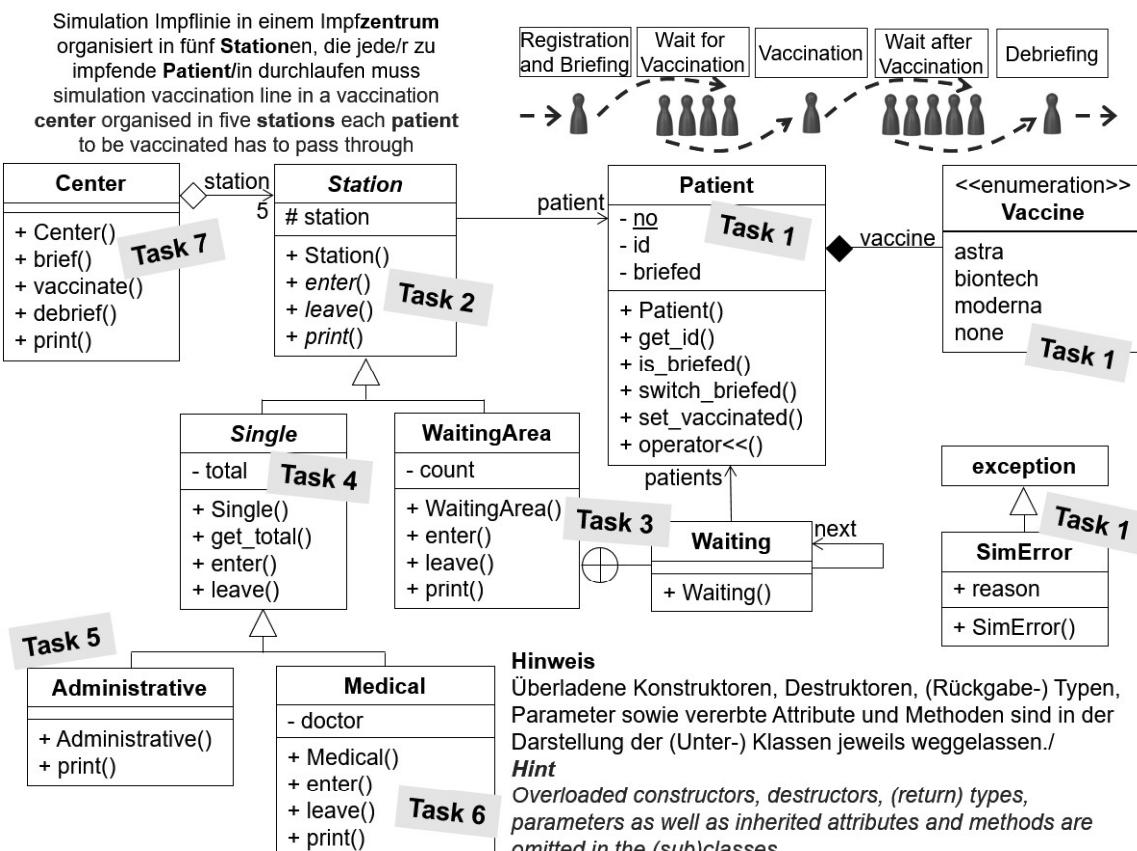


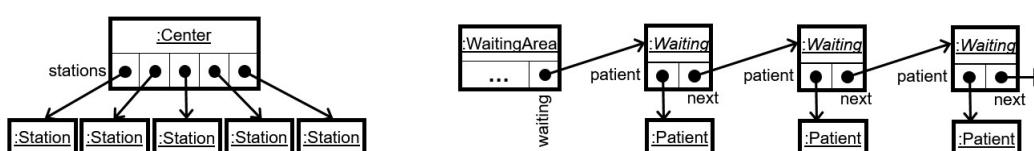
Einführende Erläuterungen/Introductory Explanations

Aufgabe ist eine vereinfachte Simulation eines Corona-Impfzentrums (**Center**) mit fünf **Stationen** in einer Impfstrasse. Drei verschiedene Impfstoffe (**Vaccine**) werden verimpft. Im UML-Modell unten ist eine vereinfachte Klassenhierarchie für **Stationen** gegeben mit administrativen (**Administrative**) und medizinischen (**Medical**) Einzel-Stationen (**Single**) sowie für Wartezeichen (**WaitingArea**). Eine **Station** wird in der Simulation von jeweils einer/einem **Patientin** betreten (**enter**) und verlassen (**leave**). Bei Einzelstationen ist dies immer ein- und derselbe Patient. Bei Warteräumen betritt ein Patient/eine Patientin diesen und wird am Ende einer Liste Wartender (**Waiting**) angefügt, während immer die/der erste der Liste den Warteraum wieder verlässt. Fehler in der Simulation werden durch Ausnahmeobjekte der Klasse **SimError** repräsentiert.

