



- Achten Sie darauf, nicht zu lange Zeilen, Funktionen und Dateien zu erstellen.
- Programmcode muss in englischer Sprache verfasst sein.
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich.
- Die Kommentare sollen einheitlich in englischer Sprache verfasst werden.
- Verwendbare Bibliotheken: C-Standard-Bibliothek (stdlib.h und stdio.h), Math (math.h), String (string.h). Das Nutzen anderer Bibliotheken führt zu Nichtbestehen der Abgabe.
- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Halten Sie die Regeln zu Variablen-, Funktionen- und Dateibenennungen ein und wählen Sie aussagekräftige Namen.
- Allgemeiner Hinweis: Bei Regelverletzung wird der Praktomat Ihre Abgabe zurückweisen.
- Ausgaben von Fließkommazahlen sollen auf 2 Stellen nach dem Komma begrenzt werden

Abgabemodalitäten

Die Praktomat-Abgabe wird voraussichtlich am 20. Januar 2022 freigeschaltet.

Achten Sie unbedingt darauf, diese Dateien im Praktomat bei der richtigen Aufgabe hochzuladen. Falsch hochgeladene Abgaben werden nicht bewertet. Bitte beachten Sie, dass das erfolgreiche Bestehen der öffentlichen Tests für eine erfolgreiche Abgabe nötig ist. Planen Sie für Ihren ersten Abgabeversuch entsprechend Zeit ein. Bitte beachten Sie: Sollten Sie Probleme bei der Abgabe bzw. beim Hochladen Ihrer Dateien haben, verwenden Sie einen anderen Browser. Der Internet Explorer kann Probleme verursachen. **In der Aufgabe 3 sind zu den jeweiligen C-Dateien auch die dazu gehörigen Header Dateien mit abzugeben.**

Aufgabe 3

Ziel ist die Programmierung eines vereinfachten Flight-Management-Systems, kurz FMS. Das FMS verwaltet Informationen des Flugzeugs von Pre-Engine Start bis Engine Shutdown, generiert die Darstellung und bietet eine Schnittstelle zur Programmierung von Flugrouten. In Fokus dieser Aufgabe steht das Programmieren und Darstellen von Flugrouten. Die Kommunikation mit dem Programm erfolgt über die Konsole.

Die Aufgabe gliedert sich in 4 Teile:

1. Erstellen der Datenstrukturen,
2. Erstellen der Funktion zur Ausgabe der Time Table,
3. Erstellen der Funktionen zur Ausgabe der Flugdauer und Strecke und
4. Erstellen der Testdaten sowie der main Funktion

Alle Aufgaben sind Teil eines einzigen Projektes und werden gemeinsam abgegeben.

Checkliste zur Abgabe:

In Datei „fms.h“:

- Deklaration der Datentypen
(FMS, Airline, Airplane, FlightPath, Airport, Coordinate, Time)
- Deklaration der Funktionen:
 - Generate_Time_Table
 - Generate_Distance_Table
 - Generate_Duration_Table
 - Calculate_Degree
 - Sin_Deg
 - Cos_Deg

In Datei „fms.c“:

- Definition der Funktionen:
 - Generate_Time_Table
 - Generate_Distance_Table
 - Generate_Duration_Table
 - Calculate_Degree
 - Sin_Deg
 - Cos_Deg

In Datei „fms_data.c“:

- Definition und Füllen der Arrays mit Testdaten

In Datei „main.c“:

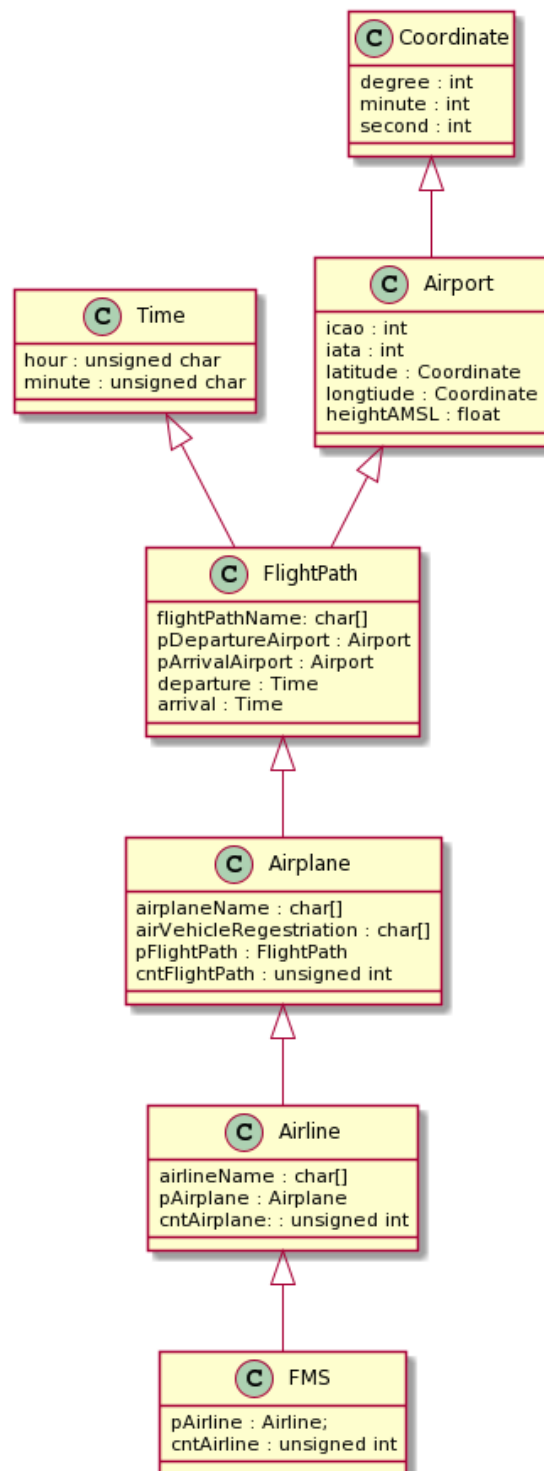
- Deklaration und Definition der main-Funktion

Abzugebene Dateien:

- fms.h, fms.c, main.c, fms_data.c

Erstellen der Datenstrukturen

Das FMS soll folgende Datenstrukturen nutzen:



Erstellen der Time Table Ausgabe

In diesem Aufgabenteil soll nun die Ausgabe der Time Table implementiert werden.

Die Funktion soll mit folgendem Funktionskopf definiert werden:

```
void Generate_Time_Table(FMS * pFMS);
```

Die Funktion erstellt eine Übersicht über alle eingetragenen Daten der Airlines.

- Die Zellenbreiten der Tabellen sind vollgendermaßen definiert, wobei die Zahlen die Zeichen pro Zelle angibt:

Generate_Time_Table: 10|10|4|4|9|8|8

Die Überschriften sollen nach dem Beispiel ausgeführt werden. Es entsteht folgende Tabelle (vollständig):

Die Time Tabelle für alle Airlines aus den Testdaten

Airline	Flight No.	from	to	Departure	Arrival	Airplane
Lufthansa	LH 2147	MUC	HAM	11:30	14:30	D-AIMA
Lufthansa	LH 2147	HAM	FRA	15:30	16:30	D-AIMA
Lufthansa	LH 2147	FRA	MUC	18:30	20:30	D-AIMA
Lufthansa	LH 2147	MUC	HAM	20:45	22:00	D-AIMA
Lufthansa	LH 2147	HAM	MUC	22:30	23:30	D-AIMA
EuroWings	EW 5953	MUC	DUS	03:00	04:15	D-AIAB
EuroWings	EW 5953	DUS	BER	08:00	10:00	D-AIAB
EuroWings	EW 5953	BER	CGN	10:12	13:15	D-AIAB
EuroWings	EW 5953	CGN	MUC	18:00	20:30	D-AIAB
EuroWings	EW 5953	MUC	DUS	21:45	23:30	D-AIAB
TUIFly	X3 2174	HAM	STR	08:00	10:00	D-ABCD
TUIFly	X3 2174	STR	BER	10:15	13:00	D-ABCD
TUIFly	X3 2174	BER	FRA	13:10	15:00	D-ABCD
TUIFly	X3 2174	FRA	HAM	15:12	17:00	D-ABCD
TUIFly	X3 2174	HAM	FRA	22:00	24:00	D-ABCD

Erstellen der Funktionen zur Berechnung von Flugdauer und Strecke

Die Berechnung der Flugdauer basiert auf den Einträgen der Airline zur jeweiligen Flugroute und zum jeweiligen Flugzeug. Sie soll einen Überblick über die Flugdauer zwischen zwei Flughäfen und die Gesamtflugdauer eines Flugzeuges geben.

