

26. JANUAR 2021

PROGRAMMIEREN, ALGORITHMEN UND DATENSTRUKTUREN I, WS 2020/21 AUFGABENBLATT 5



5.1 Der Patient

Im letzten Aufgabenblatt dieses Semesters soll ein ganzes Krankenhaus samt Patienten modelliert und zum Leben erweckt werden. Zuallererst benötigen Sie hierfür die Möglichkeit, einzelne Patienten zu speichern.

Erstellen Sie hierfür eine Klasse **Patient**. Jeder Patient soll seinen Vor- sowie Nachnamen, das Geschlecht (**männlich**, **weiblich** oder **divers**), das Geburtsdatum (z.B. als string im Format "DD.MM.JJJJ"), die Diagnose und die benötigte Behandlungsdauer (in Tagen) als Attribut speichern. Für mögliche Namen liegt die **names.h** dem Aufgabenblatt bei. Die jeweilige Diagnose ergibt sich anhand eines **enums** und kann die Werte **simulant**, **infektion**, **fraktur** oder **geheilt** annehmen (oder mit entsprechenden englischen Begriffen). Achten Sie auf eine sinnvolle Aufteilung der Klasse auf Header- und Sourcefile (.h und .cpp).

Des Weiteren benötigt ein Patient mehrere Methoden:

- einen Konstruktor, der alle notwendigen Parameter empfängt und die Attribute initialisiert. Die Behandlungsdauer soll dabei nicht als Parameter übergeben werden, sondern ergibt sich anhand der Diagnose (2 Tage für einen Simulanten, 14 Tage zur Behandlung einer Infektion, 6 Wochen für eine Fraktur sowie 0 für geheilt). Sollte keine Diagnose erstellt worden sein, wählen Sie als default-Wert **simulant**.

- eine sinnvolle Ausgabe der Patientendaten.
- eine Möglichkeit zur Reduzierung der Restbehandlungsdauer.
- eine Möglichkeit zur Anpassung der Diagnose. Sollte hierdurch die Diagnose geändert werden, muss die Behandlung von vorne starten. Entsprechend ändert sich die Behandlungsdauer des Patienten.
- Getter und Setter, wo es Ihrer Meinung nach Sinn ergibt.

Stellen Sie sicher, dass die Const-Correctness in Ihrem Code gewahrt bleibt!

5.2 Das Krankenhaus

Die zweite Klasse, die es zu implementieren gilt, ist das Krankenhaus.

Ein Krankenhaus hat eine bestimmte Anzahl an Betten sowie eine aktuelle Patientenliste. Ferner wird die Möglichkeit benötigt, um einen neuen Patienten aufzunehmen (also der Patientenliste hinzuzufügen), sofern noch Betten verfügbar sind. Eine weitere Methode soll das Vergehen von Zeit simulieren: der Eingabeparameter in Tagen wird von allen verbleibenden Behandlungsdauern abgezogen. Erreicht die Behandlungsdauer eines Patienten 0, wird er als geheilt betrachtet. Erzeugen Sie zudem Konstruktoren sowie Getter/Setter nach eigenem Ermessen.

Erstellen Sie in der `main()` ein Krankenhaus mit 60 Betten und füllen Sie es mit mind. 61 zufällig generierten Patienten (zufällige Namen können Sie der `names.h` entnehmen, die Geburtstage müssen nicht perfekt korrekt gefüllt sein).

5.3 Statische Attribute

Um Patienten eindeutig zu identifizieren, soll jeder Patient mit einer fortlaufenden Patientennummer versehen werden. Ergänzen Sie daher die Klasse `Patient` um die jeweilige Patientennummer sowie um ein statisches Attribut, welches die zuletzt vergebene Patientennummer speichert. Im Konstruktor des Patienten soll dann die Nummer an den Patienten vergeben und die statische Variable angepasst (also um 1 erhöht) werden. Damit die Patientennummer immer 6-stellig ist, initialisieren Sie den Wert mit 100000. Um bei Interesse auf dieses statische Attribut zugreifen zu können, erzeugen Sie eine ebenso statische Get-Methode.

Nun ist es möglich, einzelne Patienten anhand ihrer Nummer aus der Patientenliste eines Krankenhauses wieder zu löschen. Schreiben Sie eine entsprechende Methode für das Krankenhaus (zur Erinnerung, mittels `vec.erase(vec.begin() + pos)` lässt sich aus dem Vektor `vec` das Element an der Position `pos` entfernen). Nutzen Sie diese Methode, um zukünftig alle Patienten zu entfernen, die als `geheilt` geführt werden.

(Edit: das Entfernen einzelner Elemente aus dem Vektor funktioniert allerdings nur, wenn Sie keine Attribute des Patienten auf const gesetzt hatten, sonst streikt die interne Verschiebe-Operation. Alternativ können Sie im Patientenregister auch nur mit Pointern auf Patienten arbeiten.)

5.4 Pointer

Zu guter Letzt benötigen wir eine Methode, um zwei Patienten miteinander zu vergleichen. Dies ist nützlich, sollte ein und der selber Patient in mehreren Krankenhäusern eingewiesen worden sein.

Schreiben Sie daher eine statische Methode der Klasse `Patient`, die zwei Pointer auf jeweils einen Patienten erhält und zurückliefert, ob beide Patienten identisch sind. Basis hierfür sollen Name sowie Geburtsdatum sein. Nun können Sie im Krankenhaus eine `find()`-Methode schreiben, die einen Pointer auf einen Patienten bekommt und schaut, ob dieser im Krankenhaus derzeit behandelt wird. Der Rückgabetyt ist ein boolean. Um allerdings gleichzeitig auch Zugriff auf die Patientennummer des gefundenen Patienten zu erhalten, übergeben Sie der Methode als zweiten Parameter einen Pointer auf einen Integer. Wurde der Patient gefunden, soll hier zusätzlich die Patientennummer eingetragen werden. Wurde der Patient nicht gefunden, füllen Sie den Integer mit -1. Auf diese Weise lassen sich in C/C++ mehrere Rückgabewerte einer Methode realisieren.