

Hinweise für den Prüfling

Bearbeitungszeit: 300 Minuten

Auswahlverfahren

Wählen Sie von den zwei vorliegenden Vorschlägen einen zur Bearbeitung aus. Der nicht ausgewählte Vorschlag wird 60 Minuten nach Beginn der Bearbeitungszeit von der Aufsicht führenden Lehrkraft eingesammelt.

Erlaubte Hilfsmittel

1. ein Wörterbuch der deutschen Rechtschreibung
2. ein eingeführter Taschenrechner (Bei grafikfähigen Rechnern und Computeralgebrasystemen ist ein Reset durchzuführen.)
3. eine Liste der fachspezifischen Operatoren Fachbereich III

Sonstige Hinweise

keine

In jedem Fall vom Prüfling auszufüllen

| | |
|------------------------|----------------|
| Name: _____ | Vorname: _____ |
| Prüferin/Prüfer: _____ | Datum: _____ |

Impfzentrum

Aufgaben

Für das Management eines kommunalen Impfzentrums sollen eine Verwaltungsapplikation und eine Datenbank entwickelt werden. Das Einwohnermeldeamt stellt in Kooperation mit den Krankenkassen die Daten aller Personen zur Verfügung. Alle Personen ab dem 16. Lebensjahr sollen geimpft werden.

- 1 Objektorientierte Entwicklung des Impfzentrum-Systems
Ein erstes UML-Klassendiagramm finden Sie in Material 1.
- 1.1 Die Daten der zu impfenden Personen werden dem Impfzentrum in Form einer CSV¹-Datei zur Verfügung gestellt. Ein Datensatz hat folgenden Aufbau:
`77665;Müller;Klaus;31.12.1980;35334;Hausen;Teichweg 8;1`
Die einzelnen Elemente stehen für Personen-ID, Nachname, Vorname, Geburtsdatum, PLZ, Ort, Straße mit Hausnummer und Risikogruppe (1 ist Risikogruppe, 0 ist keine Risikogruppe). Überführen Sie die Klasse `Person` (Material 1) in Anweisungen einer objektorientierten Programmiersprache und implementieren Sie den Konstruktor ohne die weiteren Methoden.
Hinweise: Der Konstruktor erhält die Personendaten als Zeichenkette wie oben beschrieben. Das Geburtsdatum muss mithilfe der Methode `pruefeDatum()` der Klasse `ImpfzentrumVerwaltung` auf Plausibilität getestet werden, bevor ein `Datums`-Objekt erzeugt wird. Die Dokumentationen der Klassen `Date` und `String` sind in Material 2 zu finden.
(6 BE)
- 1.2 Die Methode `pruefeDatum()` der Klasse `ImpfzentrumVerwaltung` (Material 1) gibt `true` zurück, wenn es sich bei den übergebenen Werten um ein plausibles Datum handelt, ansonsten `false`. Ein Datum ist hier nur gültig, wenn die Jahreszahl im Bereich von 1910 bis 2005 liegt. Entwickeln und zeichnen Sie ein Struktogramm für diese Methode.
Hinweis: Ein Jahr ist ein Schaltjahr, wenn die Jahreszahl durch 4 teilbar ist. Jahreszahlen, die durch 100 teilbar sind, sind allerdings keine Schaltjahre, es sei denn, die Jahreszahl ist durch 400 teilbar.
(7 BE)
- 1.3 Die Methode `einladen(datum: Date)` der Klasse `ImpfzentrumVerwaltung` (Material 1) generiert eine Liste mit Personen, die zu dem übergebenen Datum zur Impfung eingeladen werden sollen. Folgende Prioritäten sind bei der Erstellung der Liste zu berücksichtigen:
 - Erste Priorität haben Personen, deren 1. Impfung zum übergebenen Datum mehr als zwanzig Tage zurückliegt.
 - In zweiter Priorität folgen alle weiteren Personen nach Alter und gegebenenfalls Risikogruppenangehörigkeit absteigend sortiert. Handelt es sich um einen Risikopatienten, wird das Alter mit 2 multipliziert.Es muss sichergestellt werden, dass die Impfkapazität eines Tages (Anzahl der Impfstraßen mal Personen am Tag je Impfstraße) nicht überschritten wird. Implementieren Sie die Methode.
Hinweis: Die Dokumentationen der Klassen `Date` und `List` sind in Material 2 zu finden.
(10 BE)

¹ CSV steht für Comma-separated values und beschreibt den Aufbau einer Textdatei zur Speicherung einfach strukturierter Daten.