Analog dazu soll die Strecke berechnet und dargestellt werden. Hier soll aber die Erdkrümmung berücksichtigt werden. Die Formel, sowie zwei Hilfsfunktionen stehen unten.

Als Strecke gilt die direkte Luftlinie zwischen beiden Flughäfen, Änderungen durch den An- und Abflugvektor, Warteschleifen, usw. sollen der Einfachheit halber vernachlässigt werden.

Die Funktionen sollen mit folgendem Funktionskopf definiert werden:

```
void Generate_Distance_Table (FMS * pFMS);
```

```
void Generate_Duration_Table (FMS * pFMS);
```

Die Zellenbreiten der Tabellen sind vollgendermaßen definiert, wobei die Zahlen die Zeichen pro Zelle angibt:

- Generate_Distance_Table: 10|10|4|4|8
- Generate_Duration_Table: 10|10|4|4|8

Die Überschriften sollen nach dem Beispiel unten ausgeführt werden. Es entstehen folgende Tabellen (nicht vollständig):

Tabelle für die Flugdauer Berechnung aus den Testdaten

Airline	Flight No.	from	to	Duration
Lufthansa	LH 2147	MUC	HAM	03:00
Lufthansa	LH 2147	HAM	FRA	01:00
Lufthansa	LH 2147	FRA	MUC	02:00
Lufthansa	LH 2147	MUC	HAM	01:15
Lufthansa	LH 2147	HAM	MUC	01:00

Total flight Time for Lufthansa A380 is 08:15

Tabelle für die Strecken Berechnung aus den Testdaten

Airline	Flight No.	from	to	Distance
Lufthansa	LH 2147	MUC	HAM	600.69
Lufthansa	LH 2147	HAM	FRA	412.14
Lufthansa	LH 2147	FRA	MUC	299.40
Lufthansa	LH 2147	MUC	HAM	600.69
Lufthansa	LH 2147	HAM	MUC	600.69

Total flight distance for Lufthansa A380 is 2513.62

Formel Erdkrümmung:

Bei der Berechnung der Strecke muss die Erdkrümmung mitberücksichtigt werden. Hierzu die folgende Formel:

$$\text{Strecke} = 6378.388 * \text{acos}(\text{sindeg}(\text{lat1}) * \text{sindeg}(\text{lat2}) + \text{cosdeg}(\text{lat1}) * \text{cosdeg}(\text{lat2}) * \text{cosdeg}(\text{lon2} - \text{lon1}))$$

Wobei lat1 und lon1 vom Airport Departure und lat2 und lon2 vom Airport Arrival sind.

Hilfsfunktionen:

```
double Sin_Deg(double x)
{
    return sin((x)*M_PI/180.0);
}
double Cos_Deg(double x)
{
    return cos((x)*M_PI/180.0);
}
```

Die Konstante M_PI wird aus der math.h entnommen und stellt den Wert PI dar.

Erstellen der Testdaten sowie der main Funktion

Erstellen der Testdaten (fms_data.c)

In der Datei fms_data.c sollen nun die Arrays mit den Testdaten gefüllt werden. Die Testdaten sind im Folgenden definiert.

Testdaten Flughafen

Jeder Flughafen soll die folgenden Kenndaten haben: ICAO, IATA, Breitengrad, Längengrad, Höhe über MSL

Die Daten können aus Wikipedia entnommen werden.

