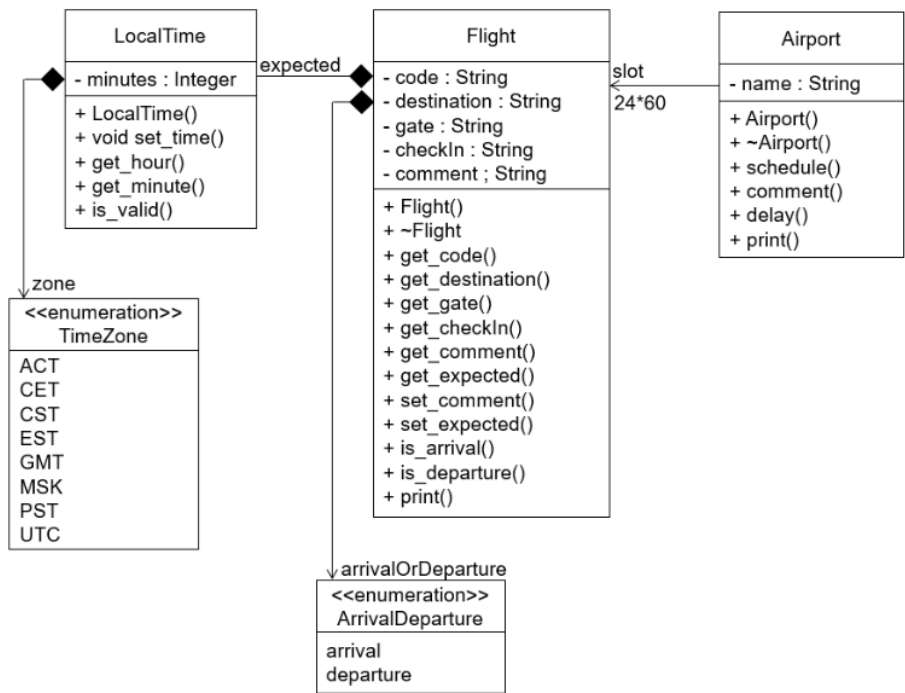


Flughafen Ankunft und Abflug/Airport Arrivals and Departures



Der Flughafen Düsseldorf plant eine Neuprogrammierung der Software für die Anzeige der An- und Abflugdaten im objektorientierten Programmierstil in C++. In dieser Aufgabe sollen Klassen und ein Beispiel für einen ersten Prototypen implementiert werden./

Dusseldorf airport plans to re-program the software showing arrivals and departures in an object-oriented programming style in C++. In this task classes and an example as a first prototype shall be implemented.

Teilaufgabe 1/Subtask 1

Zeitangaben auf Flughäfen werden immer einschliesslich Zeitzone definiert. Definieren Sie eine C++11 Aufzählung implementiert über vorzeichenlose Zahlen mit mindestens den folgenden acht Zeitzone: **ACT**, **CET**, **CST**, **EST**, **GMT**, **MSK**, **PST**, **UTC** (weitere siehe hier)./

Time points at airports are always defined including a time zone. Define a C++11 enumeration implemented by unsigned numbers with at least following eight time zones: **ACT**, **CET**, **CST**, **EST**, **GMT**, **MSK**, **PST**, **UTC** (more you can find here).

Teilaufgabe 2/Subtask 2

Definieren Sie eine C++11 Aufzählung mit den beiden Werten **arrival** (Ankunft) und **departure** (Abflug)./

Define a C++11 enumeration with two values **arrival** and **departure**.

Teilaufgabe 3/Subtask 3

Definieren Sie eine Klasse mit Namen **LocalTime**. Diese Klasse soll folgende Member haben:/

Define a class with name **LocalTime**. This class shall have following members:

- ein privates ganzzahliges Attribut **minutes**, das die Uhrzeit eines Tages in Minuten seit Mitternacht speichert (also speicherplatzsparend eine statt zwei Variablen für die Stunde und Minute)./
a private integer attribute **minutes** storing the minutes since midnight of a day (saving one variable instead of two variables for hour and minute).
- ein privates Attribut für die Zeitzone vom in Teilaufgabe 1 definierten Aufzählungstyp./
a private attribute for the time zone of enumeration data type defined in subtask 1.
- einen öffentlichen Standard-Konstruktor, der die Minuten mit **-1** (für Uhrzeit nicht gesetzt) und die Zeitzone mit **CET** initialisiert./
a public standard constructor initialising the minutes to **-1** (for clock time yet not set) and the time zone to **CET**.
- einen öffentlichen überladenen Konstruktor mit drei Parametern für eine Stunde und Minute und eine Zeitzone mit Defaultparameter **CET**, der die beiden privaten Attribute - geeignet umgerechnet - initialisiert (Hinweis: Rufen Sie einfacherweise im Rumpf die nachfolgend definierte setter-Funktion auf)./
a public overloaded constructor with three parameters for the hour, minute and a time zone with default parameter **CET** initialising the two attributes - minutes calculated respectively (hint: in the body simply call the following member function).
- eine öffentliche Methode mit Namen **set_time** mit einem Stunden- und einem Minutenwert als Parameter, die die Uhrzeit in Minuten ab Mitternacht umrechnet und an das entsprechende Attribut zuweist. Überprüfen Sie, dass die Werte für Stunde und Minute korrekt angegeben sind, ansonsten schreiben Sie eine Fehlermeldung und weisen **-1** als Wert zu./
a public method with name **set_time** with an hour and a minute value as parameters calculating the minutes since midnight and assigning the resulting minutes value to the respective attribute. Check that hour and minute are given correctly otherwise write an error message and assign value **-1**.
- zwei öffentliche Methoden mit Namen **get_hour** und **get_minute** ohne Parameter, die den Stunden- oder den Minutenwert berechnen und jeweils als ganze Zahl zurück geben (Hinweis: nutzen Sie die ganzzahlige Division durch 60 mit Rest)./
two public methods with names **get_hour** and **get_minute** without parameters calculating the hour or the minute and returning the result (hint: use integer division by 60 with rest).
- eine öffentliche Methode mit Namen **is_valid** ohne Parameter, die bei einem negativen Minutenwert **false** und ansonsten **true** zurück liefert./
a public method with name **is_valid** without parameters returning **false** for a negative value for the minutes value and **true** otherwise.

Teilaufgabe 4/Subtask 4

Weiterhin soll ausserhalb der Klasse ein überladener Ausgabeoperator << für ein Objekt der Klasse **LocalTime** definiert werden. Dieser soll die lokale Zeit immer in 5 Zeichen in der Form **hh:mm** auf den Ausgabestrom schreiben, also beispielsweise **00:00**, **09:10** oder **14:55**.

Furthermore outside of the class an overloaded output operator << for an object of class **LocalTime** shall be defined. The local time shall always be outputted with 5 characters in form **hh:mm**, for example **00:00**, **09:10** or **14:55**.

Teilaufgabe 5/Subtask 5

Definieren Sie eine Klasse mit Namen **Flight**, die folgende Member enthält:/

Define a class for flights with name **Flight** with following members:

- ein privates Attribut mit Namen **arrivalOrDeparture** vom in Teilaufgabe 2 definierten Aufzählungstyp./
a private attribute with name **arrivalOrDeparture** of enumeration data type defined in subtask 2.
- fünf private Attribute mit Namen **code** (Flugnummer), **destination** (von/nach), **gate** (Abflugschalter), **checkIn** (Schalter für das Gepäck) sowie **comment** (für Bemerkungen) alle vom Typ **string**./
five private attributes with names **code** (flight code), **destination** (from/to), **gate** (arrival or departure gate), **checkIn** (counters for dropping off baggage) as well as **comment** (for comments on this flight) all of type **string**.
- ein privates Attribut mit Name **expected** für die erwartete/verspätete An-/Abflugzeit vom obigen Typ **LocalTime**./
a private attribute with name **expected** for the expected/delayed arrival/departure time of above type **LocalTime**.
- einen öffentlichen Prototypen für einen Konstruktor mit fünf Parametern für Anflug oder Abflug, die Flugnummer, die Destination, das Gate und die Check-In-Schalter, letzterer mit dem Defaultparameter leere Zeichenkette ""./
a public prototype for a constructor with five parameters for arrival or departure, the code of the flight, the destination, the gate and the check-in counters, the last one with a default parameter empty string "" .
- einen öffentlichen Prototypen für einen Destruktor./
a public prototype for a destructor.
- sechs öffentliche Prototypen für getter-Funktionen mit den sprechenden Namen **get_code**, **get_destination**, **get_gate**, **get_checkIn**, **get_comment** und **get_expected** ohne Parameter und mit geeigneten Rückgabetypen./
six public prototypes for getter functions with speaking names **get_code**, **get_destination**, **get_gate**, **get_checkIn**, **get_comment** and **get_scheduled** without parameters and appropriate return types.
- zwei öffentliche Prototypen für setter-Funktionen mit ebenfalls sprechenden Namen **set_expected** und **set_comment** mit jeweils einem geeigneten Parameter./
two public prototypes for setter functions with also speaking names **set_expected** and **set_comment** each having one appropriate parameter.
- zwei öffentliche Prototypen für Funktionen mit ebenfalls sprechenden Namen **is_arrival** und **is_departure** ohne Parameter und mit Booleschem Rückgabewert./
two public prototypes for functions with also speaking names **is_arrival** and **is_departure** without parameter and a Boolean return value.

