Diese Kopfleiste bitte unbedingt ausfüllen! Familienname, Vorname (bitte durch eine Leerspalte trennen)

Berufsnummer

0 2

Bereich

IHK-Nummer





Abschlussprüfung Winter 2024/25

Prüflingsnummer

1201

Entwicklung und Umsetzung von Algorithmen Fachinformatiker Fachinformatikerin Anwendungsentwicklung

Termin: Mittwoch, 27. November 2024

Teil 2 der Abschlussprüfung

4 Aufgaben mit Belegsatz 90 Minuten Prüfungszeit 100 Punkte

Bearbeitungshinweise

- Bevor Sie mit der Bearbeitung der Aufgaben beginnen, überprüfen Sie bitte die Vollständigkeit dieses Aufgabensatzes. Die Anzahl der zu bearbeitenden Aufgaben ist auf dem Deckblatt links angegeben. Wenden Sie sich bei Unstimmigkeiten sofort an die Aufsicht, weil Reklamationen am Ende der Prüfung nicht anerkannt werden können
- Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
- Lesen Sie bitte den Text der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
- 4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die Vorgaben der Aufgabenstellung zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier
- Tragen Sie die frei zu formulierenden Antworten dieser offenen Aufgaben in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
- Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine stichwortartige Beantwortung zulässig
- Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder unleserliches Ergebnis wird als falsch gewertet.
- Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger Taschenrechner ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
- 9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
- 10. Für Hilfsaufzeichnungen können Sie das in der Tasche beigelegte Konzeptpapier verwenden. Bewertet werden jedoch grundsätzlich nur Ihre Eintragungen in diesem Aufgabensatz.

Bewertung Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen. 2. Aufg. Punkte 3. Aufg. 1. Aufg. Punkte Punkte 4. Aufg 15 16 19 20 21 22

| Prüfungszeit | |
|--|-----------------|
| Die entsprechende finden Sie in der Prüfungszeit im Ar Aufgabe. | Abfrage nach de |

Wird vom Korrektor ausgefüllt!

Gesamtpunktzahl

| 1 | in | 100 | 1111 |
|---|----|-----|------|
| | | | |
| | | | |
| | 24 | 25 | 26 |

Prüfungsort, Datum

Unterschrift

Sie Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:

In Kiel soll das Nahverkehrssystem (KVAG) um eine neue Stadtbahn erweitert werden. In Zusammenhang mit dieser Erweiterung sollen die IT-Systeme der KVAG erweitert und erneuert werden. Dazu werden verschiedene Teilprojekte festgelegt. Diese betreffen unter anderem die Fahrplanauskunft, die Buchungssysteme, die Anzeigen und die Vernetzung. Alle Teilprojekte sollen auch hinsichtlich der Anforderungen von Datenschutz und Datensicherheit untersucht werden.

Korrekturrand

1. Aufgabe (25 Punkte)

Die KVAG möchte ihre Fahrpläne besser an den tatsächlich vorhandenen Fahrzeiten ausrichten. Dazu wurden die Abfahrtszeiten einer Bahn an den einzelnen Haltestellen für eine Fahrt einer Bahnlinie an mehreren Tagen gemessen.

Für jeden Tag ist festgehalten, wann die Bahn die nachfolgenden Haltestellen verlassen hat (siehe Array zeiten).



Es existiert eine Klasse Abfahrtszeit, mit deren Attributen ein Datum, eine Haltestellennummer, die planmäßige Abfahrtszeit (in Minuten seit Mitternacht) und die tatsächliche Abfahrtszeit (ebenfalls in Minuten) gespeichert wird. Für die Endstation sind hier die planmäßigen bzw. tatsächlichen Ankunftszeiten gespeichert.

Klasse Abfahrtszeit

| Abfahrtszeit | |
|----------------------------|---|
| - datum : Date | |
| - haltestellenNr : Integer | |
| - planAbfahrt : Integer | <pre>// planmäßige Abfahrtszeit in Minuten seit Mitternacht</pre> |
| - istAbfahrt : Integer | // tatsächliche Abfahrtszeit in Minuten |
| ibemiunic . integer | seit Mitternacht |

Für jedes Attribut ist eine öffentliche Get-Methode vorhanden.

In dem eindimensionalen Array zeiten vom Typ Abfahrtszeit sind die Daten des untersuchten Zeitraums gespeichert. In einer Zeile sind die Daten eines Abfahrtszeit-Objektes aufgeführt.

