

**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.

Der Aufgabensatz zu Entwicklung und Umsetzung von Algorithmen besteht aus:

- den Aufgaben 1 bis 3

Bei Unstimmigkeiten ist sofort die Aufsicht zu informieren.

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

Projektbeschreibung:

Der Fahrradkurierdienst „Flinke Pedale GbR“ bietet im Stadtkreis Karlsruhe Kurierfahrten für Postsendungen an. Derzeit sind 24 Fahrräder im Einsatz, die abwechselnd von 30 Kurierfahrern benutzt werden. Alle Räder besitzen einen GPS-Tracker, so dass sich jederzeit die aktuelle Position der einzelnen Räder ermitteln lässt. Zukünftig soll die Radflotte um 8 Lasten-E-Bikes vergrößert werden, damit das Leistungsangebot um (größere) Lastentransporte und eine schnellere Auslieferung erweitert werden kann. Sie als Auszubildender (Fachinformatiker im 3. Lehrjahr) in der Firma „Flotte Bits GmbH“, werden mit Teilen des Integrationsprozesses sowie der Anpassung vorhandener Software beauftragt.

Aufgabe 1**20****Ausgangssituation**

Die aktuelle Web-Präsenz des Kurierdienstes „Flinke Pedale GbR“ befindet sich im Anhang Anlage 1. Diese soll für die zukünftigen E-Bikes von Ihnen erweitert werden.

- 1.1 Nennen Sie drei (zeitgemäße) Möglichkeiten zur Layout-Gestaltung (Positionierung von Elementen) auf einer WEB-Site. 3
- 1.2 Erstellen Sie mit der in Ihrem Berufsschulunterricht vermittelten Methode das Layout für die WEB-Site aus Anlage 1. 8
- Hinweis:
Zu positionieren sind nur die fünf <div>-Container mit den eindeutigen Identifikatoren: id='ablauf', id='map', id='auftrag', id='angebot', id='buchung' wie in Anlage 1 vorgegeben.
- 1.3 Zukünftig soll für die horizontale und vertikale Ausrichtung des Anzeigegerätes die Darstellung unterschieden werden (Anlage 1).
Notieren Sie eine mögliche CSS-Anweisung, um diese beiden Varianten zu unterscheiden. 2
- 1.4 Begründen Sie Ihrem Auftraggeber, warum im Kontext der Barrierefreiheit von HTML-Formularen, Eingabe-Elemente korrekt mit <label> beschriftet werden müssen. 3
- 1.5 Erweitern Sie das HTML-Formular „Auftragsanfrage“ um eine verbindliche Möglichkeit zur Auswahl zwischen „schnellste Lieferung“ oder „günstigste Lieferung“. Die Auswahl „schnell“ soll dabei voreingestellt sein. Die Radio-Buttons sollen an das Script unter dem Bezeichner „LiefOpt“ einen der beiden Werte [„schnell“|„guenstig“] liefern.
Notieren Sie die notwendigen HTML-Anweisungen einschließlich der korrekten Elemente-Beschriftung. 4

Auftragsanfrage:

Start

Ziel

Paketart:

Lieferoption ☒ schnell ☐ günstig

Aufgabe 2

40

Ausgangssituation

Zur Preisberechnung existieren verschiedene Tarifvarianten und Zustellgeschwindigkeiten:

1. Der Preis wird nach Entfernung in Kilometer berechnet (Luftlinie Start-Ziel).
2. Start und Ziel liegen beide entweder in Zone 1*) oder Zone 2. Zonenübergreifende Strecken werden immer über die berechnete Distanz abgerechnet → Pauschalpreis als Obergrenze.

*) Zoneneinteilung Stadtkreis: siehe Anlage 2

Entsprechend werden die Preise wie folgt berechnet:

	Berechnete Distanz	Start und Ziel innerhalb Zone1	Start und Ziel innerhalb Zone2	Aufpreis für Schnellzustellung
Transportauftrag	3,- € je angef. km	8,- €	12,- €	5,- €

- 2.1 Längen- und Breitengrade werden im Dezimal- oder Sexagesimalsystem dargestellt (Anlage 2). 10
Entwickeln Sie ein Struktogramm für eine Funktion

sex2dez(sexagesimalWert: String): float

welche die Umrechnung vom Sexagesimal- in das Dezimalsystem realisiert.