Task is a simplified simulation of a Corona vaccination **Center** with five **Stations** of a vaccination line. Three different **Vaccines** are vaccinated. In the UML model below a simplified inheritance hierarchy for **Stations** are given with **Administrative** and **Medical Single** stations as well as **WaitingAreas**. Within the simulation a **Patient** enters and leaves a **Station**. In **Single** stations it is always the same patient entering and leaving. In waiting areas a patient enters and gets appended at the end of a list of **Waiting** patients, whereas always the first of the list **leaves** the waiting area. Errors during the simulation are represented by exception objects of type **SimError**.



Beispiel Impfzentrum und Liste Wartende/Example vaccination center and list waitings



Beispiel-Programmlauf für Matrikelnr. 1234567/ Example Program for matriculation no. 1234567 Run

```
-----  
LINE DATA  
Registration and Briefing: 0 patients processed  
Wait for Vaccination: 0 waiting patients:  
Vaccination: 0 patients vaccinated by <your name here>  
Wait after Vaccination: 0 waiting patients:  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> d  
simulation error: no patient waiting in Wait after Vaccination  
-----  
LINE DATA  
Registration and Briefing: 0 patients processed  
Wait for Vaccination: 0 waiting patients:  
Vaccination: 0 patients vaccinated by <your name here>  
Wait after Vaccination: 0 waiting patients:  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> b  
patient 1234567 enters Registration and Briefing  
patient 1234567 briefed leaves Registration and Briefing  
-----  
LINE DATA  
Registration and Briefing: 1 patients processed  
Wait for Vaccination: 1 waiting patients: 1234567  
Vaccination: 0 patients vaccinated by <your name here>  
Wait after Vaccination: 0 waiting patients:  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> b  
patient 1234568 enters Registration and Briefing  
patient 1234568 briefed leaves Registration and Briefing  
-----  
LINE DATA  
Registration and Briefing: 2 patients processed  
Wait for Vaccination: 2 waiting patients: 1234567 1234568  
Vaccination: 0 patients vaccinated by <your name here>  
Wait after Vaccination: 0 waiting patients:  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> c  
patient 1234567 briefed enters Vaccination  
patient 1234567 vaccinated with AstraZeneca leaves Vaccination  
-----  
LINE DATA  
Registration and Briefing: 2 patients processed  
Wait for Vaccination: 1 waiting patients: 1234568  
Vaccination: 1 patients vaccinated by <your name here>  
Wait after Vaccination: 1 waiting patients: 1234567  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> b  
patient 1234569 enters Registration and Briefing  
patient 1234569 briefed leaves Registration and Briefing  
-----  
LINE DATA  
Registration and Briefing: 3 patients processed  
Wait for Vaccination: 2 waiting patients: 1234568 1234569  
Vaccination: 1 patients vaccinated by <your name here>  
Wait after Vaccination: 1 waiting patients: 1234567  
Debriefing: 0 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> d  
patient 1234567 vaccinated with AstraZeneca enters Debriefing  
patient 1234567 vaccinated with AstraZeneca leaves Debriefing  
-----  
LINE DATA  
Registration and Briefing: 3 patients processed  
Wait for Vaccination: 0 waiting patients:  
Vaccination: 3 patients vaccinated by <your name here>  
Wait after Vaccination: 1 waiting patients: 1234569  
Debriefing: 2 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
> d  
patient 1234569 vaccinated with AstraZeneca enters Debriefing  
patient 1234569 vaccinated with AstraZeneca leaves Debriefing  
-----  
LINE DATA  
Registration and Briefing: 3 patients processed  
Wait for Vaccination: 0 waiting patients:  
Vaccination: 3 patients vaccinated by <your name here>  
Wait after Vaccination: 0 waiting patients:  
Debriefing: 3 patients processed  
-----  
SIMULATION VACCINATION CENTER  
(a) end  
(b) new arrival at center  
(c) next vaccination  
(d) next debriefing  
>
```

Hinweis: Im nachfolgenden deutschen Text werden gendergerecht teils weibliche teils männliche Formen verwendet, diese sind immer gleichbedeutend (Beispiel Patientin/Patient).

