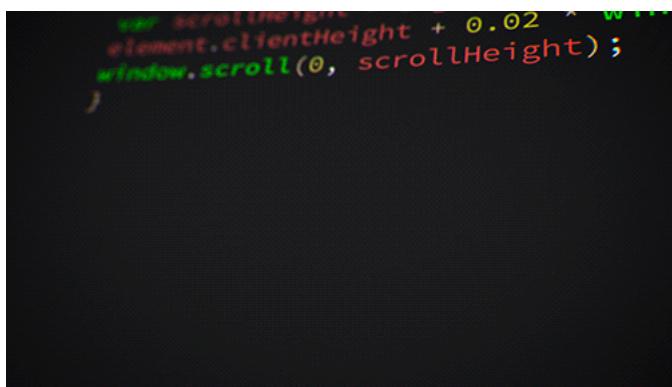


Finale Anwendung und Ausblick

Parameter	Kursinformationen
Veranstaltung:	Prozedurale Programmierung / Einführung in die Informatik
Semester	Wintersemester 2022/23
Hochschule:	Technische Universität Freiberg
Inhalte:	Anwendungen der Datenerhebung und Analyse
Link auf Repository:	https://github.com/TUBAF-IfI-LiaScript/VL_ProzeduraleProgrammierung/blob/master/12_Anwendungen.md
Autoren	Sebastian Zug & André Dietrich & Galina Rudolf



Organisatorisches

Konsultationsangebote der Tutoren und Mitarbeiter ()

- Mittwoch, 8.02.2023, 17:45 (Rammel Bau)
- weitere Termine nach Abstimmung in den Übungsgruppen sowie im Übungsordner im Git

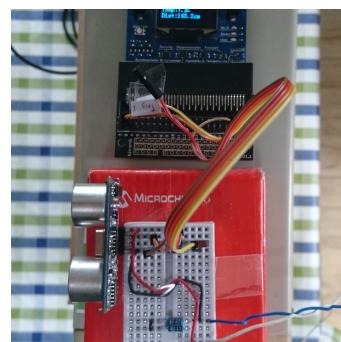
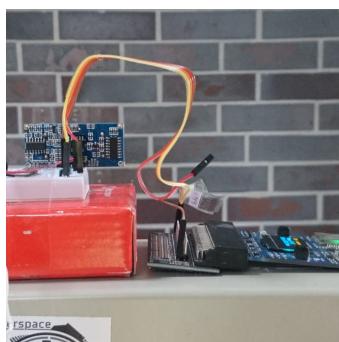
Fragen an die heutige Veranstaltung ...

- In welche strukturellen Einheiten lässt sich die Verarbeitungskette ?

