

Programmieren, Algorithmen, Datenstrukturen 1

Projektanforderungen/Aufgabenstellung/Themen

BMR-VZ-2 PAD2 SS2020



Aufgabenstellungen

Projektmanagement

- Projektstart: KW19 - 04.05.2020
- Formieren Sie Teams von 3-4 Studenten und definieren Sie eine klare Rollenverteilung
- Erstellen Sie im Team gemeinsam einen Projektantrag/Grobspezifikation (mögliche Themenstellungen siehe Präsentation) Abgabe: 14.05.2020 24h00
- Feedback zum Projektantrag: 18.05.2020
- Spezifizieren und dokumentieren Sie den geplanten Lösungsweg (Softwarekomponenten/Libraries)
- Entwicklung/Dokumentation – Möglichkeit für Zwischenpräsentationen, Diskussionen in den parallelen Vorlesungen
- Projektende/Abgabe: 08.06.2020 24h00
- Abschlusspräsentationen der Projektarbeiten (Testprogramm mit Libraries und Dokumentation) 09.06.2020-19.06.2020

Vorgehensweise:

- Grundsätzlich müssen die im Team erarbeiteten Lösungen als Funktionsbibliothek realisiert werden. Dazu sind für unterschiedlichen Themenkreise im Projekt spezifische Libraries zu implementieren.
- Für die Überprüfung der Funktionen muss eine geeignete Testapplikation mit einer detaillierten Beschreibung der angewendeten Testcases zur Verfügung gestellt werden
- Die Ein- und Ausgabeformate von Daten sowie eine allfällige Benutzeroberfläche für die Testapplikation muss detailliert dokumentiert werden

Deliverables:

- Projektdokumentation
 - Projektstruktur
 - Spezifikationen
 - Rollenbeschreibungen
 - Zeitaufwand pro Teammitglied und Zuordnung zu Teilprojekten (Funktionen,...)
- Bereitstellung und Dokumentation der entwickelten Funktionen als Libraries für die Nutzung in anderen Applikationen
- Präsentation der Lösung(en)

Bewertungsschlüssel (30% der Gesamtnote)

Bewertungskriterien (100 Punkte gesamt)

- Spezifikation 20 Punkte
- Implementierung 40 Punkte
 - Libraries 30 Punkte
 - Testapplikation 10 Punkte
- Dokumentation 20 Punkte
- Präsentation 20 Punkte

Zusatzbewertung:

- Abgabetermine nicht eingehalten -5 Punkte
- Libraries nicht portierbar -10 Punkte

Aufgabenstellungen

Themenkreis 1 – Auswertung von Sensor-Rohdaten

Es stehen von allen Sensoren des RaspberryPi SenseHAT Datensätze von ausgelesenen Werten als CSV-Dateien zur Verfügung. Die einzelnen Datensätze enthalten einen Zeitstempel sowie die Rückgabewerte der Sensoren im Raw – Format. Die für die Auswertung der Daten erforderlichen Informationen sind den entsprechenden Datenblättern der Sensoren zu entnehmen.

Sensoren:

- HTS221 - Capacitive digital sensor for relative humidity and temperature
- LPS25HB - MEMS pressure sensor: 260-1260 hPa absolute digital output barometer
- LSM9DS1 - iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer

Aufgabenstellung/Vorgehensweise: Entwicklung von Funktionen für

- Plausibilitätscheck (Outlier,...)
- Data-Cleaning – zB.: Interpolation fehlender oder ausserhalb der Spezifikation liegender Werte
- Einfache und kombinierte Auswertung der Sensordaten
- Berechnung von Bewegungsprofilen (Ausgabe als 3D Koordinaten)

Aufgabenstellungen

Themenkreis 1 – Auswertung von Sensor-Rohdaten

Deliverables:

- Bereitstellung und Dokumentation der entwickelten Funktionen als Libraries für die Nutzung in anderen Applikationen
- Testapplikation für alle entwickelten Libraries
- Ausgabedateien im lesbaren Format (z.B. Bewegungsprofil x,y,z Koordinaten)
- Präsentation des Projektes

Aufgabenstellungen

Themenkreis 1 – Auswertung von Sensor-Rohdaten

Die Datensätze stehen in folgendem Format zur Verfügung:

LSM9DS1 IMU:

```
000000005;000000000;fffffffff4;00000569;fffffffea;ffffffd9;ffffff5;0000019c;0000004b;fffffb8c
(1)           (2)           (3)           (4)           (5)           (6)           (7)           (8)           (9)           (10)
```