1 Aufzählung, Ausnahmeklasse und Klasse Patient/ Enumeration, Exception Class and Class Patient (23.5 P.)

1.1 Aufzählungsklasse/Enumeration Class (1.0 P.)

Definieren Sie eine Aufzählung als Struktur nach Standard C++11 (oder neuer) mit Namen **Vaccine** (Impfstoff) mit vier Aufzählungswerten **astra**, **biontech**, **moderna** und **none** (keiner).

Define an enumeration as structure following standard C++11 (or newer) with name **Vaccine** with four enumeration values **astra**, **biontech**, **moderna** and **none**.

1.2 Ausgabe-Operator/Output Operator (4.5 P.)

Definieren Sie einen überladenen binären Ausgabeoperator << für ein Objekt der Aufzählung **Vaccine** aus Teilaufgabe 1.1 auf einen Ausgabe-Zeichen-Strom. Im Rumpf soll je nach Aufzählungswert die entsprechende Zeichenkette **AstraZeneca**, **BioNTech**, **Moderna** oder **none** auf den als Parameter gegebenen Zeichen-Ausgabe-Strom geschrieben werden.

Define an overloaded binary output operator << for an object of enumeration **Vaccine** from subtask 1.1 onto a character output stream given as parameter. In its body regarding the enumeration value the respective string **AstraZeneca**, **BioNTech**, **Moderna** or **none** shall be written onto the character output stream given as parameter.

1.3 Klasse/Class SimError (2.5 P.)

Definieren Sie eine Klasse mit Namen **SimError** abgeleitet von der vordefinierten Standardklasse für Ausnahmen. Die Klasse soll als öffentliche Member ein C++-Zeichenketten-Attribut mit Name **reason** (Begründung) haben und einen Konstruktor mit einer C++-Zeichenkette als Parameter zur Initialisierung des Attributs.

Define a class with name **SimError** inherited from the predefined standard exception class. The class shall have as public members a C++ string attribute with name **reason** and a constructor with a C++ string as parameter initialising the attribute.

1.4 Klasse/Class Patient (15.5 P.)

Definieren Sie eine Klasse mit Namen **Patient** mit folgenden Bestandteilen:

Define a class with name **Patient** with following elements:

- private ganzzahlige Klassenvariable mit Namen **no**, über die fortlaufende IDs erzeugt werden. Initialisieren Sie diese an geeigneter Stelle im Kode mit Ihrer Matrikelnummer.
private integer class variable with name **no** used to generate sequential IDs. At an appropriate place in code initialise it with your matriculation number.
- privates konstantes ganzzahliges Attribut mit Namen **id** für eine eindeutige Patienten-ID.
private constant integer attribute with name **id** for a unique patient ID.
- privates Boolesches Attribut mit Namen **briefed** (informiert über Nebenwirkungen).
private Boolean attribute with name **briefed** (about side effects).
- privates Attribut mit Namen **vaccine** (Impfstoff) vom Aufzählungstyp in Teilaufgabe 1.1.
private attribute with name **vaccine** of the enumeration type in subtask 1.1.
- öffentliche Deklarationen (also nur die Funktionsprototypen) von:
public declarations (only the function prototypes) of:
 - Standardkonstruktor.
standard constructor.
 - Member-Funktion mit Namen **get_id** ohne Parameter und mit ganzzahligem Rückgabewert.
member function with name **get_id** without parameter and with integer return value.