- 1.4 Für die tägliche Abwicklung der Impfungen werden die zu impfenden Personen jeder Impfstraße mithilfe jeweils einer Warteschlange verwaltet (Material 1). Eine Person wird bei ihrer Anmeldung im Impfzentrum am Ende der Warteschlange einer Impfstraße hinzugefügt. Nach dem Impfvorgang wird die Person abgemeldet und dann von der ersten Position der Warteschlange entfernt. Es kann erforderlich sein, dass eine Person aus der Warteschlange gelöscht werden muss, obwohl sie nicht an erster Position der Warteschlange ist. Das ist zum Beispiel der Fall, wenn die Person ihr Einverständnis verweigert oder die Ärztin bzw. der Arzt (im Folgenden Arzt genannt) eine Impfuntauglichkeit festgestellt hat.

- 1.4.1 Bei den Datenstrukturen Warteschlange und Stapel handelt es sich um besondere Formen der verketteten Liste. Beschreiben Sie die wesentlichen Eigenschaften der Datenstrukturen verkettete Liste, Warteschlange sowie Stapel und erläutern Sie deren Umsetzung in einer objektorientierten Programmiersprache unter Berücksichtigung benötigter Operationen.

Hinweis: Es sind nur einfach verkettete Liste zu berücksichtigen.

(6 BE)

- 1.4.2 Implementieren Sie die Klasse `Warteschlange` aus Material 1.

Hinweis: Die Methode `hinzufuegen()` fügt die Person an das Ende der Warteschlange hinzu. Die Methode `entfernen()` entnimmt die am längsten wartende Person. Die Methode `loeschen()` sucht eine bestimmte Person in der Warteschlange und kettet sie aus.

(12 BE)

- 1.5 Das System des Impfzentrums soll zukünftig auch die Mitarbeiterschaft der Impfteams verwalten. Impfteams bestehen aus Empfangskräften und medizinischen Fachkräften. Zu den medizinischen Fachkräften gehören zum Beispiel Pflegekräfte und Ärzte. Für die Aufgaben der Empfangskräfte des Impfzentrums wird Ihnen ein UML-Anwendungsfalldiagramm (Material 3) vorgelegt.

- 1.5.1 Erläutern Sie den Zweck von Anwendungsfalldiagrammen. Beschreiben Sie die wesentlichen Notationselemente des vorliegenden UML-Anwendungsfalldiagramms sowie die Unterschiede zwischen extend- und include-Beziehungen.

(5 BE)

- 1.5.2 Folgende weitere Anwendungsfälle sollen vom System bereitgestellt werden:

- Medizinische Fachkräfte führen Impfungen durch. Sie melden die Personen nach Abschluss der Impfung ab.
- Sie tragen die Impfung in den Impfpass ein, wenn einer vorliegt.
- Wenn kein Impfpass vorliegt, drucken sie eine Impfbescheinigung aus.
- Die Impftauglichkeit einer Person wird von einem Arzt geprüft.
- Der Arzt dokumentiert eventuelle Impfreaktionen.

Entwickeln und zeichnen Sie für diese Anwendungsfälle ein Anwendungsfalldiagramm in UML-Notation.

Hinweis: Das UML-Anwendungsfalldiagramm in Material 3 ist entsprechend zu ergänzen.

