

**Fachinformatiker/-in Fachrichtung
Anwendungsentwicklung**

FA 234

**Entwicklung und Umsetzung
von Algorithmen**

Bearbeitungszeit:
90 Minuten

Verlangt:

Alle Aufgaben

Hilfsmittel: Nicht programmierter Taschenrechner

Bewertung: Die Bewertung der einzelnen Aufgaben ist durch Punkte näher vorgegeben.

Zu beachten: Die Prüfungsunterlagen sind vor Arbeitsbeginn auf Vollständigkeit zu überprüfen.

Dieser Aufgabensatz besteht aus:

- den Aufgaben 1 bis 4
- den Anlagen 1 und 2

Bei Unstimmigkeiten ist sofort die Aufsicht zu informieren.

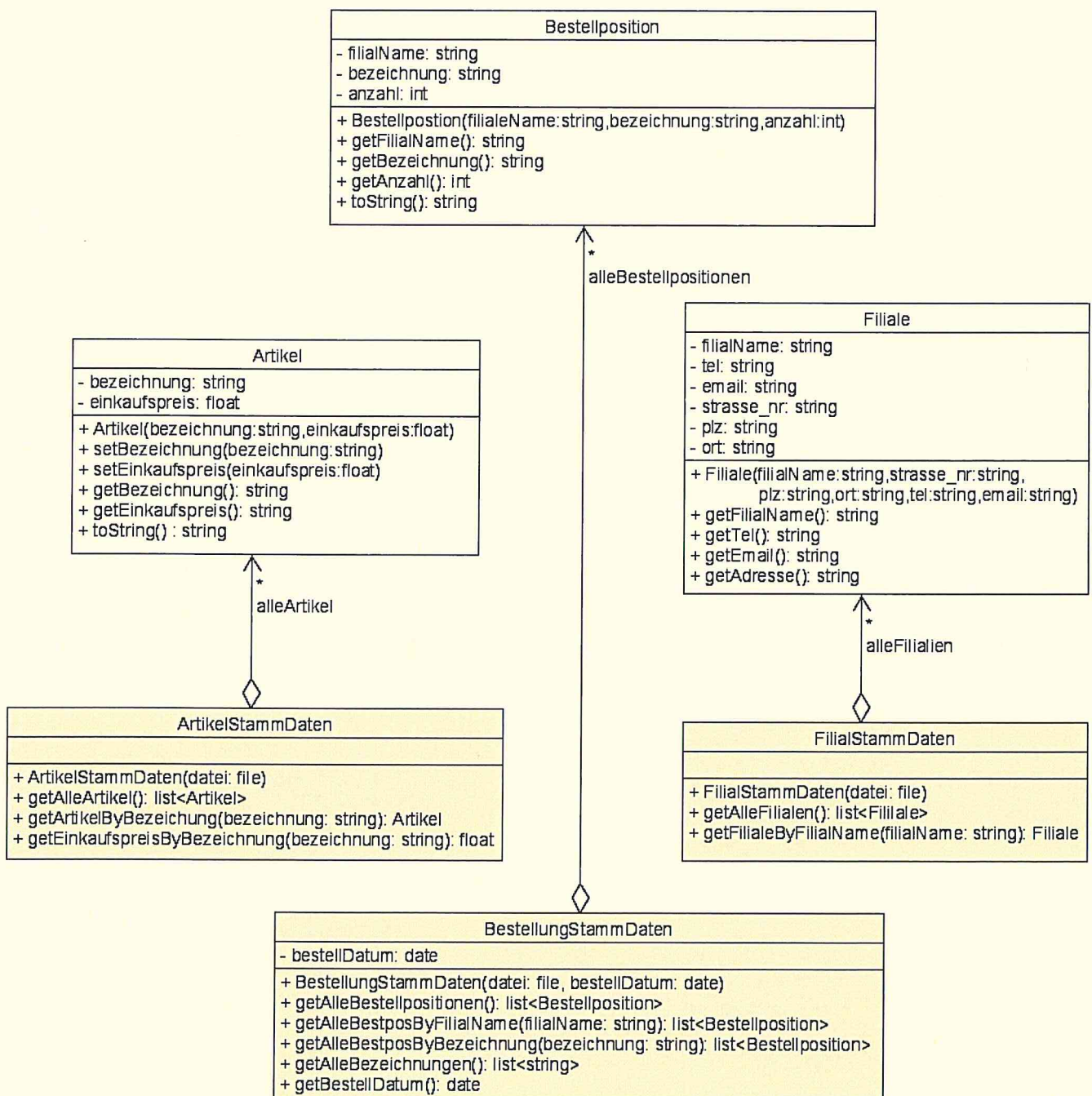
Klare und übersichtliche Darstellung der Rechengänge mit Formeln und Einheiten wird entscheidend mitbewertet.

Projekt: Umsetzen einer Bestellsoftware**Projektbeschreibung:**

In Planung befindet sich die Bestellverwaltung des Blumengroßhändlers Flowers2You. Das weltweit agierende Unternehmen baut sein Händlernetz sukzessive aus. Kleine Blumenläden können bei Flowers2You Blumen ordern.