Auflistung der Daten des Arrays zeiten

| Datum | Haltestellen Nummer | Planmäßige Abfahrt | Tatsächliche Abfahrt |
|----------|---------------------|--------------------|----------------------|
| 1.9.2024 | 0 | 480 | 480 |
| 1.9.2024 | 1 | 483 | 483 |
| 1.9.2024 | 2 | 485 | 486 |
| | | | |
| 1.9.2024 | 15 | 579 | 583 |
| 2.9.2024 | 0 | 480 | 480 |
| 2.9.2024 | 1 | 483 | 484 |
| | | | |
| 2.9.2024 | 15 | 579 | 582 |
| | | | |

Die Fahrzeit auf einer Strecke ist die Differenz zweier Abfahrtszeiten aufeinanderfolgender Haltestellen. Eine Verspätung liegt vor, wenn die Fahrzeit auf einer Strecke mehr als zwei Minuten länger war als die planmäßige Fahrzeit.

Es soll ein Algorithmus für eine Funktion

ermittleFahrzeiten(Abfahrtszeit[] zeiten) : Integer[]

entwickelt werden, der für jede der 15 Strecken die Anzahl der Verspätungen ermittelt.

Die ermittelten Häufigkeiten sollen in einem eindimensionalen Array vom Typ Integer gespeichert und von der Funktion zurückgegeben werden.

Ergebnis-Array (Beispiel):

| Index | Anzahl Fahrzeiten > 2 | Kommentar | |
|-------|-----------------------|-------------------------------------|--------------------|
| [0] | 0 | Anzahl Verspätungen (im Beispiel 0) | auf der Strecke 0 |
| [1] | 1 | · co | auf der Strecke 1 |
| [2] | 4 | | auf der Strecke 2 |
| | | | |
| [14] | 3 | | auf der Strecke 14 |

Korrekturrand

Stellen Sie den Algorithmus in Pseudocode dar.

ermittle_fahrzeiten(Abfahrtszeit[] zeiten) : Integer[]

- fahrplanService: FahrplanService

Die Fahrgäste sollen die Möglichkeit haben, in der App die aktuellen Abfahrtszeiten und Verspätungen einsehen zu können. Diese Daten sollen über eine API abgerufen werden. Ihre Aufgabe ist es, den API-Endpunkt zu implementieren.

Folgender Auszug aus dem Klassendiagramm ist gegeben:

+ handleAbfahrtenRequest(haltestellenId: int): HttpResponse

FahrplanController

| createError | Response(nachricht: string, statusCode: int): HttpResponse |
|---------------------------------|---|
| - getNaechst | eAbfahrten(haltestellenld: int, maxAbfahrten: int): Abfahrt[] |
| | |
| | FahrplanService |
| + getAbfahrt | en(haltestellenid: int): Abfahrt[] |
| | |
| | |
| | Abfahrt |
| + verbindunç | |
| + verbindunç + haltesteller | gsld: int |

| HttpResponse | |
|--|--|
| - statusCode: int - headers: HttpHeader[] | |
| | |
| + HttpResponse(code: int) | |
| + addHeader(name: string, value: string): void | |
| + setBody(content: string): void | |

| DateTime |
|------------------------------|
| + now(): DateTime |
| + compare(dt: DateTime): int |

Die Methode compare aus der Klasse DateTime gibt folgende Rückgabewerte zurück:

- Wenn der Datumswert des aktuellen Objekts vor dem des übergebenen Objekts ist: -1
- Wenn beide Datumswerte gleich sind: 0
- Wenn der Datumswert des aktuellen Objekts nach dem des übergebenen Objekts ist: 1
- a) In der Klasse FahrplanController soll zunächst die Methode createErrorResponse(nachricht: string, statusCode: int): HttpResponse. implementiert werden.

Die Nachricht und der Status-Code werden übergeben.

Die erzeugte HTTP-Antwort soll wie folgt aussehen.

HTTP < der übergebene Status-Code >

Content-Type: text/plain

Content-Length: < Anzahl der Bytes des Response Bodys >

< Response Body: die übergebene Nachricht >

Implementieren Sie die Methode createErrorResponse.