(6 BE)

- 1.5.3 Das UML-Klassendiagramm in Material 1 soll um das Personal und die Anwendungsfälle aus Aufgabe 1.5.2 erweitert werden. Folgende weitere Festlegungen sind zu berücksichtigen:
- Alle Mitarbeiter und Mitarbeiterinnen (im Folgenden Mitarbeiter genannt) sind Personen, die geimpft werden können. Jeder Mitarbeiter hat eine eindeutige Personalnummer.
 - Mitarbeiter haben Einsätze, die mit dem Datum zu speichern sind.
 - Bei den Mitarbeitern handelt es sich entweder um Empfangskräfte oder medizinische Fachkräfte. Empfangskräfte erhalten eine Aufwandentschädigung. Medizinische Fachkräfte verfügen über eine Qualifikation.
 - Ärzte sind spezielle Fachkräfte mit einer bestimmten Fachrichtung (z.B. Allgemeinmedizin).
 - Die Impfreaktion einer Person und der für die Impfung verantwortliche Arzt sind im System festzuhalten.

Modellieren und zeichnen Sie das erweiterte UML-Klassendiagramm inklusive der erforderlichen Konstruktoren und Methoden.

Hinweise: Die Sichtbarkeit von Methoden und Attributen sowie set- und get-Methoden müssen nicht dargestellt werden. Datentypen, Übergabeparameter und Rückgabetyper sind aufzuführen.

(8 BE)

2 Entwicklung einer Datenbank für das Impfzentrum

In einer ersten Version der Datenbank möchte das Impfzentrum die Daten der zu impfenden Personen verwalten. Ebenso sollen die Daten der Impfstoffe und ihrer Lagerstandorte, der Produzenten sowie der Impfstofflieferungen verwaltet werden. Dabei gilt:

Eine Person wird zweimal geimpft, wobei zwischen den Impfterminen drei Wochen liegen sollen. Es muss sichergestellt sein, dass eine Person bei beiden Impfungen den gleichen Impfstoff erhält.

Ein Entity-Relationship-Modell (ERM) ist in Material 4 dargestellt.

2.1 Das ERM soll in das relationale Modell einer Datenbank transformiert werden.

2.1.1 Beschreiben Sie das relationale Datenbankmodell. Nennen Sie mindestens einen Vor- und einen Nachteil.

(4 BE)

2.1.2 Überführen Sie das ERM (Material 4) in das relationale Modell in der 3. Normalform und begründen Sie Ihre Vorgehensweise.

Hinweis: Alle Relationen sind in der Schreibweise `Relation(PK, Attribut, ..., FK#)` anzugeben.

(8 BE)

2.2 Einige Daten der Datenbank sollen ergänzt, geändert und ausgewertet werden.

2.2.1 Formulieren Sie eine SQL-Anweisung, die alle Personen mit Personen-ID, Vor- und Nachnamen sowie dem Termin der ersten Impfung auflistet, die bisher nur einmal mit dem Impfstoff mit der Bezeichnung BT223a2 geimpft wurden.

(3 BE)

- 2.2.2 Implementieren Sie eine SQL-Anweisung, die alle Impfstoffgruppen mit der Anzahl der im 1. Quartal 2021 zum zweiten Mal erfolgten Impfungen, absteigend sortiert nach der Anzahl, ausgibt.

(3 BE)

- 2.2.3 Der kanadische Impfstoffproduzent PharmaCorn (ID 96) in Vancouver (Postleitzahl BC V6T 1Z3) liefert am 26.04.2021 erstmalig 2500 Dosen des Impfstoffs BT223a2. Entwickeln Sie die SQL-Anweisungen, um die Datenbank zu ergänzen und die Lagermenge zu aktualisieren.

Hinweise: Es existieren bereits mehrere Lagerstandorte für den Impfstoff BT223a2. Die 2500 neuen Dosen können nur eingelagert werden, wenn an einem Standort genügend Platz vorhanden ist. Sie können davon ausgehen, dass es mindestens einen Standort gibt, der die gelieferte Menge aufnehmen kann. Der Standort mit der geringsten Anzahl an BT223a2-Impfdosen ist als Lagerstandort zu wählen.