- Member-Funktion mit Namen **is_briefed** ohne Parameter und Booleschem Rückgabewert.
member function with name **is_briefed** without parameter and Boolean return value.
- Member-Funktion mit Namen **switch_briefed** ohne Parameter und ohne Rückgabewert.
member function with name **switch_briefed** without parameter and without return value.
- Member-Funktion mit Namen **set_vaccinated** (setze geimpft) mit einem Parameter vom Aufzählungstyp in Teilaufgabe 1.1 und ohne Rückgabewert.
member function with name **set_vaccinated** with a parameter of the enumeration type in subtask 1.1 and without return value.
- überladener befreundeter binärer Ausgabeoperator << für eine Referenz auf ein Objekt der Klasse auf einen Ausgabe-Zeichen-Strom.
an overloaded friend binary output operator << for a reference to an object of this class onto a character output stream.
- Definieren Sie ausserhalb der Klassendefinition von **Patient** den/die:
Define outside of the class definition of **Patient** the:
 - Standard-Konstruktor. Das Attribut **id** soll mit dem post-inkrementierten Wert der Klassenvariable **no** initialisiert werden, als nicht informiert und nicht geimpft (Wert **none** aus der Aufzählung).
standard constructor. Initialise attribute **id** with the post-incremented value of the class variable **no**, not briefed and not vaccinated (value **none** from the enumeration).
 - Member-Funktion **get_id**, die den Wert des entsprechenden Attributs zurück liefert.
member function with name **get_id** returning the value of the respective attribute.
 - Member-Funktion **is_briefed**, die den Wert des entsprechenden Attributs zurück liefert.
member function with name **is_briefed** returning the value of the respective attribute.
 - Member-Funktion **switch_briefed**, die den Wert des entsprechenden Attributs negiert.
member function with name **switch_briefed** negating the value of the respective attribute.
 - Member-Funktion **set_vaccinated**, die den Wert des entsprechenden Attributs auf den Wert des Parameters setzt.
member function with name **set_vaccinated** assigning the parameter value to the respective attribute.
 - Ausgabeoperator. Schreiben Sie die Zeichenkette **patient** und die Patienten-ID, falls diese dieser geimpft ist die Zeichenkette **vaccinated with** und den im Attribut gespeicherten Impfstoff oder falls nicht, aber bereits informiert, die Zeichenkette **briefed** auf den jeweiligen Zeichen-Ausgabe-Strom (siehe Beispieldaten auf Seite 2).
output operator. Write string **patient** and the patient's ID, if she/he is vaccinated string **vaccinated with** and the vaccine stored in the attribute value, or if not but already briefed string **briefed** onto the character output stream (see example outputs at page 2).

2 Klasse/Class Station (5.5 P.)

Definieren Sie eine abstrakte Klasse mit Name **Station** mit folgenden Mitgliedern:

Define an abstract class with name **Station** with following members:

- geschütztes Attribut mit Namen **station** vom Typ C++-Zeichenkette für die Stationsbezeichnung.
protected attribute with name **station** of type C++ string for the station description.
- geschütztes Zeigerattribut mit Namen **patient** vom Typ **Patient** für die/den aktuell durchgeschleuste Patientin/Patienten.
protected pointer attribute with name **patient** of type **Patient** for the actually passed through patient.
- öffentliche Definition eines Konstruktors mit einem C++-Zeichenketten-Parameter und einem Zeigerparameter auf einen Patienten, der die beiden Attribute mit den Parameterwerten initialisiert.
public definition of a constructor with a C++ string parameter and a pointer parameter to a patient initialising the two attributes with the parameter values.

-
- öffentliche Definition eines virtuellen Destruktors.
public definition of a virtual destructor.
 - rein virtuelle Member-Funktion mit Namen **enter** mit einem Zeigerparameter auf eine Patientin ohne Rückgabewert.
pure virtual member function with name **enter** with a pointer parameter to a patient and no return value.
 - rein virtuelle Member-Funktion mit Namen **leave** ohne Parameter und einem Zeiger auf einen Patienten als Rückgabe.
pure virtual member function with name **leave** without parameter and a pointer to a patient as return value.
 - rein virtuelle Member-Funktion **print** ohne Parameter und ohne Rückgabewert.
pure virtual member function with name **print** without parameter and no return value.