Ihnen wurde die Umsetzung der Bestellsoftware angetragen. Ihr Projektleiter hat bereits eine Planung in Form eines UML-Klassendiagramms vorgenommen.

Dieses finden Sie nachfolgend dargestellt:



Aufgabe 1 Implementierung von Klassen und Assoziationen (Anlage 1) 20

- 1.1 Implementieren Sie die Klasse „Bestellposition“ in der an der Schule eingeführten Programmiersprache. 10
- 1.2 Erklären Sie, welche Art der Beziehung zwischen den Klassen „Artikel“ und „ArtikelStammDaten“ vorliegt 4
- 1.3 Die Datenstrukturen Listen und Mengen (Set/HashSet) stehen in der Diskussion um die Beziehung zwischen „BestellungStammDaten“ und „Bestellposition“ zu realisieren. 6

Begründen Sie anhand der Unterschiede zwischen Listen und Mengen, warum Listen in diesem Anwendungsfall die bessere Wahl sind.

Aufgabe 2 Implementierung konkreter Funktionalitäten (Anlagen 1 und 2) 36

Folgende Programmteile sollen als Programmcode umgesetzt werden.
Die Funktion der Methoden kann der Beschreibung aus Anlage 1 entnommen werden.

- 2.1 Entwickeln Sie aus der Klasse „ArtikelStammDaten“ folgende Methoden: 18

+ ArtikelStammDaten(datei: file)

+ getEinkaufspreisByBezeichnung(bezeichnung: string): float

Hinweis:

Exception handling ist nicht erforderlich.

- 2.2 Erzeugen Sie eine vorläufige Rechnung für die Bestellung der Filiale "Strauch GmbH" vom 03.05.2023. Gehen Sie davon aus, dass pro Filiale nur eine Bestellung am Tag eingeht. Entwickeln Sie hierfür die Programmsequenz mit Hilfe der Klassen „ArtikelStammDaten“, „BestellungStammDaten“ und „FilialStammDaten“. 18

Die Daten der CSV-Dateien finden Sie in Anlage 2.

Stellen Sie die Rechnung auf der Konsole wie nachfolgend abgebildet dar.

Strauch GmbH
Bergstraße 21
88000 München

Nr	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	Feuerdorn	10	5.0	50.0
2	rote Rosen	10	2.3	23.0
3	Feuerdorn	5	5.0	25.0
Gesamtpreis:				98.0

Aufgabe 3 Export

3.1 Der Großhändler

Exemplare
ren Sie
Datei h

```
<?xml  
<Beste  
<
```

```
< 41  
<1
```

```
</  
</Best
```

3.2 Begründung
austausch

Aufgabe 4 Datenbank

4.1 Zukünftige

Bislang v
händler
stellung

Im Vorge
aus dem

Erstellen
ten und l

4.2 Geben S

Kennzei

class Artikel

Implementiert eine Zugriffsklasse für einen bestimmten Artikeldatensatz.

Definierte Methoden:

Artikel(bezeichnung:string, einkaufspreis:float)

Konstruktor-Methode der Klasse Artikel. Die Bezeichnung und der Einkaufspreis werden in gleichnamigen Attributen gespeichert.

setBezeichnung(bezeichnung:string)

Speichert die Bezeichnung eines Artikels im gleichnamigen Attribut.

setEinkaufspreis(einkaufspreis:float)

Setzt den Einkaufspreis auf den übergebenen Wert.

getBezeichnung():string

Liefert die Bezeichnung eines Artikels

getEinkaufspreis():float

Liefert den Einkaufspreis eines Artikels

toString():string (Hinweis: Methodenname in Python: def `__str__(self):`)

Erzeugt eine textuelle Beschreibung des Artikels, die die Bezeichnung und den Einkaufspreis enthält.

class ArtikelStammDaten

Objekte der Klasse ArtikelStammDaten lesen aus einer zugehörigen CSV-Datei Artikel-Daten ein. Diese werden im Hauptspeicher in Form von Artikel-Objekten in einer Datenstruktur unter dem Namen `alleArtikel` abgelegt.

Definierte Methoden:

ArtikelStammDaten(datei:file)

Liest aus der im Parameter `datei` angegebenen CSV-Datei die Artikel-Daten ein und erzeugt für jeden Datensatz die zugehörigen Objekte in der Datenstruktur `alle Artikel`.

getAlleArtikel():list<Artikel>

Liefert alle Artikel, die aus der CSV eingelesen wurden.

getArtikelByBezeichnung(bezeichnung:string):Artikel

Liefert anhand einer Artikel-Bezeichnung den zugehörigen Artikel-Datensatz.

Der Einfachheit halber ist davon auszugehen, dass die Bezeichnung eindeutig ist.

getEinkaufspreisByBezeichnung(bezeichnung:string):float

Ermittelt anhand einer Artikel-Bezeichnung den zugehörigen Artikel-Datensatz und liefert hierzu den Einkaufspreis.