Anwendung

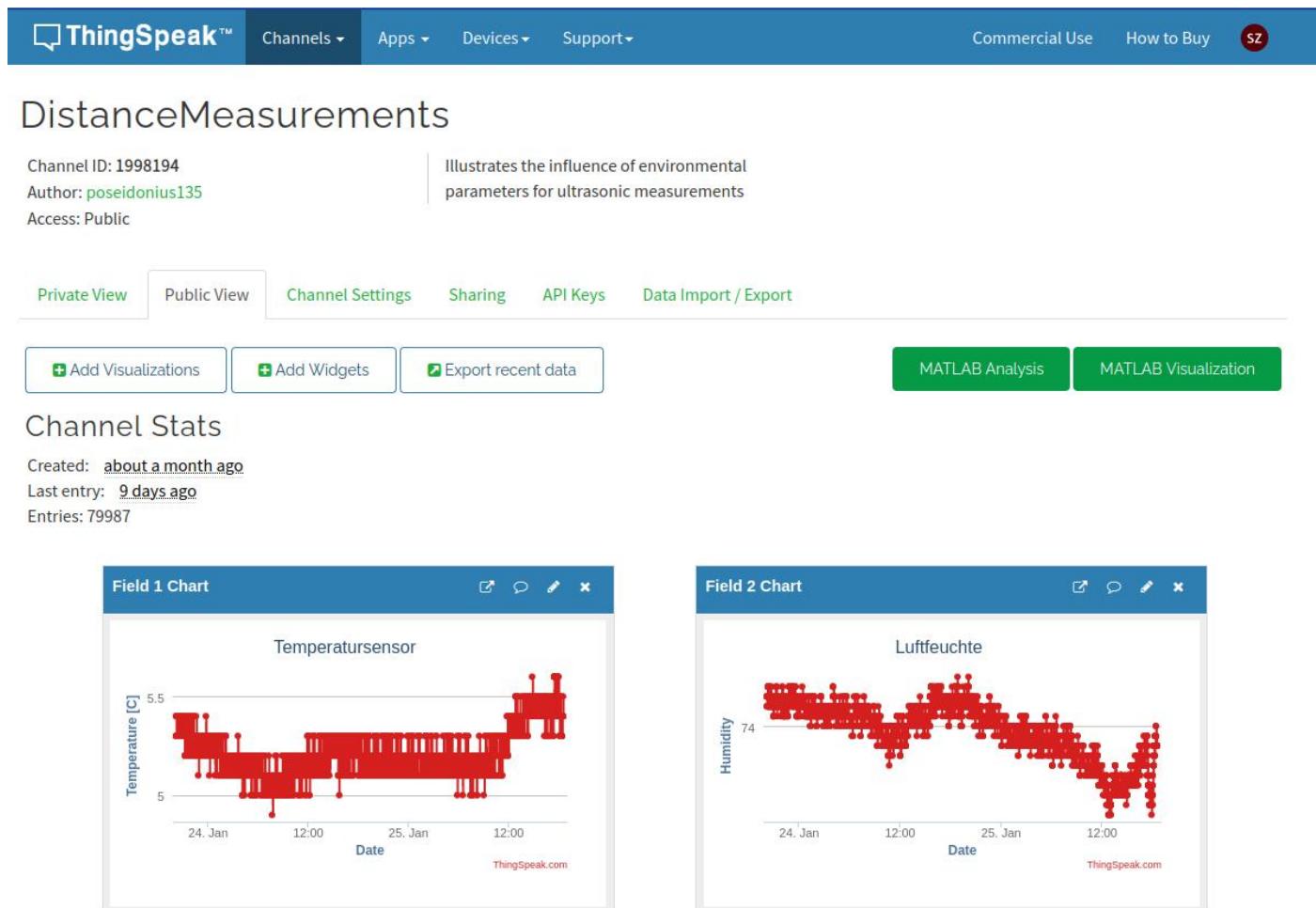
Mit Blick auf die Sensoren haben wir darüber gesprochen, dass die Distanzmessung von Ultraschallsensoren von der Temperatur abhängt.

Wir wollen dies mit einer sehr einfachen Messkampagne bestätigen. Dazu wurde ein [HC-SR04](#) Ultraschallsensor in einem ungeheizten Gebäude auf eine Wand gerichtet. Die Distanz zur Wand betrug 1.45m



Datenerhebung

Der Messaufbau wurde über 3 Wochen im Januar 2023 betrieben. Die gesammelten Daten wurden über ein WLAN-Kommunikation mit der Webseite <https://thingspeak.com/> abgelegt. Der zugehörige Code findet sich im [Projektrepository](#).



<https://thingspeak.com/channels/1998194>

Datenfilterung

Über den Export von Thinkpeak wurde das gesamten Datenset als CSV-Datei exportiert.

```
created_at,entry_id,field1,field2,field3,latitude,longitude,elevation,status
2023-01-06T16:44:16+00:00,1,12.600000,69.699997,8512.000000,,,
2023-01-06T16:44:36+00:00,2,12.600000,69.800003,8610.000000,,,
2023-01-06T16:44:57+00:00,3,12.400000,69.800003,8365.000000,,,
2023-01-06T16:45:17+00:00,4,12.400000,70.000000,8536.000000,,,
```

Folgende Adaptionen waren notwendig:

- Wie im [Programmcode](#) sichtbar, wurde alle 20s eine Messung erhoben. Diese zeitliche Auflösung ist für die weitere Verwendung (die Daten werden bei jedem Start der Analysen von Github geladen) zu hoch.
- Die Spaltenköpfe treffen keine Aussage zu zur Bedeutung der Messwerte - hier sollten aussagekräftige Spaltenbezeichner verwendet werden.