10 Punkte

| | | <u> </u> |
|---|--|----------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| * | | |
| | | |
| | | |

private createErrorResponse(nachricht: string, statusCode: int): HttpResponse

| b) | Die App soll die nächsten Abfahrten anzeigen. | Korrekturrand |
|----|--|---------------|
| | Dazu soll in der Klasse FahrplanController die private Methode getNaechsteAbfahrten (haltestellenId: int, maxAbfahrten: int): Abfahrt[] erstellt werden. | |
| | Diese Methode soll die nächsten Abfahrten ab dem aktuellen Zeitpunkt zurückgeben. maxAbfahrten gibt an, wie viele Abfahrten maximal zurückgegeben werden sollen. | |
| | Die Methode soll das Abfahrt Array mit den ermittelten Abfahrten zurückgeben. | |
| | Mithilfe der Methode FahrplanService.getAbfahrten können die Abfahrten von einer Haltestelle abgerufen werden. getAbfahrten liefert ein aufsteigend nach Abfahrtszeit sortiertes Array von Abfahrten zurück. | |
| | Implementieren Sie die Methode getNaechsteAbfahrten. 15 Punkte | |
| | <pre>private getNaechsteAbfahrten (haltestellenId: int, maxAbfahrten: int): Abfahrt[]</pre> | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | , | |
| | | |

| 3. Aufgabe (29 Punkte) | Korrekturrand |
|--|---|
| a) Es sollen stichprobenartig Kontrollen von Mitarbeitern der Stadtbahn auf den Fahrten durchgeführt werden. Die in einer relationalen Datenbank verwaltet werden. Für einen ersten Entwurf eines relationalen Datenmodells lie | egen Ihnen |
| folgende Informationen vor: — Jede Kontrolle findet immer auf einer bestimmten Fahrt statt. | |
| Auf einer Fahrt können auch mehrere Kontrollen durchgeführt werden. | |
| Zu jeder Kontrolle gehören mindestens zwei Mitarbeiter. | |
| Bei jeder Kontrolle kann es zu verschiedenen Arten von Vorkommnissen kommen. | |
| Jedem Vorkommnis wird die Art der Maßnahme zugordnet, welche ergriffen wurde. | |
| Erstellen Sie ein relationales Datenmodell, welches der 3. Normalform genügt. Tragen Sie in die Tabellen alle no Primär- und Fremdschlüssel ein und kennzeichnen Sie diese mit PK bzw. FK. Weitere Attribute können vernachlä | otwendigen ässigt werden. 16 Punkte |
| Tragen Sie alle Beziehungen zwischen den Tabellen mit den entsprechenden Kardinalitäten ein. | To Pulikte |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| * | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

| b) Im Vorfeld der Einführung des neuen Systems wurden bereits Daten manuell erfasst. Jetzt wird in Betrac | ht gezogen, diese Da- |
|---|-----------------------|
| ten ins System zu importieren. Folgender Auszug aus einer Tabelle mit den Daten wurde erstellt. Der Dat | enumfang beträgt über |
| 1.000 Zeilen: | |

| Datum | Mitarbeiter | Fahrt | Vorkommnis | Maßnahme |
|-----------|------------------|----------|---------------------|------------------------------|
| 1.7. | Müller, Meier | Linie 13 | Kein Ticket | erhöhtes Beförderungsentgelt |
| 3.8.23 | Schmitt, Müller | 1 | Ohne Fahrkarte | 60 EUR Strafe |
| 31.4. | Mueller, Schmitt | L 4 | Randalierer | Anzeige |
| 4.7. | Müller; Meier | Linie 6 | Kein Ticket | 60 EUR |
| 2023-4-13 | Schmitt, | Linie17 | Ticket nicht lesbar | Verwarnung |
| 23.6.23 | Schmitt; Hans | Linie 3 | Kein Fahrschein | 60 EUR erhöhtes Entgelt |

| oa) Die vorliegenden Daten sollen beurteilt werden. | | |
|---|----------------------|--|
| Beschreiben Sie drei Probleme bezüglich der Qualität der Daten und nennen Sie jeweils eine Folge, die sich du Import der mangelhaften Daten ergeben würde. | ırch den 9 Punkte | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| b) Zeigen Sie einen möglichen Weg auf, wie die Daten dennoch importiert werden können. | 2 Punkte | |
| | | |
| | | |
| | | |
| c) Beurteilen Sie abschließend die Importidee und argumentieren Sie für oder gegen den Import der Daten. | 2 Punkte | |
| | | |
| | | |
| | | |