- 2.2 Zur Preisberechnung der Zustellung wird eine Funktion/Methode 25

calcPreis(float startLON, float startLAT, float zielLON, float zielLAT, bool schnell): float

benötigt. Dabei sind LON und LAT jeweils die Längen- und Breiten-Koordinaten von „start“ und „ziel“ in Dezimaldarstellung.

Diese Funktion/Methode berechnet den Preis für die schnellste oder günstigste Variante.

Schreiben Sie den Quellcode zu dieser Funktion/Methode in der an Ihrer Berufsschule vermittelten Programmiersprache. Dabei sollen mehrfach erforderliche Berechnungen in einer Unterfunktion ausgelagert werden.

Hinweis zur Umrechnung:

LAT 1 Grad = 111,13 km, LON 1 Grad = 71,44 km

- 2.3 In einem Unit-Test soll die Funktion/Methode *calcPreis()* geprüft werden. 5
Notieren Sie einen Test-Vektor mit vier verschiedenen Eingabesätzen und den jeweils zu erwartenden Ergebnissen. Dabei sollen die Eingaben möglichst verschiedene Szenarien abbilden.
Begründen Sie die Wahl der Datensätze und nennen Sie mögliche Gründe für fehlerhafte Testergebnisse.

Beispiel:

Eingabe: 3.00 = *calcPreis(8.40383, 49.00756, 8.40308, 49.00593, false)*

Testergebnis: Ja → Funktion arbeitet korrekt, da Distanz kleiner 1 km und Kilometerpreis günstiger als Pauschalpreis für Zone 1.
Nein → Fehler, z. B. fehlerhafte Ermittlung günstigster Zustellungspreis.

Beispieldaten siehe Anlage 2.

Aufgabe 3

30

Ausgangssituation

Momentan (ohne E-Bikes) liegen die Daten in Form einer Datenbank-Tabelle (Anlage 3) vor. Zur automatischen Weiterverarbeitung werden Sie beauftragt, diese Daten aufzubereiten.

- | | | |
|-------|---|----|
| 3.1 | Nennen und erläutern Sie Ihrem Auftraggeber drei Nachteile und mögliche Fehlerquellen dieser nichtnormalisierten Datenspeicherung. | 6 |
| 3.2 | Überführen Sie diese Datenstruktur nachvollziehbar in die dritte Normalform. Beachten Sie dabei, dass redundante Daten zu eliminieren sind. | 12 |
| 3.3 | Entwickeln Sie SQL-Statements zur Abfrage auf die Ausgangstabelle „Kurierfahrten“. | |
| 3.3.1 | Ermitteln Sie den Kunden mit der längsten (gefahrte Kilometer) Kurierfahrt. Geben Sie die Kundendaten und die berechneten Kilometer aus. | 3 |
| 3.3.2 | Geben Sie eine Zusammenstellung der Tarife und die Anzahl der im jeweiligen Tarif durchgeführten Fahrten an. | 3 |
| 3.4 | Das nachfolgende SQL-Statement wird korrekt ausgeführt. Notieren Sie die Ergebnisausgabe (auf Basis der gegebenen Beispieldaten) des SQL-Clients in Tabellenform. | 6 |

```
SELECT ID_Kunde AS KundenID , SUM(km_berechnet) AS Kilometer
FROM Kurierfahrten
GROUP BY ID_Kunde
ORDER BY SUM(km_berechnet) DESC LIMIT 0,3 ;
```


Anlage 1

WEB-Site (Orientierung: Querformat)

Ablauf:

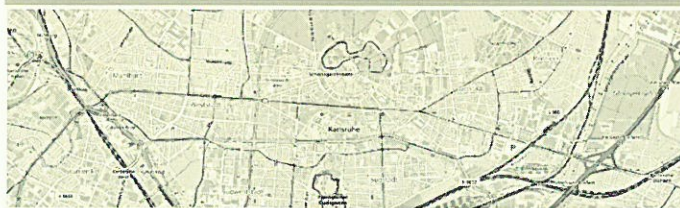
1. Wählen Sie Start und Ziel (durch Setzen von Pins auf der Karte oder Eingabe der Koordinaten) für Ihre Kurierfahrt.
2. Geben Sie Ihre Sendungsart (Paketart) an.
3. Der frühestmögliche Abholtermin (kann nach hinten verschoben werden) und Kosten werden errechnet.
4. Durch Eingabe der Kunden ID (ggf. vorher über Link als Kunde registrieren) wird der Auftrag verbindlich gebeucht.



WEB-Site (Orientierung: Hochformat)

Ablauf:

1. Wählen Sie Start und Ziel (durch Setzen von Pins auf der Karte oder Eingabe der Koordinaten) für Ihre Kurierfahrt.
2. Geben Sie Ihre Sendungsart (Paketart) an.
3. Der frühestmögliche Abholtermin (kann nach hinten verschoben werden) und Kosten werden errechnet.
4. Durch Eingabe der Kunden ID (ggf. vorher über Link als Kunde registrieren) wird der Auftrag verbindlich gebeucht.



Auftragsanfrage:

Start
Ziel
Paketart:

Angebot:

Abholung
Lieferung
Preis

Buchung:

Kunden ID

[als neuer Kunden registrieren](#)

<div id='ablauf' > ... </div>

<div id='map' > ... </div>

<div id='auftrag' > ... </div>

<div id='angebot' > ...

<div id='buchung' > ...

Anlage 2

Algorithmen



Mittelpunkt:

Karlsruhe Marktplatz;

Radius Zone 1: 4 km

Radius Zone 2: 6 km

Hinweis zur Umrechnung:

(LAT) Breitenkreis 1 Grad = 111,13 km

(LON) Längengrad 1 Grad = 71,44 km

Zustellgebiet „Flinke Pedale“

Zustellgebiet „Flinke Pedale“

Hinweis zur Umrechnung Sexagesimalsystem ins Dezimalsystem:

1° Grad = 60' Bogenminuten = 3600" Bogensekunden

1' Bogenminute = 60" Bogensekunden

Soll aus einer Sexagesimalzahl eine Dezimalzahl werden, müssen die Bestandteile der Sexagesimalzahl in Grad umgerechnet und aufsummiert werden.

Beispiel Koordinatendarstellung:

Marktplatz Karlsruhe	Breitengrad (N)	Längengrad (E)
Dezimalsystem	49.009230°	8.403903°
Sexagesimalsystem	49° 0' 33,228"	8° 24' 14,0508"

Beispieldaten (nach Google-Maps)

Von – nach	startLON	startLAT	zielLON	zielLAT	start-Zone	ziel-Zone	Entfernung
KIT Zentrum, Karlsruhe Ettlinger Tor, Karlsruhe	8,40383	49,00756	8,40308	49,00593	1	1	0,20 km
Schloss, Karlsruhe Zool. Garten, Karlsruhe	8,40442	49,01388	8,40320	48,99907	1	1	1,62 km
Kronenplatz, Karlsruhe Rosenapotheke, Rüppurr	8,40963	49,00950	8,40619	48,97101	1	2	4,28 km
Karlsruher Bürgerbüro Ost DB Regio AG, Karlsruhe	8,45741	49,02933	8,38341	48,98515	2	1	7,28 km

Anlage 3 Datenbank-Tabelle „Kurierfahrten“ (Datenauszug)