Der Einfachheit halber ist davon auszugehen, dass die Bezeichnung eindeutig ist.

class Bestellposition

Implementiert eine Zugriffsklasse für einen Datensatz einer Bestellposition.

Definierte Methoden:

Bestellposition(filialName:string, bezeichnung:string, anzahl:int)

Konstruktor Methode der Klasse Bestellposition. Der Name der Filiale, die Bezeichnung des Artikels sowie die Anzahl der bestellten Artikel werden in gleichnamigen Attributen gespeichert.

getFilialName():string

Liefert den Namen der Filiale, die den Artikel bestellt.

getBezeichnung():string

Liefert die Bezeichnung des Artikels, der bestellt wird.

getAnzahl():int

Liefert, wieviele Artikel dieses Artikels bestellt werden.

toString():string (Hinweis: Methodenname in Python: def __str__(self):)

Erzeugt eine textuelle Beschreibung des Bestellposition, in der alle Attribute der Bestellposition aufgeführt werden.

class BestellungStammdaten

Objekte der Klasse BestellungStammdaten lesen aus einer zugehörigen CSV-Datei Bestellpositions-Daten ein. Diese werden im Hauptspeicher in Form von Bestellposition-Objekten in einer Datenstruktur unter dem Namen alleBestellpositionen abgelegt.

Definierte Methoden:

BestellungStammdaten(datei:file, bestellDatum:date)

Liest aus der im Parameter datei angegebenen CSV-Datei die Bestell-Daten ein und erzeugt für jeden Datensatz die zugehörigen Objekte vom Typ Bestellposition in der Datenstruktur alleBestellpositionen.

getAlleBestellpositionen():list<Bestellposition>

Liefert alle Bestellpositionen.

getAlleBestposByBezeichnung(bezeichnung): list<Bestellposition>

Liefert alle Bestellpositionen derer Artikel, deren Bezeichnung gleich der Bezeichnung ist, die im Parameter bezeichnung steht.

getAlleBestposByFilialName(filialName): list<Bestellposition>

Erhält einen Namen einer Filiale und ermittelt genau die Bestellpositionen, die zu der Filiale gehören.

Das Ergebnis wird zurückgeliefert.

getAlleBezeichnungen():list<string>

Liefert in einer Liste nur die Bezeichnungen der bestellten Artikel.

getBestellDatum():date

Liefert das Datum der Bestellung

class FilialStammDaten

Objekte der Klasse FilialStammDaten lesen aus einer zugehörigen CSV-Datei Filial-Daten ein. Diese werden im Hauptspeicher in Form von Filial-Objekten in einer Datenstruktur unter dem Namen alleFilialen abgelegt.

Definierte Methoden:

FilialStammDaten(datei:file)

Ließt aus der im Parameter datei angegebenen CSV-Datei die Filial-Daten ein und erzeugt für jeden Datensatz die zugehörigen Objekte vom Typ Filiale in der Datenstruktur alleFilialen.

getAlleFilialen():list<Filiale>

Liefert alle Filialen, die aus der CSV eingelesen wurden.

getFilialeByFilialName(filialName):Filiale

Liefert anhand eines filialNamens das zugehörige Filial-Objekt. Der Name der Filiale ist als Eindeutig anzunehmen.

class Filiale

Definierte Methoden:

Filiale(filialName:string, strasse_nr:string, plz:string, ort:string, tel:string, email:string)

Konstruktor der Klasse Filiale.

Die übergebenen Parameter werden in den gleichnamigen Attributen gespeichert.

getAdresse():string

Liefert aus den Attributen filialName, strasse_nr, plz und ort einen zeilenweise formatierten Adress-String.

getEmail():string

Liefert die Email-Adresse der Filiale.

getFilialName():string

Liefert den Namen der Filiale.

getTel():string

Liefert die Telefonnummer der Filiale.

Anlage 2 Daten der CSV-Dateien zu Aufgabe 2

Artikelpreise.csv

Artikelbezeichnung;aktuellerPreis
Rhododendron;19.5
Flieder;19.5
Feldmannstreu;3
Feuerdorn;5
Glockenblume;1.8
Kugeldistel;3
Kastanie;32
Linde;42.5
Wein;10
Levkoje;1
rote Rosen;2.3

Bestelldaten.csv

Filiale;Artikel;Anzahl
Baumgarten;Flieder;30
Strauch GmbH;Feuerdorn;10
Weißmann;Kugeldistel;4
Baumgarten;Glockenblume;20
Strauch GmbH;rote Rosen;10
Strauch GmbH;Feuerdorn;5
Flora GmbH;Feldmannstreu;5
Weißmann;Kastanie;7
Weißmann;Hortensien;5
Flora GmbH;Rhododendron;7
Baumgarten;Flieder;30

Filialdaten.csv

FilialName;FilialOrt;FilialStrasse;FilialPLZ
Flora GmbH;Freiburg;Gartenlaubenweg 12;78000
Baumgarten;Bräunlingen;Hinter den Höfen 3;78199
Strauch GmbH;München;Bergstraße 21;88000
Weißmann;Zittau;Bahnhofsstraße 14;23453