(6 BE)

- 2.2.4 Für die Terminplanung soll eine nach Impfstoffen gruppierte Aufstellung mit der jeweiligen Anzahl an vorrätigen Dosen und der Anzahl der noch für die 2. Impfung benötigten Dosen in der folgenden Form ausgegeben werden:

| Impfstoff | Vorrätige Dosen | Benötigte Dosen |
|-----------|-----------------|-----------------|
| ApolloVac | 668 | 420 |
| BT223a2 | 488 | 620 |
| Vec23.Cov | 52 | 102 |

Entwickeln Sie eine entsprechende SQL-Anweisung.

(4 BE)

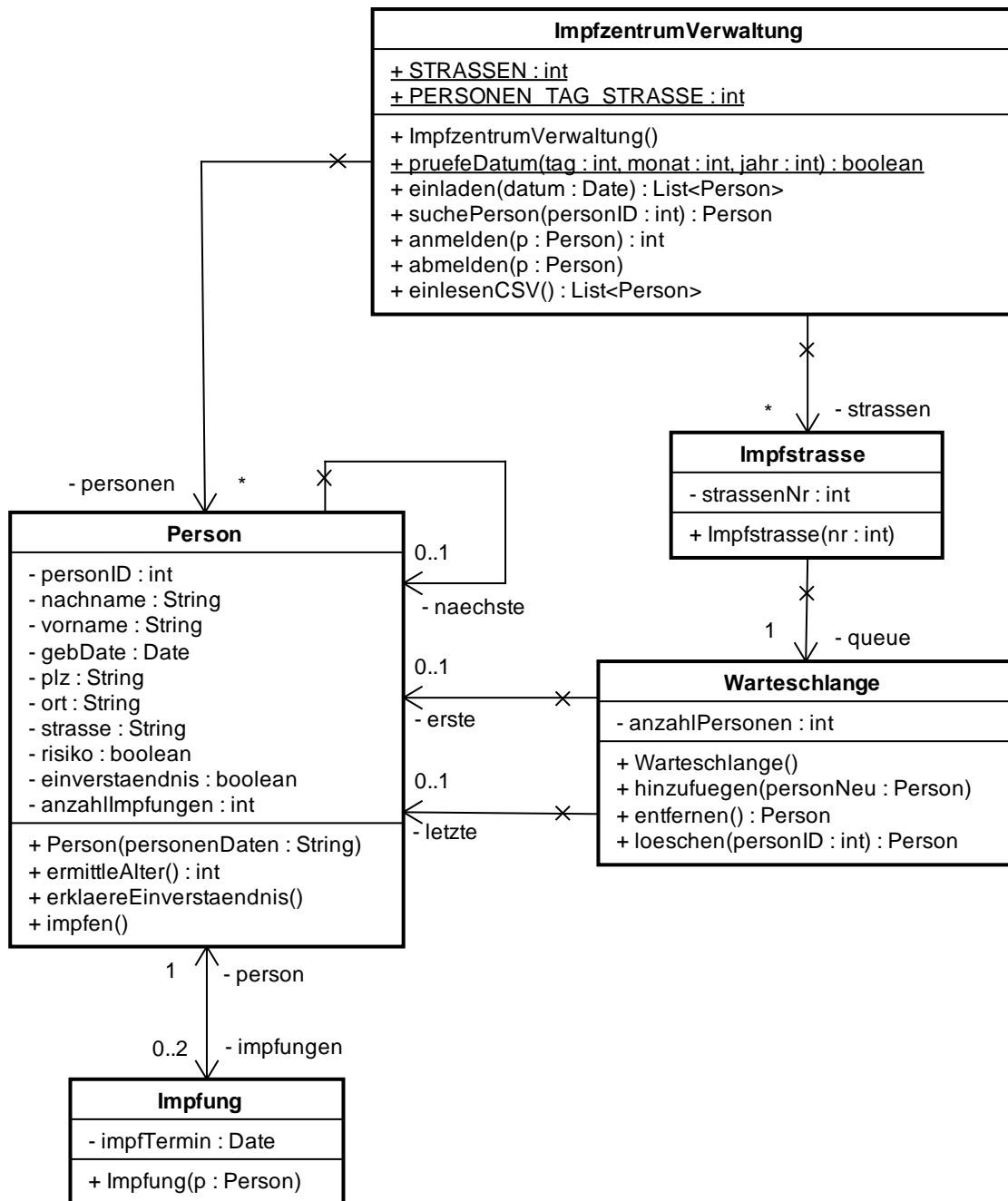
- 2.3 Ärzte, die die Impfungen begleiten, sind verpflichtet, Daten zu den geimpften Personen einzuholen und für Auswertungen zu protokollieren. In Material 5 ist ein Ausschnitt der Erhebung zu finden.
Analysieren Sie die Struktur der Tabelle hinsichtlich eventuell auftretender Probleme.
Überführen Sie die Tabelle in die 3. Normalform und begründen Sie Ihre Vorgehensweise.