ID	Kunde	ID_Kunde	Datum	Start_LAT_LON	Ziel_LAT_LON	ID_Rad	Typ_Rad	Tarif	startZeit	zielZeit	Fahrzeit	Km_berechnet
:	:	:	:	:	:	:		:	:	:	:	:
12	Adam Adler 76149 Karlsruhe Mitteltorstrasse	200	15.06.2021	49,031051 / 8,335458	48,993220 / 8,455621	2	Trekkingbike	Zone 2	12:40:00	13:20:00	0:40:00	9,6
13	Carla Colbe 76187 Karlsruhe Schulstrasse 13	400	15.06.2021	49,000653 / 8,302155	49,035553 / 8,456822	3	Rennrad	km	8:10:00	9:00:00	0:50:00	11,7
14	Adam Adler 76149 Karlsruhe Mitteltorstrasse	200	16.06.2021	49,053133 / 8,378037	48,990090 / 8,408626	4	Lastenrad	Zone 2	9:35:00	10:05:00	0:30:00	7,4
15	Emil Eisen 76139 Karlsruhe Hallesche Alle 4	600	17.06.2021	48,959814 / 8,289185	49,006177 / 8,532679	2	Trekkingbike	km	20:00:00	21:20:00	1:20:00	18,2
16	Bernd Bayer 76135 Karlsruhe Scheffelstrasse 13	300	17.06.2021	49,034674 / 8,286272	48,962205 / 8,466161	2	Trekkingbike	km	17:15:00	18:15:00	1:00:00	15,2
17	Dora Daubner 76275 Ettlingen Mörschstr21	500	18.06.2021	48,996462 / 8,352547	49,000762 / 8,389690	3	Rennrad	Zone 1	7:35:00	7:45:00	0:10:00	2,7
18	Bernd Bayer 76135 Karlsruhe Scheffelstrasse 13	300	19.06.2021	48,962045 / 8,290884	49,002673 / 8,504518	3	Rennrad	km	9:50:00	11:00:00	1:10:00	15,9
19	Carla Colbe 76187Karlsruhe Schulstrasse 13	400	21.06.2021	49,081763 / 8,396002	48,973361 / 8,405712	2	Trekkingbike	km	5:25:00	6:15:00	0:50:00	12,1
20	Dora Daubner 76275 Ettlingen Mörschstr21	500	21.06.2021	49,028944 / 8,368569	49,026079 / 8,471502	3	Rennrad	Zone 2	14:00:00	14:30:00	0:30:00	7,4
21	Dora Daubner 76275 Ettlingen Mörschstr21	500	21.06.2021	49,010158 / 8,373425	49,027671 / 8,441884	4	Lastenrad	Zone 1	12:40:00	13:00:00	0:20:00	5,3
22	Adam Adler 76149 Karlsruhe Mitteltorstrasse	200	23.06.2021	48,977345 / 8,315403	49,052656 / 8,386048	3	Rennrad	km	9:50:00	10:30:00	0:40:00	9,8
:	:	:	:	:	:	:		:	:	:	:	:

Fachinformatiker/-in Fachrichtung Anwendungsentwicklung
Entwicklung und Umsetzung von Algorithmen**FA 234****Lösungsvorschläge:**

Lösungsvorschläge sind in der Regel Vorschläge der einreichenden Schulen; sie sind im Wortlaut nicht bindend. Anderslautende, aber zutreffende Antworten sind ebenfalls als richtig zu werten.

**Nur für die Hand
des Prüfers!
Punkte**

Aufgabe 1**20**

- 1.1 - CSS – Grid Layout 3
 - CSS – Flexboxen
 - CSS – Frameworks (z. B. Bootstrap)
 - CSS – Eigenschaft position:absolut (zusammen mit Größenangaben)
 - HTML – Tabellen

- 1.2 8

// HTML - Struktur

```
<div class="grid-container">
  <div id="ablauf"> ...</div>
  <div id="map"> ... </div>
  <div id="auftrag"> ... </div>
  <div id="angebot"> ... </div>
  <div id="buchung"> ... </div>
</div>
```

// CSS - Formatanweisungen

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(4 , 25vw);
  grid-template-rows: repeat(4 , 25vh);
}
```