3 Klasse/Class WaitingArea (20.5 P.)

Definieren Sie eine Klasse mit Name **WaitingArea** (Warteraum) als direkte öffentliche UnterkLASSE der abstrakten Klasse **Station** mit folgenden Elementen:

Define a class with name **WaitingArea** as direct public subclass of abstract class **Station** with following elements:

- private Klassendefinition in der Klasse mit Namen **Waiting** (Wartend) mit folgenden Membern:
private class definition in class with name **Waiting** with following members:
 - öffentliches Zeigerattribut mit Namen **patient** vom Typ **Patient**.
public pointer attribute with name **patient** of type **Patient**.
 - öffentliches Zeigerattribut mit Namen **next** vom Typ **Waiting** zum Aufbau einer Liste von wartenden Patienten.
public pointer attribute with name **next** of type **Waiting** to build up a list of waiting patients.
 - öffentliche Definition eines Konstruktors mit einem Zeigerparameter auf einen Patienten, der das entsprechende Zeigerattribut initialisiert und den Zeiger **next** auf einen Nullzeiger initialisiert.
public definition of a constructor with a pointer parameter to a patient initialising the respective pointer attribute and pointer **next** to a null pointer.
- privates ganzzahliges Attribut mit Namen **count**, in der die Anzahl Wartender gezählt wird.
private integer attribute with name **count** for the number of waiting patients.
- privates Zeigerattribut mit Namen **waiting** vom Typ Zeiger auf **Waiting**, das den Kopf einer Liste Wartender speichert.
private pointer attribute with name **waiting** of type pointer to **Waiting** storing the head of a list of waiting patients.
- öffentliche Definition eines Konstruktors mit einer C++-Zeichenkette für die Bezeichnung des Warte- raums als Parameter, der das entsprechende Attribut mit dem Parameterwert initialisiert, das At- tribut **count** mit dem Wert 0 und den Kopf der Liste als Nullzeiger.
public definition of a constructor with one C++ string parameter for the description of the waiting area initialising the respective attribute with the parameter value, the attribute **count** with value 0 and the head of the list as null pointer.
- öffentliche Definition einer virtuellen Member-Funktion mit Namen **enter** ohne Rückgabewert und mit einem Zeigerparameter auf einen Patienten, der den Warteraum betritt. Im Rumpf soll der Patient am Ende der Liste der Wartenden eingefügt und die Anzahl **count** inkrementiert werden.
public definition of a virtual member function with name **enter** without return value and with a pointer parameter to a patient entering the waiting area. In the body the patient shall be appended at the end of the list of waiting and the attribute **count** be incremented.

- öffentliche Definition einer virtuellen Member-Funktion mit Namen **leave** ohne Parameter und einem Zeiger auf eine Patientin als Rückgabe. Ist die Liste Wartender leer, soll ein neues Ausnahmeobjekt der Klasse **SimError** (aus Teilaufgabe 1.3) mit den konkatenierten Zeichenketten **no patient waiting in** und der Stationsbezeichnung als Begründung geworfen werden. Ansonsten soll die erste Patientin der Liste zurück gegeben werden, zuvor das erste Listenelement gelöscht und die Anzahl **count** der Wartenden dekrementiert werden.
 public definition of a virtual member function with name **leave** without parameter and a pointer to a patient as return value. If the list of waiting patients is empty, a new exception object of class **SimError** (from subtask 1.3) shall be thrown with concatenated strings **no patient waiting in** and the station description as reason. Otherwise the first patient of the list shall be returned, before the first list element shall be deleted and variable **count** decremented.
- öffentliche Definition einer virtuellen Member-Funktion **print** ohne Parameter und ohne Rückgabewert. Im Rumpf sollen die Stationsbezeichnung, die Anzahl Wartender **count** und hinter der Zeichenkette **waiting patients:** die IDs aller Wartender auf den Standard-Zeichen-Ausgabe-Strom geschrieben werden (siehe Beispieldateien auf Seite 2).
 public definition of a virtual member function with name **print** without parameter and no return value. In the body the station description, how many waiting **count**, and after string **waiting patients:** the IDs of all waiting patients shall be written onto the standard character output stream (see example outputs at page 2).

