

Task 4

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.

Subtask 1

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).

Subtask 2

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

Subtask 3

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

- a private integer attribute minutes storing the minutes since midnight of a day (saving one variable instead of two variables for hour and minute).
- a private attribute for the time zone of enumeration data type defined in subtask 1
- a public standard constructor initializing the minutes to -1 (for clock time yet not set) and the time zone to CET.
- a public overloaded constructor with three parameters for the hour, minute and a time zone with default parameter CET initializing the two attributes - minutes calculated respectively (hint: in the body simply call the following member function).
- a public method with name sets 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.
- 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).
- a public method with name is_valid without parameters returning false for a negative value for the minutes value and true otherwise.

Subtask 4

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.

Task 4

Subtask 5

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

- a private attribute with name arrivalOrDeparture of enumeration data type defined in subtask 2.
- 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.
- a private attribute with name expected for the expected/delayed arrival/departure time of above type LocalTime.
- 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 "".
- a public prototype for a destructor.
- 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.
- two public prototypes for setter functions with also speaking names set_expected and set_comment each having one appropriate parameter.
- two public prototypes for functions with also speaking names is_arrival and is_departure without parameter and a Boolean return value.
- a public prototype of a function with name print with a parameter of above type LocalTime and without return value.

Subtask 6

Define outside of the class with name Flight

- the constructor initialising the given parameter values to the regarding attributes and the comment to an empty string.
- the destructor with an output of the flight code.
- the six getter and two setter member functions.
- 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.

Task 4

Subtask 7

Define a class with name Airport with following members:

- a private attribute with the name of the airport as C++ string.
- 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 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 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 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 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 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).

Subtask 8

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

- Define an object of type Airport initialised with name DUSSELDORF AIRPORT.
- 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.
- Send to the object for Dusseldorf airport a message print to output the table of arrivals and a second message to output all departures.

Task 4

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

Task 4