- (1) Zeitstempel in Millisekunden (Start bei 0)
- (2-4) Accelerometer (FS = +-16g), mg/LSB x,y,z
- (5-7) Gyroskop (FS = +- 2000 dps), mdps/LSB x,y,z
- (8-10) Magnetometer (FS = +- 12 gauss (mgauss/LSB) x,y,z

Verwendete Konfiguration der Sensoren des LSM9DS1 für die Testdaten

```
// Enable the gyroscope
CTRL_REG4,0b00111000; // z, y, x axis enabled for gyro
CTRL_REG1_G,0b10111000; // Gyro ODR = 476Hz, 2000 dps
ORIENT_CFG_G,0b10111000; // Swap orientation

// Enable the accelerometer
CTRL_REG5_XL,0b00111000; // z, y, x axis enabled for accelerometer
CTRL_REG6_XL,0b00101000; // +/- 16g

//Enable the magnetometer
CTRL_REG1_M, 0b10011100; // Temp compensation enabled, Low power mode,80Hz ODR
CTRL_REG2_M, 0b01000000; // +/-12gauss
CTRL_REG3_M, 0b00000000; // continuos update
CTRL_REG4_M, 0b00000000; // lower power mode for Z axis
```

Aufgabenstellungen

Themenkreis 1 – Auswertung von Sensor-Rohdaten

LPS25HB – Temperatur – Druck

```
1587731401.167600;01;f2;c9;fd;3d
      (1)          (2) (3) (4) (5) (6)
      (24.04.2020      (35,04° C) (992hPa)
      14:30:01.1676)
```

- (1) Zeitstempel in Sekunden (Linux Format)
- (2) TEMP_OUT_L (Temperatur LSB) (° F)
- (3) TEMP_OUT_H (Temperatur MSB)
- (4) PRESS_OUT_XL (Druck LSB)
- (5) PRESS_OUT_L (Druck Mid Part)
- (6) PRESS_OUT_H (Druck MSB)

HTS221 – Temperatur - rel. Luftfeuchtigkeit

```
1587727640.143500;dd;02;96;06
      (1)          (2) (3) (4) (5)
      (24.04.2020      (33,27° C) (28,35%)
      14:30:01.1435)
```

- 1) Zeitstempel in Sekunden (Linux Format)
- (2) TEMP_OUT_L (Temperatur LSB) (0x)
- (3) TEMP_OUT_H (Temperatur MSB) (0x)
- (4) HUMIDITY_OUT_L (Druck LSB) (0x)
- (5) HUMIDITY_OUT_H (Druck MSB) (0x)

HTS221 – Parameter für Umrechnung (0x)

t0_out_l	=	fb	h0_out_l	=	05
t0_out_h	=	ff	h0_out_h	=	00
t1_out_l	=	d5	h1_out_l	=	aa
t1_out_h	=	02	h1_out_h	=	d9
t0_degC_x8	=	9e	h0_rh_x2	=	44
t1_degC_x8	=	09	h1_rh_x2	=	86
t1_t0_msb	=	c4			

```
t_gradient_m = 0,02  t_intercept_c = 19,84
Temp = 33,53  Hum = 28,27  t_gradient_m = 0,02  t_intercept_c = 19,84
```

```
1587728067.118800;00fb;00ff;00d5;0002;009e;0009;00c4;00eb;0002;0005;0000;00aa;00d9;0044;0086;00ae;0006
```


Aufgabenstellungen

Themenkreis 2 – Sortieralgorithmen

Ihr Team wird von einer Speditionsfirma beauftragt, die Routen Ihrer LKW Flotte zu organisieren. Ihrem Team steht eine CSV Datei zur Verfügung, welche folgende Städteinformationen enthält:

- Name der Stadt
- Einwohnerzahl
- Standort (Längen- & Breitengrad)
- Ländername & -kürzel
- etc.



Aufgabenstellungen

Themenkreis 2 – Sortieralgorithmen

Der Kunde wünscht sich folgende Punkte von Ihrer Software:

- Die Suche nach Städtenamen soll schnell funktionieren
- Die Städte sollen nach Namen und nach Einwohnerzahl sortiert ausgegeben werden können
- Wenn der Benutzer **N** Städtenamen eingibt, soll die kürzeste Route ermittelt werden (*Traveling Salesman Problem*)

- **Zusatzaufgabe:**

Aus logistischen Gründen sollen die Waren erst in die Hauptstädte und von dort in extra Fahrten zu den einzelnen Zielorten gebracht werden.