4 Klasse/Class Single (8.0 P.)

Definieren Sie eine abstrakte Klasse mit Name **Single** (Einzelstation) als direkte öffentliche Unterklasse der abstrakten Klasse **Station** mit folgenden Mitgliedern:

Define an abstract class with name **Single** (single station) as direct public subclass of the abstract class **Station** with following members:

- privates ganzzahliges Attribut mit Name **total** zur Speicherung der Gesamtanzahl in der Station durchgeschleuster Patienten.
 private integer attribute with name **total** to store the total number of passed through patients of the station.
- öffentliche Definition eines Konstruktors mit einer C++-Zeichenkette für die Stationsbezeichnung als Parameter, der diese geeignet initialisiert und das Attribut **total** mit dem Wert 0.
 public definition of a constructor with one C++ string parameter for the station description appropriately initialising it and attribute **total** with value 0.
- öffentliche Definition einer Member-Funktion mit Namen **get_total**, die den Wert des entsprechenden Attributs zurück liefert.
 public definition of a member function with name **get_total** returning the value of the respective attribute.
- öffentliche Definition einer virtuellen Member-Funktion mit Namen **enter** ohne Rückgabewert und mit einem Zeigerparameter auf eine Patientin, die die Einzelstation betritt. Im Rumpf sollen auf den Standard-Zeichen-Ausgabe-Strom die Patientin, die Zeichenkette **enters** und die Stationsbezeichnung geschrieben werden, danach die als Parameter eintretende Patientin dem Attribut **patient** als aktuell durchgeschleust zugewiesen, das Attribut **total** inkrementiert sowie der Patientin eine Nachricht **switch_briefed** gesendet werden.
 public definition of a virtual member function with name **enter** without return value and with a pointer parameter to a patient entering the single station. In the body the patient, string **enters** and the station name shall be written onto the standard character output stream, the given parameter patient stored in **patient** as actually passed through, attribute **total** incremented as well as a message **switch_briefed** be sent to the patient, .

-
- öffentliche Definition einer virtuellen Member-Funktion mit Namen **leave** ohne Parameter und einem Zeiger auf einen Patienten als Rückgabe. Im Rumpf sollen der durchgeschleuste Patient, die Zeichenkette **leaves** und die Stationsbezeichnung auf den Standard-Zeichen-Ausgabe-Strom geschrieben und der aktuell durchgeschleuste Patient zurück gegeben werden.
public definition of a virtual member function with name **leave** without parameter and a pointer to a patient as return value. In the body the passed through patient, string **leaves** and the station description shall be written onto the standard character output stream and the actually passed through patient getting returned.

5 Klasse/Class Administrative (3.0 P.)

Definieren Sie eine Klasse mit Name **Administrative** als direkte öffentliche Unterklasse der abstrakten Klasse **Single** mit folgenden Mitgliedern:

Define a class with name **Administrative** as direct public subclass of the abstract class **Single** with following members:

- öffentliche Definition eines Konstruktors mit einer C++-Zeichenkette für die Stationsbezeichnung als Parameter, der diese geeignet initialisiert.
public definition of a constructor with one C++ string parameter for the station description appropriately initialising it.
- öffentliche Definition einer virtuellen Member-Funktion **print** ohne Parameter und ohne Rückgabewert. Im Rumpf sollen die Stationsbezeichnung, die Gesamtanzahl durchgeschleuster Patienten und dahinter die Zeichenkette **patients processed** auf den Standard-Zeichen-Ausgabe-Strom geschrieben werden (siehe Beispieldateien auf Seite 2).
public definition of a virtual member function with name **print** without parameter and no return value. In the body the station description, the total number of passed through patients and behind string **patients processed** be written onto the standard character output stream (see example outputs at page 2).

6 Klasse/Class Medical (8.5 P.)

Definieren Sie eine Klasse mit Name **Medical** als direkte öffentliche Unterklasse der abstrakten Klasse **Single** mit folgenden Mitgliedern:

Define a class with name **Medical** as direct public subclass of the abstract class **Single** with following members:

- privates Attribut mit Name **doctor** vom Typ C++-Zeichenkette für die Impfarztin/den Impfarzt.
private attribute with name **doctor** of type C++ string for the vaccinator.
- öffentliche Definition eines Konstruktors mit zwei C++-Zeichenketten als Parameter für die Stationsbezeichnung und die Impfarztin/den Impfarzt zur Initialisierung.
public definition of a constructor with two C++ string parameters for the station description and the vaccinator to initialise.
- öffentliche Definition einer virtuellen Member-Funktion mit Namen **enter** ohne Rückgabewert und mit einem Zeigerparameter auf einen Patienten, der die medizinische Station betritt. Ist dieser Patient nicht zuvor über mögliche Nebenwirkungen informiert worden, soll ein neues Ausnahmeobjekt der Klasse **SimError** (aus Teilaufgabe 1.3) mit der Zeichenkette **not briefed patient can not be vaccinated** als Begründung geworfen werden. Ansonsten soll die gleichnamige Member-Funktion der direkten Oberklasse aufgerufen werden und der Patient den Impfstoff **astra** erhalten.
public definition of a virtual member function with name **enter** without return value and with a