Hinweis: Alle Relationen sind in der Schreibweise `Relation(PK, Attribut, ..., FK#)` anzugeben.

(12 BE)

Material 1

UML-Klassendiagramm Impfzentrum



Hinweise: Die Konstante STRASSEN der Klasse ImpfzentrumVerwaltung legt die Anzahl der Impfstraßen fest, die Konstante PERSONEN_TAG_STRASSE die maximale Anzahl an Impfungen je Impfstraße und Tag. Die Methode anmelden() der Klasse ImpfzentrumVerwaltung sucht die Impfstraße mit der geringsten Anzahl wartender Personen, fügt die übergebene Person der entsprechenden Warteschlange hinzu und gibt die Nummer der Impfstraße zurück. Die Methode abmelden() entfernt die Person nach Abschluss des Impfvorgangs aus der entsprechenden Warteschlange. Der Impftermin der Klasse Impfung ist das aktuelle Datum bei der Objekterzeugung.

Material 2

Klassendokumentationen

Klasse Date

`Date()`

erzeugt ein `Date`-Objekt mit dem aktuellen Systemdatum.

`Date(day: int, month: int, year: int)`

erzeugt ein `Date`-Objekt mit den Werten von `day`, `month` und `year`.

`addDays(numberOfDays: int): Date`

liefert eine Kopie des `Date`-Objekts mit der addierten Anzahl von Tagen.

`isBefore(d: Date): boolean`

liefert `true`, wenn das `Date`-Objekt vor dem des Parameters `d` liegt.

`isAfter(d: Date): boolean`

liefert `true`, wenn das `Date`-Objekt nach dem des Parameters `d` liegt.

`isBetween(d1: Date, d2: Date): boolean`

liefert `true`, wenn das `Date`-Objekt zwischen den Parametern `d1` und `d2` liegt.

`toString(): String`

liefert eine String-Repräsentanz des `Date`-Objekts im Format `dd.mm.yyyy`.

| Date |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| + Date() + Date(day : int, month : int, year : int) + addDays(numberOfDays : int) : Date + isBefore(d : Date) : boolean + isAfter(d : Date) : boolean + isBetween(d1 : Date, d2 : Date) : boolean + toString() : String |

Klasse List

`List<T>()`

erzeugt eine generische Liste mit Elementen des Typs `T`.

`add(obj: T)`

hängt das Objekt `obj` vom Typ `T` am Ende der Liste an.

`add(index: int, obj: T)`

fügt das Objekt `obj` vom Typ `T` an der Position `index` in die Liste ein.

`contains(obj: T): boolean`

liefert `true`, wenn das Objekt `obj` in der Liste enthalten ist, ansonsten `false`.

`get(index: int): T`

liefert das Listenelement an der Position `index` zurück bzw. `null`, falls `index` negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

`remove(index: int): T`

entfernt das Objekt vom Typ `T` an der Position `index` aus der Liste und gibt es zurück.

`remove(obj: T): boolean`

entfernt das Objekt `obj` aus der Liste. Falls `obj` mehrmals in der Liste enthalten ist, wird nur das erste Vorkommen entfernt. Der Rückgabewert ist `true`, falls das Objekt gefunden und entfernt wurde, sonst `false`.