Beispiel (Graz, Dresden, Frankfurt, Linz):

Route 1: Zentrale->Berlin->Wien->Zentrale

Route 2: Wien->Graz->Innsbruck->Linz->Wien

Route 3: Berlin->Dresden->Frankfurt->Berlin

Zusätzlich zu den Routen können Sie die Anzahl benötigter Transporter und die gefahrenen Kilometer ausgeben. Auch die Kilometer pro Route wären interessant. Als Zentrale können Sie sich eine Stadt aussuchen und bitte achten Sie darauf, dass Städtenamen auch doppelt vorkommen können.



Themenkreis 2 – Sortialgorithmen

- Datenstrukturen als auch Algorithmen können von Ihrem Team gewählt werden
- Implementieren Sie in Ihrer Bibliothek zu jedem Anwendungsfall (Suche nach Namen, Sortieren nach Namen & Einwohnerzahl und Berechnung der kürzesten Route) mindestens zwei Algorithmen und vergleichen Sie diese miteinander
 - Best-Case, Average-Case, Worst-Case
 - Big O Notation bzw Landau-Symbole
 - Dokumentieren Sie Ihre Erkenntnisse
 - Erstellen Sie Testprogramme, welche die Funktionen Ihrer Library testen
 - Argumentieren Sie, warum Sie sich für einen bestimmten Algorithmus entschieden haben
 - Zeitmessungen in Ihrer Testapplikation sollten Sinnvoll durchgeführt werden, um die Algorithmen vergleichen zu können
 - Es sollte auch möglich sein neue Städte mittels Konsole einzulesen. Diese Städte sollen auch bei einem neuen Programmstart zur Verfügung stehen.
- Die Daten für die kürzeste Route können auf der Konsole ausgegeben werden und müssen zusätzlich in eine CSV Datei gespeichert werden
 - **Zusatzaufgabe:** Eine grafische Ausgabe mittels externer Bibliothek (zB Graphviz)

Alle allgemeinen Kriterien zur Bewertung gelten auch für diese Aufgabe.

Aufgabenstellungen

Themenkreis 3 – Zeitseriendatenbank

Entwickeln Sie eine Datenbank samt zugehöriger Library zum effizienten Speichern und Auswerten von zeitbasierten Daten.

Die Datenbank soll zweierlei Datenarten unterstützen: absolute Werte und Zähler. Dazu soll es möglich sein, beim Erstellen einer neuen Datenbank, anzugeben, wieviele Datenwerte in welcher Auflösung gespeichert werden sollen. So soll beispielsweise eine Datenbank angelegt werden können, die einen Temperaturwert alle 5 Minuten über eine Woche speichert oder ein Fahrzeugzähler einer Autobahn alle Fahrzeuge der letzten 30 Minuten für 5 Monate speichert.

Die Libraryfunktionen sollen nachträgliche Änderungen in der Datenbank unterbinden und sich jeweils merken, für welches Zeitintervall bereits eingetragen wurde. Überlegen Sie Strategien zum Umgang mit fehlenden Werten in vorangegangenen Intervallen. Bedenken Sie dabei nicht nur die Speicherung der Daten sondern auch etwaige Auswertungsfunktionen.

Zum Auswerten der Daten sollen sinnvolle Auswertungsfunktionen wie Minimum, Maximum, Durchschnitt usw. für definierte Intervalle zur Verfügung stehen. Überlegen Sie dabei auch, welche Funktionen für absolute Werte relevant sind und welche für Zähler.

Aufgabenstellungen

Themenkreis 3 – Zeitseriendatenbank

Folgendes ist zur Lösung der Aufgabe erforderlich:

- Definieren Sie das Dateiformat für Ihre Datenbank; Ihre Library muss diese Dateien lesen und schreiben können
- Spezifizieren Sie die Programmierschnittstelle (API) Ihrer Library – also die Funktionen die von anderen Programmierern verwendet werden können
- Erstellen Sie Testprogramme, die den Funktionsumfang Ihrer Library demonstrieren

Alle allgemeinen Kriterien zur Bewertung gelten natürlich für diese Aufgabe ebenso.

Zusatzaufgabe: implementieren Sie ein Langzeitarchiv-Funktion in dem Daten, bevor Sie aus der Datenbank gelöscht werden, zusammengefasst, ausgedünnt und gespeichert werden.

Beispielsweise sollen im Langzeitarchiv die Zahl der Fahrzeuge die an einem ganzen Tag vorbeigekommen sind für 3 Jahre gespeichert werden. Das Archiv soll ebenfalls vom Benutzer konfiguriert werden (Speicherdauer, Intervall) – im Prinzip also die gleiche Art von Datenbank.

