstrukturen vermeiden Datenredundanzen und stellen die Konsistenz der gespeicherten Daten sicher. D. h., dass z.B. Anomalien vermieden werden und Daten nicht mehrfach gepflegt/erfasst werden müssen.

b) Beschreiben Sie an einem Beispiel, was man unter einer Löschanomalie versteht. (4 Punkte) *(7.1.2 Softwarequalität bewerten)* (4.3.2 Daten und ihre Strukturierung)
Eine Löschanomalie führt zu inkonsistenten Daten in Tabellen, die in Beziehung stehende Daten verwenden. Löscht man z. B. einen Datensatz in einer übergeordneten Tabelle (z.B. Kunde), bleiben in der untergeordneten Detailtabelle (z. B. Aufträge) Datensätze bestehen, die auf den gelöschten Datensatz verweisen (z. B. über die Kundennummer beim Auftrag).

c) Normalisieren Sie die Daten der Tabelle „Auftritte“ und erstellen Sie ein vollständiges ER-Modell. (4.2.2 Daten aufbereiten,strukturieren, analysieren,visualisieren, dokumentieren und interpretieren) (7.3.7 Modellierung: z. B. eEPK, Struktogramm,UML, PAP) (16 Punkte)



Anhang

Benotungsschema:

Aufgabenteil	Teilaufgabe	Niveau	Punkte
A1	-	I	20
A2	1	I	15
	2a	II	3
	2b	II	6
	2c	II	6
	2d	II	9
	2e	III	6
A3	1a	II	3
	1b	II	3
	1c	II	3
	1d	II	4
	1e	I	1
	1f	I	1
	2a	I	4
	2b	II	4
	2c	III	16
			104

Niveau	Punkte	%
I	41	39%
II	41	39%
III	22	21%
	104	100%

Notenstufe	ab Prozent	Klausurpunkte
Ungenügend	0%	0
Mangelhaft	25%	26
Ausreichend	45%	47
Befriedigend	60%	62
Gut	75%	78
Sehr Gut	90%	94

Quelltexte des zur Verfügung gestellten, unvollständigen BlueJ-Projekts

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Agentur
{
    Kuenstlerliste meineKuenstler = new Kuenstlerliste();
    JFrame fenster = new JFrame("Agenturprogramm");
    JPanel scheibe = new JPanel();
    JButton knopf = new JButton("Künstlerliste drucken");
    JButton eingabeknopf = new JButton("Künstler eintragen");
    JLabel lname = new JLabel("Name:");
    JLabel lhonorar = new JLabel("Honorar:");
    JLabel ltelefon = new JLabel("Telefon:");
    JTextField name = new JTextField("Namen eingeben");
    JTextField telefon = new JTextField("Telefonnummer eingeben");
    JTextField honorar = new JTextField("Honorar eingeben");


    public Agentur()
    {
        knopf.addActionListener(new KnopfListener());
        eingabeknopf.addActionListener(new EingabeListener());
        guiaufbauen();
    }

    class KnopfListener implements ActionListener
    {
        public void actionPerformed(ActionEvent click)
        {
            //Klausuraufgabe
        }
```

```
    }

    class EingabeListener implements ActionListener
    {
        public void actionPerformed(ActionEvent click)
        {
            //Klausuraufgabe
        }

    }

    public void guiaufbauen()
    {
        scheibe.setLayout(new GridLayout(3,3));
        scheibe.add(lname);
        scheibe.add(ltelefon);
        scheibe.add(lhonorar);
        scheibe.add(name);
        scheibe.add(telefon);
        scheibe.add(honorar);
        scheibe.add(eingabeknopf);
        scheibe.add(knopf);
        fenster.add(scheibe);
        fenster.pack();
        fenster.setVisible(true);
    }
}
```

Klasse Kuenstlerliste

```
import java.util.ArrayList;
public class Kuenstlerliste
{
    private ArrayList<Kuenstler> dieKuenstler;

    public Kuenstlerliste()
    {
        dieKuenstler = new ArrayList<Kuenstler>();
        //ein paar Testfälle anlegen
        dieKuenstler.add(new Kuenstler("Rossi","0231-5023166",575.99));
        dieKuenstler.add(new Kuenstler("Stradivari","02304-12345",499.99));
        //Klausuraufgabe
    }

    public void kuenstlerHinzufuegen(String pName, String pTel, double
pHonorar)
    {
        dieKuenstler.add(new Kuenstler(pName,pTel,pHonorar));
    }

    public void Listedrucken()
    {
        //Klausuraufgabe
    }
  }
```

Klasse Kuenstler

```
public class Kuenstler
{
        private String name,telefon;
```

```
    private double honorar;

    public Kuenstler(String pName, String pTel, double pHonorar)
    {
        name = pName;
        telefon = pTel;
        honorar = pHonorar;
    }

    public String gibName()
    {
        return name;
    }
    public String gibTelefon()
    {
        return telefon;
    }
    public double gibHonorar()
    {
        return honorar;
    }
}
```

Klasse Musiker

```
public class Musiker extends Kuenstler
{
    String Instrument;

    public Musiker(String pName,String pTel, double pHonorar, String
pInstrument)
    {
        super(pName,pTel,pHonorar);
        Instrument = pInstrument;
    }

    public String gibInstrument()
    {
        return Instrument;
    }
}
```

Klasse Schauspieler

```
public class Schauspieler
{
    String Rolle;

    public Schauspieler()
    {

    }

}
```

Klasse Sänger

```
public class Saenger
{
    public String Stimme;

    public Saenger()
    {
```