`size(): int`

liefert die Anzahl der Elemente in der Liste zurück.

| List<T> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| + List<T>() + add(obj : T) + add(index : int, obj : T) + contains(obj : T) : boolean + get(index : int) : T + remove(index : int) : T + remove(obj : T) : boolean + size() : int |

Material 2 (Fortsetzung)**Klasse `String`**

`equals(str: String): boolean`

liefert `true`, wenn beide Strings gleich sind, ansonsten `false`.

`split(str: String): String[]`

teilt einen String am Trennzeichen `str`. Die Teil-Strings werden in einem Feld zurückgeliefert.

`startsWith(str: String): boolean`

liefert `true`, wenn der String mit `str` beginnt, ansonsten `false`.

String

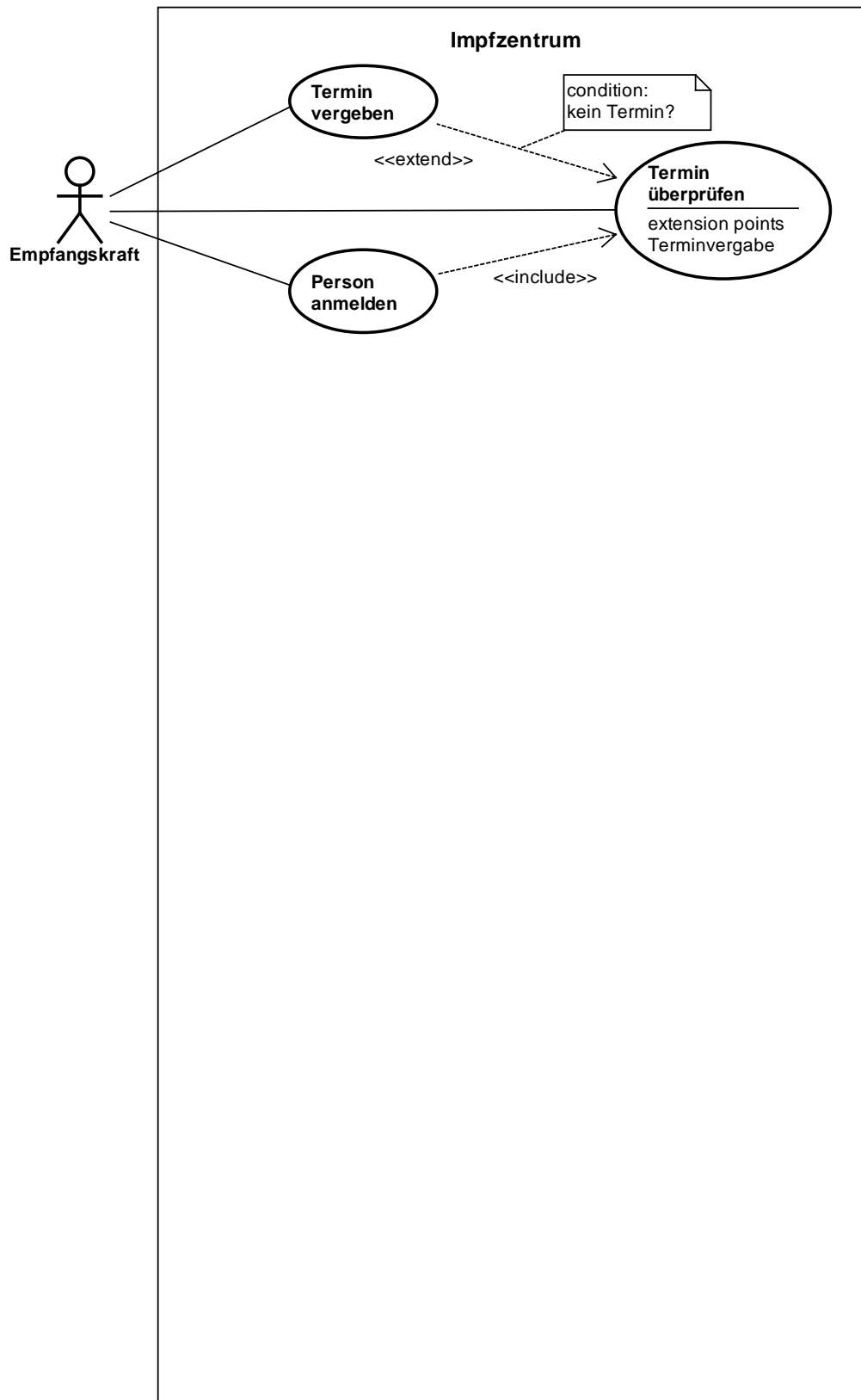
+ `equals(str : String) : boolean`

+ `split(str : String) : String[]`

+ `startsWith(str : String) : boolean`

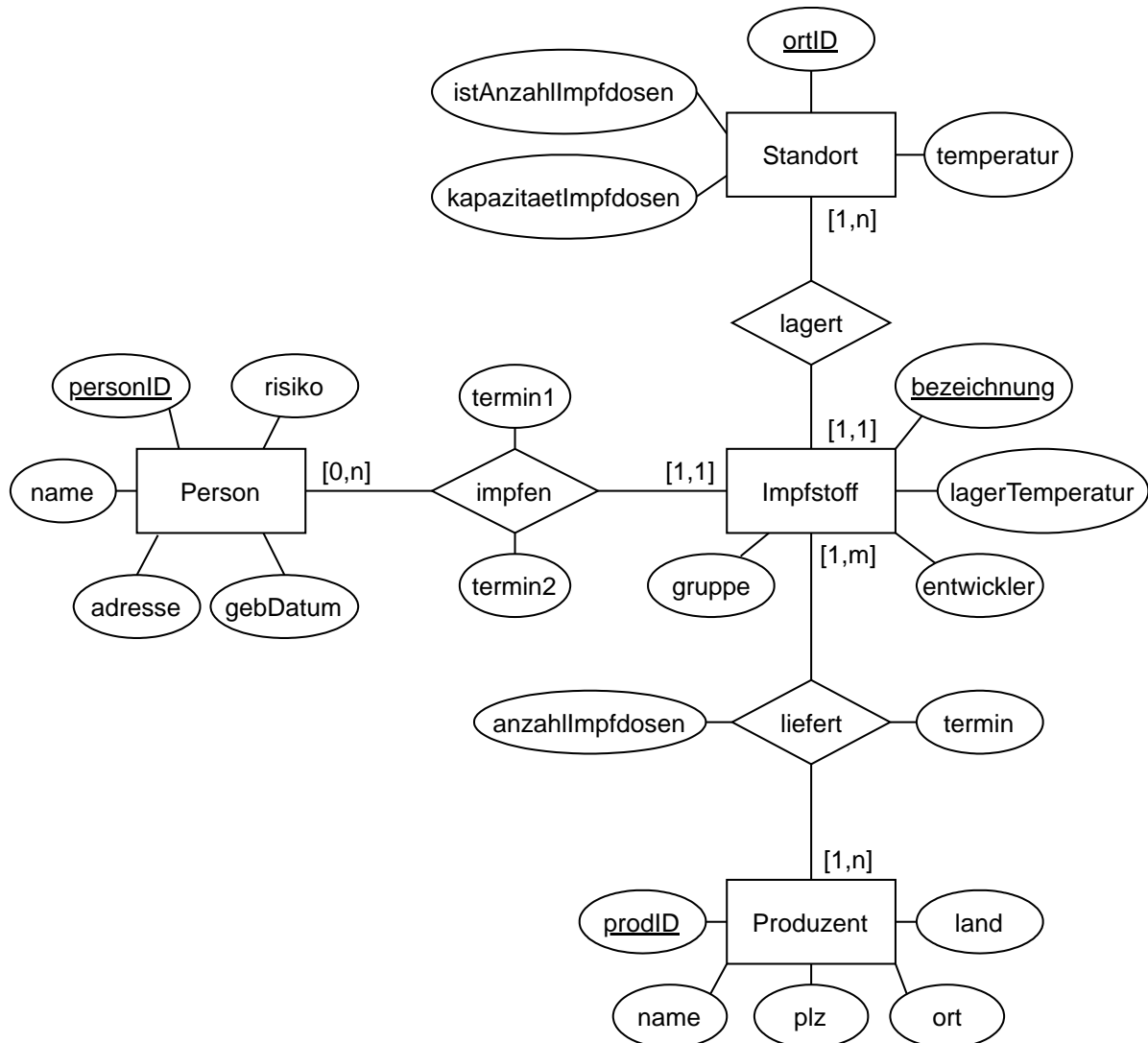
Material 3

UML-Anwendungsfalldiagramm



Material 4

Entity-Relationship-Modell