Für die Realisierung wurde in Python-Skript verwendet

```
import pandas as pd

df=pd.read_csv("distanceMeasurements.csv", sep=',', header = 0)
# Löschen der irrelevanten Spalten aus dem Datensatz
df.drop(['entry_id', 'latitude', 'longitude', 'elevation', 'status'], axis=1,
    inplace=True)
# Entfernen aller Messungen mit geradem Index
df_filtered = df[df.reset_index().index % 2 != 0] .copy()
df_filtered.head(5)

df_filtered.reset_index(drop=True, inplace = True)
# Umhennen der Spaltennamen
```

Datenanalyse

Fragekomplex 1: Welche Temperaturrentwicklung konnte beobachtet werden?

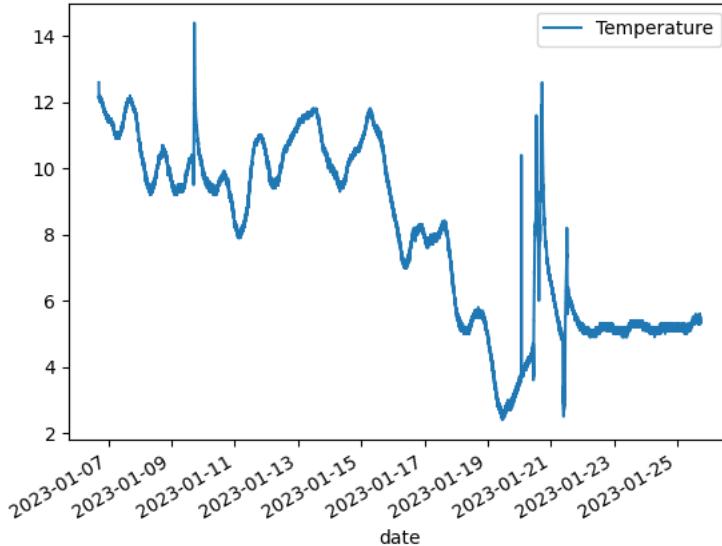
evaluateDataSet.py

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 url="https://raw.githubusercontent.com/TUBAF-IfI-LiaScript/"+ \
5     "VL_ProzeduraleProgrammierung/master/examples/"+ \
6     "12_Anwendungen/US_experiment/Datenset/"+ \
7     "distanceMeasurements_filtered.csv"
8
9 df=pd.read_csv(url, sep=',', header = 0)
10 # Einfügen einer "echten" Zeitspalte
11 df['date'] = pd.to_datetime(df['created_at'])
12 print(df.head(3))
13
14 df.plot(x="date", y="Temperature")
15
16 #plt.show()
17 plt.savefig('foo.png') # notwendig für die Ausgabe in LiaScript
```

```
      created_at ... date
0 2023-01-06T16:44:36+00:00 ... 2023-01-06 16:44:36+00:00
1 2023-01-06T16:45:17+00:00 ... 2023-01-06 16:45:17+00:00
2 2023-01-06T16:45:58+00:00 ... 2023-01-06 16:45:58+00:00
```

[3 rows x 5 columns]

foo.png



Welche Spitzenwerte in Bezug auf die Temperatur wurden erreicht? Zu welchen Zeitpunkten traten diese auf?

Fragekomplex 2: Welche Auswirkung hatte dies für die die Distanzmessung?

Die Schallgeschwindigkeit lässt sich nach der Gleichung

$$v_{schall}(m/s) = 331.3 + (0.606 \cdot T)$$

abschätzen. Dabei ist zu beachten, dass die Zeitmessung in *ns* erfolgte. Im Ergebnis wollen wir aber eine Ausgabe in *cm* realisieren.

evaluateDataSet.py