```
#ablauf {
  grid-column-start:1; grid-column-end:5 ;
  grid-row-start:1 ; grid-row-end:1;
}
```

```
#map {
  grid-column: 1/4;
  grid-row:2/5 ;
}
```

```
#auftrag {
  grid-column-start:4; grid-column-end:5 ;
  grid-row-start:2 ; grid-row-end:3;
}
```

```
#angebot {
  grid-column-start:4; grid-column-end:5 ;
  grid-row-start:3 ; grid-row-end:4;
}
```

```
#buchung {
  grid-column-start:4; grid-column-end:5 ;
  grid-row-start:4 ; grid-row-end:5;
}
```

- 1.3 Mit Medienabfragen (Media Queries) können Sie die Darstellung eines Dokuments für verschiedene Ausgabemedien festlegen: 2
 - @media screen and (orientation: portrait) { ... }
 - @media screen and (orientation: landscape) { ... }

- 1.4 Damit ein HTML-Formular barrierefrei genutzt werden kann, benötigt jedes Steuerelement eine Beschriftung, die dem Element eindeutig zugeordnet ist und beispielsweise bei der Nutzung eines Screenreaders die Beschreibung wiedergibt (Web Content Accessibility Guidelines (WCAG)). 3

- 1.5 `<input type="radio" id="schnell" name="LiefOpt" checked value="schnell">`
`<label for="schnell">schnell</label>`
`<input type="radio" id="guenstig" name="LiefOpt" value="guenstig">`
`<label for="guenstig">günstig</label>` 4

Aufgabe 2

40

2.1

10

```

sexa2dez(sexaWert: String):float
//Beispiel für Übergabewert: 49° 0' 33,228"

```

```

lokale Variablen:
var dezWert: float
var gms: StringArray[]
var grad, minute, sekunde: float

```

```

gms ← Teile sexaWert an den Leerzeichen in Teilstrings.

```

```

grad ← Entferne bei gms[0] das '°'-Zeichen und Umwandlung in float

```

```

minute ← Entferne bei gms[1] das '-Zeichen und Umwandlung in float

```

```

sekunde ← Entferne bei gms[2] das "'-Zeichen und Umwandlung in float

```

```

dezWert ← grad + minute/60 + sekunde/3600

```

```

return dezWert

```

2.2

25

```

private static double calcPreis(double startLon, double startLat, double ziellon, double ziellat, boolean schnell) {
    /* ----- Dummydaten für Test ----- */

    double preisProKilometerStandard = 3.00;
    double preisZone1Standard = 8.00;
    double preisZone2Standard = 12.00;
    double schnellzustellungStandard = 5.00;

    double zone1 = 4, zone2 = 6;
    double zoneCenterLat = 49.009225, zoneCenterLon = 8.403908;

    /* ----- */

    double minPreis = 0, distPreis;

    double entfernungStarpunktZone = calcDistance(zoneCenterLon, zoneCenterLat, startLon, startLat);
    double entfernungZielpunktZone = calcDistance(zoneCenterLon, zoneCenterLat, ziellon, ziellat);
    double entfernungStartZiel = calcDistance(startLon, startLat, ziellon, ziellat);

    distPreis = entfernungStartZiel * preisProKilometerStandard;

    if(schnell == true) {
        minPreis = schnellzustellungStandard;
    }

    if (entfernungStarpunktZone <= zone1 && entfernungZielpunktZone <= zone1) {
        if (distPreis < preisZone1Standard) {
            minPreis += distPreis;
        } else {
            minPreis += preisZone1Standard;
        }
    } else if (entfernungStarpunktZone <= zone2 && entfernungZielpunktZone <= zone2) {
        if (distPreis < preisZone2Standard) {
            minPreis += distPreis;
        } else {
            minPreis += preisZone2Standard;
        }
    } else {
        minPreis += distPreis;
    }

    return minPreis;
}

private static double calcDistance(double startLon, double startLat, double ziellon, double ziellat) {
    double distfaktorLAT = 111.13;
    double distfaktorLON = 71.44;