pointer parameter to a patient entering the medical station. If the patient is not briefed beforehand about possible side effects, a new exception object of class **SimError** (from subtask 1.3) shall be thrown with string **not briefed patient can not be vaccinated** as reason. Otherwise the same name method of the direct upper class shall be called and the patient shall get vaccine **astra**.

- öffentliche Definition einer virtuellen Member-Funktion mit Name **leave** ohne Parameter und einem Zeiger auf eine Patientin als Rückgabe. Im Rumpf soll die gleichnamige Member-Funktion der direkten Oberklasse geeignet aufgerufen werden.
public definition of a virtual member function with name **leave** without parameter and a pointer to a patient as return value. In the body the same name method of the direct upper class shall be called appropriately.
- öffentliche Definition einer virtuellen Member-Funktion **print** ohne Parameter und ohne Rückgabewert. Im Rumpf sollen die Stationsbezeichnung, die Gesamtanzahl durchgeschleuster Patienten und nach der Zeichenkette **patients vaccinated by** die Impfärztin/den Impfarzt auf den Standard-Zeichen-Ausgabe-Strom geschrieben werden (siehe Beispieldateien auf Seite 2).
public definition of a virtual member function with name **print** without parameter and no return value. In the body the station description, the total number of passed through patients and after string **patients vaccinated by** the vaccinator be written onto the standard character output stream (see example outputs at page 2).

7 Klasse/Class Center (13.5 P.)

Definieren Sie eine Klasse mit Name **Center** (Impfzentrum) mit folgenden Mitgliedern:

Define a class with name (vaccination) **Center** with following members:

- privates Feld von fünf Zeigern vom Typ **Station** mit Namen **station** für fünf aufeinanderfolgende Stationen einer Impfstrasse des Impfzentrums.
private array of five pointers of type **Station** with name **station** for five consecutive stations of a vaccination line of this vaccination center.
- öffentliche Definition eines Standard-Konstruktors, der aufeinanderfolgend folgende fünf Objekte in der Impfstrasse auf dem Heap erzeugt und deren Adressen in den Zeigern im Feld speichert:
public definition of a standard constructor consecutively generating following five objects of the vaccination line on heap and storing their addresses in pointers of the array:
 - Objekt der Klasse **Administrative** mit Stationsbezeichnung **Registration and Briefing** (Registrierung und Vorbesprechung).
object of class **Administrative** with station description **Registration and Briefing**.
 - Objekt der Klasse **WaitingArea** mit Stationsbezeichnung **Wait for Vaccination** (Warten auf Impfung).
object of class **WaitingArea** with station description **Wait for Vaccination**.
 - Objekt der Klasse **Medical** mit Stationsbezeichnung **Vaccination** und Ihrem Namen als Impfärztin/Impfarzt.
object of class **Medical** with station description **Vaccination** and your name as vaccinator.
 - Objekt der Klasse **WaitingArea** mit Stationsbezeichnung **Wait after Vaccination** (Warten nach der Impfung).
object of class **WaitingArea** with station description **Wait after Vaccination**.
 - Objekt der Klasse **Administrative** mit Stationsbezeichnung **Debriefing** (Nachbesprechung).
object of class **Administrative** with station description **Debriefing**.