Hinweise: Ein Impfstoff wird von einem Forschungsinstitut entwickelt (Attribut entwickler). Die Herstellung und Lieferung eines Impfstoffs erfolgt durch Produzenten. Ein Impfstoff gehört zu einer Impfstoffgruppe (Attribut gruppe), wie zum Beispiel mRNA-Impfstoff oder Vektor-Impfstoff. Ein Impfstoff muss bei einer gewissen Temperatur (z.B. -70°C) gelagert werden (Attribut lagerTemperatur). Die verschiedenen Lagerstandorte weisen durch eine entsprechende Kühlung eine jeweils konstante Temperatur (Attribut temperatur) auf.

Material 5

Tabelle Impfprotokoll (Ausschnitt)

| Impfiling | Risikofaktoren | Impfstoff | Impf-Termin | Impfreaktionen | Verantwortlich |
|------------------------------------------------|----------------------------------------|-----------------------------------------------|-------------|----------------------------------------------------------------|-------------------------------------------------------|
| Nr. 123456 Ilka Meier geb. 12.04.1937 | Diabetes Übergewicht | BT223a2 (Gruppe mRNA- Impfstoff) | 07.04.21 | keine | Kürzel HEIL Dr. Silke Heiler (Allgemeinmedizin) |
| Nr. 776311 Karl Huber geb. 08.05.1943 | Bluthochdruck Übergewicht Rheuma | BT223a2 (Gruppe mRNA- Impfstoff) | 07.04.21 | Schwindel, Übelkeit Muskelschmerzen allergische Reaktion | Kürzel BADE Kai Bader (Allgemeinmedizin) |