    double entfernung, distLat, distLon;

    distLat = (ziellat - startLat) * distfaktorLAT;
    distLon = (ziellon - startLon) * distfaktorLON;

    entfernung = Math.sqrt(distLat * distLat + distLon * distLon);

    return entfernung;
}

```


2.3 Schülerabhängige Lösungen. Sinnvolle Testdaten wären z. B. 5

- Start- und Zielpunkt sind identisch
- Entfernung zwischen Start- und Zielpunkt kleiner als Zonengröße, Start- und Zielpunkt jedoch außerhalb der Zonen.
- Zielpunkt liegt direkt auf Zonengrenze

Hinweis: Begründung der SuS sollten bei einem fehleranfälligen Grenzfall erklären, weshalb es zu einem unerwarteten Ergebnis kommen könnte.

Aufgabe 3

30

3.1 - Ein Nachteil nichtnormalisierter Datenbasen ist der erhöhte Speicherbedarf infolge von redundanten Werten. 6

- Redundante Werte können zu Anomalien (Fehlern) bei der Datenmanipulation führen.

- Attribute, die mehrwertige Daten enthalten lassen sich schlecht auslesen und mit anderen Daten in Beziehung setzen.

3.2 Ausgang: Tabelle Kurierfahrten mit ID als Primärschlüssel in Relationen Darstellung 12

Kurierfahrten (ID, Kunde, ID_Kunde, Datum, Start_LAT_LON, Ziel_LAT_LON, ID_Rad, Typ_Rad, Tarif, startZeit, zielZeit, Fahrzeit, Km_berechnet)

Schritt 0: Beseitigung redundanter Daten laut Aufgabenstellung
Km_berechnet und Fahrzeit sind berechenbar und somit redundant.
Kurierfahrten (ID, Kunde, ID_Kunde, Datum, Start_LAT_LON, Ziel_LAT_LON, ID_Rad, Typ_Rad, Tarif, startZeit, zielZeit)

Schritt 1: 1.Normalform (Atomarität)
Kunde ist nicht atomar -> muss zerlegt werden,
Start_LAT_LON, Ziel_LAT_LON ebenso
(oder in MySQL/MariaDB Verwendung des Datentyps **POINT**).
Kurierfahrten (ID, K_Zuname, K_Vorname, K_PLZ, K_Ort, K_Strasse, K_HausNr, ID_Kunde, Datum, Start_LAT_LON, Ziel_LAT_LON, ID_Rad, Typ_Rad, Tarif, startZeit, zielZeit)

Schritt 2: 2.Normalform (1NF + voll funktionale Abhängigkeit) Zweite Normalform liegt vor.

Schritt 3: 3.Normalform (1NF + 2NF + keine transitiven Abhängigkeiten)
Kunde (mit allen Attributen) ist nur von ID_Kunde abhängig.
Typ_Rad ist nur von ID_Rad abhängig
Kunde (ID_Kunde, K_Zuname, K_Vorname, K_PLZ, K_Ort, K_Strasse, K_HausNr)
Rad (ID_Rad, Typ_Rad)
Kurierfahrten (ID, ID_Kunde, Datum, Start_LAT_LON, Ziel_LAT_LON, ID_Rad, Tarif, startZeit, zielZeit)

3.3.1 **SELECT** kunde, Km_Berechnet **FROM** kurierfahrten 3

ORDER BY Km_Berechnet **DESC LIMIT** 1 ;

3.3.2 **SELECT** Tarif , **COUNT**(Tarif) **AS** Anzahl 3

FROM kurierfahrten **GROUP BY** Tarif ;

3.4 Ausführung ist einfach (und sollte ohne Taschenrechner möglich sein). 6

Zuerst wird für jeden Kunden die Summe der gefahrenen km berechnet.

Danach wird absteigend nach dieser Summe sortiert und die ersten drei Zeilen dargestellt.

KundenID	Kilometer
300	31,1
200	26,8
400	23,8