- öffentliche Definition einer Member-Funktion mit Name **brief** (vorbesprechen), die einen Zeiger-parameter auf einen Patienten hat und nichts zurück liefert. Im Rumpf soll der als Parameter gegebene Patient die erste Station **Registration and Briefing** betreten (**enter**), dieser danach die Station wieder verlassen (**leave**) und die zweite Station **Wait for Vaccination** (Warten auf Impfung) betreten (**enter**).
 public definition of a member function with name **brief** with a pointer parameter to a patient returning nothing. In the body the as parameter given patient shall **enter** the first station **Registration and Briefing**, then **leave** the station and **enter** second station **Wait for Vaccination**.
- öffentliche Definition einer Member-Funktion mit Name **vaccinate** (impfen) ohne Parameter und ohne Rückgabewert. Im Rumpf soll eine Patientin die zweite Station **Wait for Vaccination** verlassen (**leave**), die dritte Station **Vaccination** betreten (**enter**), diese dann wieder verlassen (**leave**) und die vierte Station **Wait after Vaccination** betreten (**enter**).
 public definition of a member function with name **brief** without parameter and without return value. In the body a patient shall **leave** the second station **Wait for Vaccination**, **enter** the third station **Vaccination**, then **leave** it again and **enter** the forth station **Wait after Vaccination**.
- öffentliche Definition einer Member-Funktion mit Name **debrief** (vorbesprechen) ohne Parameter und ohne Rückgabewert. Im Rumpf soll ein Patient die vierte Station **Wait after Vaccination** verlassen (**leave**), die fünfte Station **Debriefing** betreten (**enter**), diese dann wieder verlassen (**leave**) und danach das Objekt vom Typ **Patient** gelöscht werden.
 public definition of a member function with name **debrief** without parameter and without return value. In the body a patient shall **leave** the forth station **Wait after Vaccination**, **enter** the fifth station **Debriefing**, then **leave** it again and afterwards the object of type **Patient** be deleted.
- öffentliche Definition einer Member-Funktion mit Name **print** ohne Parameter und ohne Rückgabewert. Im Rumpf soll die Zeichenkette **LINE DATA** auf den Standard-Zeichen-Ausgabe-Strom geschrieben werden und in einer Schleife jedem Objekt, auf das die Zeiger im Feld **station** zeigen, eine Nachricht **print** geschickt werden. Die gestrichelten Linien können ignoriert werden.
 member function with name **print** without parameter and without return value. In the body string **LINE DATA** shall be written onto the standard character output stream and in a loop a message **print** shall be sent to each object the pointers in array **station** point to. The dashed lines can be ignored.

8 main (7.5 P.)

Schreiben Sie die Funktion **main** mit folgenden Definitionen und Anweisungen:

Write function **main** with following definitions and statements:

- Definieren Sie ein Objekt mit Namen **c** vom Typ **Center**.
 Define an object with name **c** of type **Center**.
- Definieren Sie ein Objekt mit Namen **choice** vom Typ **char**.
 Define an object with name **choice** of type **char**.
- Definieren Sie eine do-while-Schleife, in deren Rumpf Sie einen try-catch-Block definieren.
 Senden Sie darin zuerst dem Objekt **c** eine Nachricht **print**.
 Programmieren Sie danach ein kleines Menü und folgende Funktionalität (siehe auch dazu die Beispielausgaben auf Seite 2):
 Define a do-while-loop, where in its body a try-catch-block is defined.
 Inside first send to object **c** a message **print**.

Program afterwards a small menu and following functionality (see also example outputs at page 2):

(a) **end**:

Beenden der Schleife.

end of the loop.

(b) **new arrival at center**:

ein neuer Patient soll als Objekt auf dem Heap erzeugt werden und geeignet Parameter einer Nachricht **brief** an das Objekt **c** sein.

a new patient shall be generated as object on heap and appropriately be parameter of a message **brief** to object **c**.

(c) **next vaccination**:

senden einer Nachricht **vaccinate** an das Objekt **c**.

send a message **vaccinate** to object **c**.

(d) **next debriefing**:

senden einer Nachricht **debrief** an das Objekt **c**.

send a message **debrief** to object **c**.

Definieren Sie am Ende des Schleifenrumpfs einen Ausnahmebehandler für ein möglicherweise geworfenes Ausnahmeobjekt vom Typ **SimError** (Teilaufgabe 1.3) und schreiben als Fehlerbehandlung nach der Zeichenkette **simulation error**: die im gefangenem Ausnahmeobjekt gespeicherte Begründung auf den Standard-Zeichen-Fehler-Ausgabe-Strom.

At the end of the body of the loop define an exception handler for a possibly thrown exception object of type **SimError** (subtask 1.3) and write as error handling after string **simulation error**: the reason stored in the caught exception object onto the standard character **error** output stream.

**VIEL ERFOLG!/
GOOD SUCCESS!**