| e Tabellenauszüge (siehe perforierte Anlage) stehen für die im Folgenden beschrieben SQL-Anweisungen zur Verfügung: Erstellen Sie eine SQL-Anweisung, mit der Sie alle aktiven Haltestellen der Linie 250 erhalten. 3 Punkte Ergebnistabelle Am Faulbach Von-Lohe-Str. Windmühlenstr. Ebertplatz Derstellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HST_Linie HST_Linie HSTP_Abfahrt_Plan Am Faulbach 250 2024-12-01 12:12:00 Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 10:03:00 | Aufgabe (21 Punl | | | | | Korrekturrand |
|--|----------------------|------------------|---------------------------------------|--------------------------------------|-----------|---------------|
| Ergebnistabelle: Name der Haltestelle | e Tabellenauszüge (s | iehe perforierte | Anlage) stehen für die im Folgende | n beschrieben SQL-Anweisungen zur Ve | erfügung: | |
| Ergebnistabelle: Name der Haltestelle | Erstellen Sie eine S | QL-Anweisung, | nit der Sie alle aktiven Haltestellen | der Linie 250 erhalten. | 3 Punkte | |
| Name der Haltestelle Am Faulbach Von-Lohe-Str. Windmühlenstr. Ebertplatz | | | | | | |
| Am Faulbach Von-Lohe-Str. Windmühlenstr. Ebertplatz Destellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HSt_Name HSt_Linie HStP_Abfahrt_Plan Am Faulbach 250 2024-12-01 12:12:00 Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 10:03:00 | | stelle | | | | |
| Von-Lohe-Str. Windmühlenstr. Ebertplatz Berstellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HSt_Name HSt_Linie HStP_Abfahrt_Plan Am Faulbach 250 2024-12-01 12:12:00 Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 10:03:00 | | | | | | |
| Windmühlenstr. Ebertplatz Derstellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HSt_Name | | | | | | |
| Erstellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HSt_Name HSt_Linie HStP_Abfahrt_Plan Am Faulbach 250 2024-12-01 12:12:00 Am Faulbach 250 2024-12-01 10:03:00 Am Faulbach 250 2024-12-01 10:03:00 | | | | | | |
| Destellen Sie eine SQL-Anweisung, mit der Sie alle Abfahrtzeiten der Haltestelle "Am Faulbach" absteigend nach Datum und Uhrzeit sortiert erhalten. 6 Punkte Ergebnistabelle: HSt_Name HSt_Linie HStP_Abfahrt_Plan Am Faulbach 250 2024-12-01 12:12:00 Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 10:03:00 | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | 22000 | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Uhrzeit sortiert erhalten. Ergebnistabelle: HSt_Name | | | | | | |
| Am Faulbach 250 2024-12-01 11:12:00 Am Faulbach 250 2024-12-01 10:03:00 | | HSt_Linie | HStP_Abfahrt_Plan | | | |
| Am Faulbach 250 2024-12-01 10:03:00 | Am Faulbach | 250 | 2024-12-01 12:12:00 | | | |
| | Am Faulbach | 250 | 2024-12-01 11:12:00 | | | |
| | Am Faulbach | 250 | 2024-12-01 10:03:00 | | | |
| | | | | | | |
| | | | | | | |
| | | | | | - | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | 5 | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Dieses Blatt kann an der Perforation aus dem Aufgabensatz herausgetrennt werden!

Tabelle: Haltestelle

| HSt_IdKey | HSt_Name | HSt_Aktiv | HSt_Linie |
|-----------|----------------------|-----------|-----------|
| 1 | Halfengasse | 1 | 124 |
| 2 | Nesselrodestr. | 1 | 124 |
| 3 | Amsterdamer Gürtel | 1 | 124 |
| 4 | Xantener Str. | 1 | 124 |
| 5 | Worringer Str. | 1 | 124 |
| 6 | Am Faulbach | 1 | 250 |
| 7 | Von-Lohe-Str. | 1 | 250 |
| 8 | Schützenhofstr. | 0 | 250 |
| 9 | Windmühlenstr. | 1 | 250 |
| 10 | Reichensberger Platz | 1 | 124 |
| 11 | Ebertplatz | 1 | 250 |
| 12 | Ebertplatz | 1 | 140 |
| weitere | | | |