| Nr. 270887 Vera Stark geb. 23.11.1950 | keine | BT223a2 (Gruppe mRNA- Impfstoff) | 07.04.21 | Schmerzen an der Injektionsstelle | Kürzel WIRT Dr. Lena Wirt (Innere Medizin) |
| ... | ... | ... | ... | ... | ... |
| Nr. 620087 Ina Kraus geb. 16.09.1948 | keine | Vec23.Cov (Gruppe Vektor- impfstoff) | 08.04.21 | Fieber Kopfschmerzen Appetitlosigkeit Gelenkschmerzen | Kürzel KLEB Dr. Kurt Kleber (Internist) |
| ... | ... | ... | ... | ... | ... |
| Nr. 221009 Lars Bauer geb. 16.06.1954 | Diabetes | BT223a2 (Gruppe mRNA- Impfstoff) | 09.04.21 | keine | Kürzel REIC Prof. Dr. Til Reich (Orthopäde) |
| ... | ... | ... | ... | ... | ... |
| Nr. 123456 Ilka Meier geb. 12.04.1937 | Diabetes Übergewicht | BT223a2 (Gruppe mRNA- Impfstoff) | 29.04.21 | Kopfschmerzen Gelenkschmerzen | Kürzel WIRT Dr. Lena Wirt (Innere Medizin) |
| Nr. 620087 Ina Kraus geb. 16.09.1948 | keine | Vec23.Cov (Gruppe Vektor- Impfstoff) | 29.04.21 | Müdigkeit Kopfschmerzen | Kürzel KLEB Dr. Kurt Kleber (Innere Medizin) |
| ... | ... | ... | ... | ... | ... |