Flughafen Berlin-Tegel „Otto Lilienthal“	
TXL	FLUGHAFEN BERLIN BRANDENBURG
	
Kenndaten	
ICAO-Code	EDDT (bis 1995: EDBT)
IATA-Code	TXL
Koordinaten	52° 33′ 35″ N, 13° 17′ 16″ O 37 m (121 ft.) über MSL

https://de.wikipedia.org/wiki/Liste_der_Verkehrsflugh%C3%A4fen_in_Deutschland

Folgenden Flughäfen sollen in ihrem Programm zum Testen verwendet werden:

- Hamburg
- Berlin-Brandenburg
- Düsseldorf
- Köln/Bonn
- Frankfurt am Main
- Stuttgart
- München

Testdaten Fluggesellschaften

Jede Fluggesellschaft hat einen Namen, genau ein Flugzeug (Name und Kennung) und fünf Flugrouten für das Flugzeug.

Die folgenden Fluggesellschaften sollen im Programm angelegt werden:

- Lufthansa
- EuroWings
- TUIfly

Folgende Flugzeuge mit Kennung und Zugehörigkeit zu den Fluggesellschaften existieren:

- A380 (D-AIMA, Lufthansa)

- A320 (D-AIAB, EuroWings))
- B747 (D-ABCD, TUIfly)

Testdaten Flugverbindungen

Folgende Flugverbindungen existieren:

Airline	Flight No.	from	to	Departure	Arrival	Airplane
Lufthansa	LH 2147	MUC	HAM	11:30	14:30	D-AIMA
Lufthansa	LH 2147	HAM	FRA	15:30	16:30	D-AIMA
Lufthansa	LH 2147	FRA	MUC	18:30	20:30	D-AIMA
Lufthansa	LH 2147	MUC	HAM	20:45	22:00	D-AIMA
Lufthansa	LH 2147	HAM	MUC	22:30	23:30	D-AIMA
EuroWings	EW 5953	MUC	DUS	3:00	4:15	D-AIAB
EuroWings	EW 5953	DUS	BER	8:00	10:00	D-AIAB
EuroWings	EW 5953	BER	CGN	10:12	13:15	D-AIAB
EuroWings	EW 5953	CGN	MUC	18:00	20:30	D-AIAB
EuroWings	EW 5953	MUC	DUS	21:45	23:30	D-AIAB
TUIfly	X3 2174	HAM	STR	8:00	10:00	D-ABCD
TUIfly	X3 2174	STR	BER	10:15	13:00	D-ABCD
TUIfly	X3 2174	BER	FRA	13:10	15:00	D-ABCD
TUIfly	X3 2174	FRA	HAM	15:12	17:00	D-ABCD
TUIfly	X3 2174	HAM	FRA	22:00	24:00	D-ABCD

Erstellen der main Funktion (main.c)

Als letztes soll nun die main Funktion erstellt werden. Diese ist einfach und ruft der Reihe nach die Funktionen Generate_Time_Table, Generate_Duration_Table und Generate_Distance_Table.

Hinweis zu Arrays

Hinweis zur Übergabe von Arrays an Funktionen:

Listen sind in unserer Aufgabe einfache Array's.

In C ist möglich ein Array als Referenz an eine Funktion zu übergeben. Um dann innerhalb der Funktion auf ein Array Element zuzugreifen kann wie gewohnt der Index verwendet werden.

Beispiel:

```
void function(char * ar)
{
    int i;
    for (i=0; i< 2; i++) ar[i]=i;
}

void main(void)
{
    int arr[30];
    function(arr);
}
```

Eigenschaften:

- Arrays sind statisch. Das bedeutet, ihre Größe wird zur Compilezeit festgelegt und kann während der Laufzeit des Programms nicht mehr verändert werden.
- Arrays enthalten nur Werte von einem Datentyp (z. B. int, float oder komplexen Typen wie structs).
- Die Werte eines Arrays liegen im Speicher direkt hintereinander.
- Der Name eines Arrays verhält sich wie ein Zeiger auf das erste Element. Wenn zum Beispiel *arr* ein Array mit 30 Integern repräsentiert (`int arr[30];`), so gilt: `&arr[0] == arr` (Die Adresse des ersten Elementes ist gleich dem Namen).

Länge von Arrays

Die Länge (oder Anzahl der Elemente) eines Arrays lässt sich über die Funktion „`sizeof`“ bestimmen:

```
length = sizeof(arr)/sizeof(arr[0]) /*ergibt dann 30 */
```