Aufgabenstellungen

Themenkreis 4 – Auswertung und Zeichnen von Personendaten

Schreiben Sie ein Programm um Personenkontakte auswerten zu können.

Die damit in Verbindung stehenden Algorithmen zur Auswertung finden sie in der Literatur unter Graphensuche.

Die Aufgabe teilt sich dabei in folgende Teile

- Testdatenerstellung
- Auswertung
- Visualisierung

Aufgabenstellungen

Themenkreis 4 – Auswertung und Zeichnen von Personendaten

Testdaten sollen selber automatisch generiert werden mithilfe einer Namensdatenbank und einfachen Zufallswerten. Versuchen Sie auch Familienstrukturen zu generieren. Ein Eintrag sollte in Verbindung mit **N** anderen Einträgen stehen können und auch die Kontaktzeit beinhalten bzw Start und Endzeit (time.h). Auch soll es möglich sein, dass eine Person öfters zu einer anderen Kontakt hat.

Eine VornamenDB → <https://www.heise.de/ct/ftp/07/17/182/>

Aufgabenstellungen

Themenkreis 4 – Auswertung und Zeichnen von Personendaten

Auswertung überlegen sie sich sinnvolle Auswertungen. Abfrage nach Person. Durchschnittskonakzeit. Backtracing. Graphensuche Dijkstra, A-Star

<https://de.wikipedia.org/wiki/Dijkstra-Algorithmus>

→ <https://www.youtube.com/watch?v=2poq1Pt32oE>

https://de.wikipedia.org/wiki/A*-Algorithmus

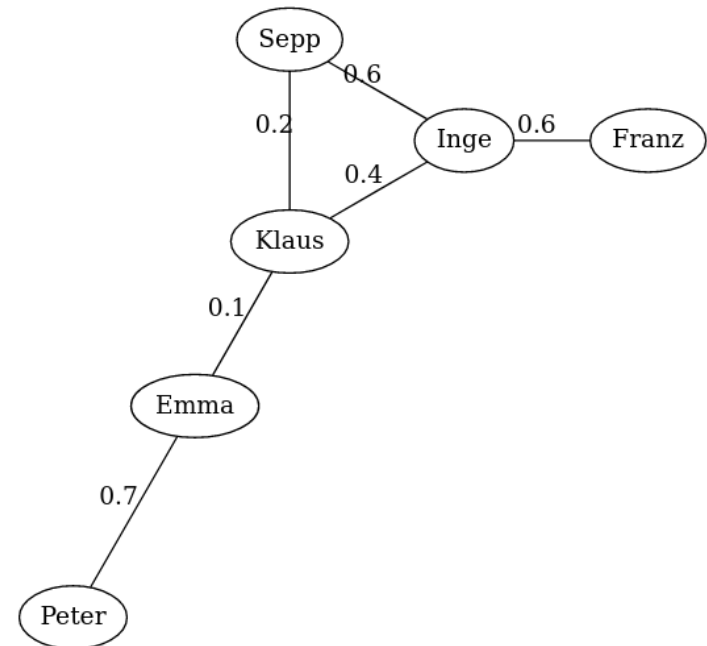
Aufgabenstellungen

Themenkreis 4 – Auswertung und Zeichnen von Personendaten

Ihr Projekt sollte eine **Visualisierung** der Daten ermöglichen. Dazu muss allerdings keine Visualisierung implementiert werden. Sie können das Programm Graphviz verwenden um Ihre Daten zu visualisieren. Dazu müssen Sie allerdings eine Datei schreiben die Graphviz lesen kann.

<https://dreampuf.github.io/GraphvizOnline>

```
graph {  
    Klaus -- Sepp[label="0.2",weight="0.2"];  
    Klaus -- Inge[label="0.4",weight="0.4"];  
    Inge -- Sepp[label="0.6",weight="0.6"];  
    Inge -- Franz[label="0.6",weight="0.6"];  
    Peter -- Emma[label="0.7",weight="0.7"];  
    Emma -- Klaus[label="0.1",weight="0.7"];  
}
```



Ausarbeitung der Projekte

Für die Ausarbeitung und Zusammenarbeit im Team werden folgende Software-Tools empfohlen:

- cmake
- git
- doxygen

Herzlichen Dank für Ihre Aufmerksamkeit und Mitarbeit!

Lektor/en

<mail@technikum-wien.at>

Department Computer Science