- ein öffentlicher Prototyp für eine Funktion mit Namen **print** mit einem Parameter vom obigen Typ **LocalTime** und ohne Rückgabewert./
a public prototype of a function with name **print** with a parameter of above type **LocalTime** and without return value.

Teilaufgabe 6/Subtask 6

Definieren Sie ausserhalb der Klasse mit Namen **Flight**/

Define outside of the class with name **Flight**

- den Konstruktor, der die als Parameter übergebenen Werte den entsprechenden Attributen zuweist und die Bemerkung/den Kommentar auf eine leere Zeichenkette initialisiert./
the constructor initialising the given parameter values to the regarding attributes and the comment to an empty string.
- den Destruktor mit einer Ausgabe der Flugnummer./
the destructor with an output of the flight code.
- die sechs getter- und zwei setter-Memberfunktionen./
the six getter and two setter member functions.
- die Funktion zur Ausgabe der Flugdaten formatiert in einer Zeile wie in den Beispielen unten, wobei die als Parameter übergebene Zeit in der Spalte **scheduled** (geplant) stehen soll./
the function printing flight data formatted in one line like in the examples below, whereas the time given as parameter shall be placed in column **scheduled**.

Teilaufgabe 7/Subtask 7

Definieren Sie eine Klasse mit Namen **Airport** (Flughafen), die folgende Member enthält:/

Define a class with name **Airport** with following members:

- ein privates Attribut mit Namen **name** für den Namen des Flughafens als C++ **string**./
a private attribute with the **name** of the airport as C++ **string**.
- ein privates Attribut mit Namen **slot** (Zeitscheibe) als Feld von Zeigern auf Flüge (Teilaufgabe 5) mit 24 Zeilen (für 24 Stunden eines Tages) und 60 Spalten (für 60 Minuten einer Stunde eines Tages), das mit Nullzeigern initialisiert werden soll (also maximal 24*60 Starts und Landungen an einem Tag; aufgrund von Luftverwirbelungen und aus Sicherheitsgründen muss ein Mindestabstand von einer Minute zwischen zwei Starts oder Landungen liegen). In diesem Feld sollen Zeiger auf geplante ankommende oder abgehende Flüge zu jeder Minute eines Tages gespeichert werden können./
a private attribute with name **slot** as array of pointers to flights (subtask 5) with 24 rows (for the hours of a day) and 60 columns (for the minutes of each hour of a day), getting initialised by null pointers (i.e. maximum 24*60 starts and landings each day; due to turbulences of the air and for security reasons at least one minute pause has to be inbetween two starts). In this array pointers to scheduled landings or starting flights in each minute of a day shall get stored.
- Definition eines öffentlichen Konstruktors mit einem Namen als Parameter zur Initialisierung des entsprechenden Attributs./
definition of a public constructor with a name as parameter to initialise the respective attribute.