Tabelle: Haltestelle_Plan (Planzeiten der täglich geplanten Ankünfte und Abfahrten)

| LICAD LAIV | HC4D HC41 H | USID A L SI DI | |
|------------|---------------|---------------------|---------------------|
| HStP_IdKey | HStP_HStIdKey | HStP_Ankunft_Plan | HStP_Abfahrt_Plan |
| 1 | 1 | 2024-12-01 10:01:00 | 2024-12-01 10:01:00 |
| 2 | 2 | 2024-12-01 10:04:00 | 2024-12-01 10:04:00 |
| 3 | 3 | 2024-12-01 10:08:00 | 2024-12-01 10:08:00 |
| 4 | 4 | 2024-12-01 10:09:00 | 2024-12-01 10:09:00 |
| 5 | 5 | 2024-12-01 10:13:00 | 2024-12-01 10:18:00 |
| 6 | 10 | 2024-12-01 10:22:00 | 2024-12-01 10:22:00 |
| 7 | 6 | 2024-12-01 10:03:00 | 2024-12-01 10:03:00 |
| 8 | 7 | 2024-12-01 10:06:00 | 2024-12-01 10:06:00 |
| 9 | 9 | 2024-12-01 10:11:00 | 2024-12-01 10:11:00 |
| 10 | 11 | 2024-12-01 10:12:00 | 2024-12-01 10:12:00 |
| 11 | 6 | 2024-12-01 11:12:00 | 2024-12-01 11:12:00 |
| 12 | 6 | 2024-12-01 12:12:00 | 2024-12-01 12:12:00 |
| weitere | | | |

Tabelle: Haltestelle_Zeiten (Istzeiten der täglichen Ankünfte und Abfahrten)

| HStZ_IdKey | HStZ_HStldKey | HStZ_HStPIdKey | HStZ_Ankunft | HSt_Abfahrt |
|------------|---------------|----------------|---------------------|---------------------|
| 1 | 1 | 1 | 2024-12-01 10:01:00 | 2024-12-01 10:01:00 |
| 2 | 2 | 2 | 2024-12-01 10:05:00 | 2024-12-01 10:05:00 |
| 3 | 3 | 3 | 2024-12-01 10:09:00 | 2024-12-01 10:09:00 |
| 4 | 4 | 4 | 2024-12-01 10:10:00 | 2024-12-01 10:10:00 |
| 5 | 5 | 5 | 2024-12-01 10:15:00 | 2024-12-01 10:18:00 |
| 6 | 10 | 6 | 2024-12-01 10:22:00 | 2024-12-01 10:22:00 |
| 7 | 6 | 7 | 2024-12-01 10:03:00 | 2024-12-01 10:03:00 |
| 8 | 7 | 8 | 2024-12-01 10:06:00 | 2024-12-01 10:06:00 |
| 9 | 9 | 9 | 2024-12-01 10:11:00 | 2024-12-01 10:11:00 |
| 10 | 11 | 10 | 2024-12-01 10:12:00 | 2024-12-01 10:12:00 |
| weitere | | | | |

c) Erstellen Sie eine SQL-Anweisung, mit der Sie alle Verspätungen aller Haltestellen erhalten. Ergebnistabelle:

12 Punkte

Korrekturrand

| HSt_Name | HSt_Linie | HStP_Ankunft_Plan | HStZ_Ankunft | Verspätung in Minuten |
|--------------------|-----------|---------------------|---------------------|-----------------------|
| Nesselrodestr. | 124 | 2024-12-01 10:04:00 | 2024-12-01 10:05:00 | 1 |
| Amsterdamer Gürtel | 124 | 2024-12-01 10:08:00 | 2024-12-01 10:09:00 | 1 |
| Xantener Str. | 124 | 2024-12-01 10:09:00 | 2024-12-01 10:10:00 | 1 |
| Worringer Str. | 124 | 2024-12-01 10:13:00 | 2024-12-01 10:15:00 | 2 |

| PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜF | |
|---|------|
| PRITEING VEHI - NICHI BEVIANDIEII DER DRITE | LING |

| Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende | Wie | beurteilen Sie nach | ı der I | Bearbeitung | der | Aufgaben | die zu | r Verfügung | stehende | Prüfungszeit |
|---|-----|---------------------|---------|-------------|-----|----------|--------|-------------|----------|--------------|
|---|-----|---------------------|---------|-------------|-----|----------|--------|-------------|----------|--------------|

- Sie hätte kürzer sein können.
 Sie war angemessen.
 Sie hätte länger sein müssen.