- definition of a public constructor with a name as parameter to initialise the respective attribute.
definition of a public destructor deleting the whole pointer array with all flight objects on heap.
- Definition eines öffentlichen Destruktors, der das gesamte Zeigerfeld mit allen Flug-Objekten auf dem Heap löscht./
definition of a public destructor deleting the whole pointer array with all flight objects on heap.
- Definition einer öffentlichen Funktion mit Namen **schedule** und einer Uhrzeit **LocalTime** sowie einem Zeiger auf einen Flug als Parameter ohne Rückgabewert, in der dieser Flug im entsprechenden Zeit-**slot** gespeichert wird./
definition of a public function with name **schedule** having a **LocalTime** and a pointer to a flight as parameters without return value and storing the flight at the scheduled time **slot**.
- Definition einer öffentlichen Funktion mit Namen **comment** und einer Uhrzeit vom Typ **LocalTime** sowie einer Zeichenkette als Parameter ohne Rückgabewert, in der die Zeichenkette als Bemerkung/Kommentar zu dem um diese Uhrzeit geplanten Flug gespeichert wird./
definition of a public function with name **comment** with a **LocalTime** and a string as parameters having no return value and storing the string as comment of the flight at this scheduled time **slot**.
- Definition einer öffentlichen Funktion mit Namen **delay** und zwei Uhrzeiten vom Typ **LocalTime** als Parameter ohne Rückgabewert, in der die verspätete Uhrzeit im zweiten Parameter im Flug-Objekt zu der im ersten Parameter geplanten Uhrzeit gespeichert wird./
definition of a public function with name **delay** with two **LocalTime** parameters having no return value and storing the time in second parameter in the flight object at the scheduled time given in the first parameter.
- Definition einer öffentlichen Funktion mit Namen **print** mit einem Parameter vom obigen Aufzählungstyp aus Teilaufgabe 2 für An- oder Abflüge ohne Rückgabewert. Schreiben Sie im Rumpf dieser Funktion eine Tabelle aller im Feld gespeicherten ankommenden Flüge oder aller abfliegenden je nach Parameterwert (siehe Beispiele unten)./
definition of a public function with name **print** with one parameter of above enumeration type from subtask 2 for arrivals or departures without return value. In its body write a table of all stored arriving flights or starting flight given by the parameter value (see example below).

Teilaufgabe 8/Subtask 8

Schreiben Sie eine C++ **main**-Funktion mit folgenden Definitionen und Anweisungen:/

Write a C++ **main** function with following definitions and statements:

- Definieren Sie ein Objekt vom Typ **Airport** initialisiert mit Namen **DUSSELDORF AIRPORT**./
Define an object of type **Airport** initialised with name **DUSSELDORF AIRPORT**.
- Senden Sie diesem Objekt nacheinander geeignete Nachrichten mit neuen Objekten vom Typ **Flight** auf dem Heap zu den jeweils eingeplanten Uhrzeiten als Parameter mit allen unten im Beispiel gezeigten Flugdaten. Senden Sie ebenfalls geeignete Nachrichten zur Speicherung von Kommentaren/Bemerkungen und Verspätungen wie unten gezeigt. Es dürfen dabei nur die oben definierten Konstruktoren und Memberfunktionen aufgerufen und benutzt werden, also weder weitere Memberfunktionen oder Konstruktoren oder Parameter in diesen hinzu definiert noch welche weggelassen und auch keine Sichtbarkeiten von Mitgliedern der Klassen geändert werden./
Successively send to this object appropriate messages with new objects of type **Flight** on heap and with the scheduled time as parameters including all example flight data shown below. Also send appropriate messages to store comments and delays as shown below. Only above defined constructors and member functions are allowed to be used and called, no further or changed member functions, constructors, additional or deleted parameters or changed visibilities of class member data.

- Senden Sie dem Objekt für den Flughafen Düsseldorf eine Nachricht **print** zur Ausgabe der Tabelle aller gespeicherten Ankünfte sowie eine zweite Nachricht zur Ausgabe aller gespeicherten Abflüge./
Send to the object for Dusseldorf airport a message **print** to output the table of arrivals and a second message to output all departures.

Beispiel/Example

DUSSELDORF AIRPORT ARRIVALS

=====

Flight	From	Scheduled	Expected	Gate	Check-in	Comments
LH 2010	Munich	12:40	13:05	A04		
EW 9347	Manchester	14:50		B04		

DUSSELDORF AIRPORT DEPARTURES

=====

Flight	To	Scheduled	Expected	Gate	Check-in	Comments
AF 1307	Paris	09:10		B51	192-194	departed
SU 2537	Moscow	10:40		C31	252-255	boarding
EW 9466	London-Heathrow	11:15		B35	151-170	
LH 2011	Munich	13:25		A40	115-120	Code Sharing
XQ 959	Izmir	14:55	15:20	C45	240-242	

flight AF 1307 deleted

flight SU 2537 deleted

flight EW 9466 deleted

flight LH 2010 deleted

flight LH 2011 deleted

flight EW 9347 deleted

flight XQ 959 deleted