```
1 import pandas as pd
2
3 url="https://raw.githubusercontent.com/TUBAF-IfI-LiaScript/" + \
4     "VL_ProzeduraleProgrammierung/master/examples/" + \
5     "12_Anwendungen/US_experiment/Datenset/" + \
6     "distanceMeasurements_filtered.csv"
7
8 df=pd.read_csv(url, sep=',', header = 0)
9 df["US_speed"] = (331.3 + (0.606 * df["Temperature"])) / 1000 / 1000 * 100
10 df["US_distance"] = df["US_duration"]*df["US_speed"] / 2
11 print(df.head(3))
```

```
      created_at   Temperature ...   US_speed   US_distance
0  2023-01-06T16:44:36+00:00        12.6   ...  0.033894  145.911776
1  2023-01-06T16:45:17+00:00        12.4   ...  0.033881  144.605986
2  2023-01-06T16:45:58+00:00        12.3   ...  0.033875  142.496786

[3 rows x 6 columns]
```

Ausblick

Ein Blick zurück in den Oktober ... mit welchen Zielen/ Motivation waren wir gestartet:

Vorlesung I - 25.10.2022

- Anwendungssicht

Wir möchten Sie in die Lage versetzen einfache Messaufgaben mit einem Mikrocontroller zu realisieren und die Daten auszuwerten.

- Algorithmische Perspektive

Wir möchten Sie dazu erüchtigen den Algorithmusbegriff der Informatik zu durchdringen und anwenden zu können.

- Konzeptionelle Perspektive

Sie erlernen grundlegende Elemente der prozeduralen und der objektorientierten Programmierung.

- Umsetzungssicht

Wir vermitteln Grundkenntnisse in den Programmiersprachen C++ und Python.

Exkurs

Bislang programmieren wir unsere Anwendungen in einem klassischen Format. Wir schreiben den Code in eine Datei und führen diesen mit einem Mausklick in der Entwicklungsumgebung oder einem Aufruf im Terminal aus.

Insbesondere im Bereich der Datenverarbeitung und der künstlichen Intelligenz sind aber auch die Verwendung von Notebooks etabliert. Jupyter Notebook (früher IPython Notebooks) ist eine web-basierte interaktive Umgebung, mit der Jupyter-Notebook-Dokumente erstellt werden können. Dabei wird der Code in Blöcken aufgeteilt - die unabhängig von einander ausgeführt werden können. Ergänzt werden diese durch Kommentarblöcke, die wiederum in Markdown geschrieben werden können.

Achtung: Notebooks erlauben einen schnellen Einstieg in die Arbeit und eine gute Kommentierung der Codeblöcke, sind aber aus dem Blick des Entwicklungsprozesses kritisch zu sehen:

- Ausgabe und Code verschmelzen in einem Dokument. Dies macht die Nachvollziehbarkeit von Änderungen am Code aufwändig.
- Die beliebige Reihung der Ausführung während der Entwicklung führt dazu, dass Notebooks häufig nicht intuitiv von oben nach unten ausführbar sind.
- Notebooks selbst eignen sich nicht, um Code zu modularisieren.

Zusammenfassung

Was konnten / wollten wir dabei nicht leisten:

- Der Softwareentwicklungsprozess ist völlig unberücksichtigt geblieben.
- Sie haben die Programmiersprachen C++ und Python nur in den Grundkonzepten kennen gelernt.
- Die unterschiedlichen Programmiertechniken von C++ für Mikrocontroller und Desktopanwendungen wurden nicht berührt.
- ...

Wenn Sie Lust auf mehr "Software" haben, bietet die Veranstaltung *Softwareentwicklung* im kommenden Semester einen guten Ansatzpunkt!

Klausurvorbereitung

Gehen Sie bitte von folgenden Aufgabentypen für die Klausur aus:

- Korrektur eines Programmes: *Finden Sie Fehler im nachfolgenden Codebeispiel ...*
- Entwerfen eines Programmes / Ergänzen von Methoden: *Erweitern Sie den Code um eine Methode XY...*
- Analyse eines Programmes: *Welchen Wert gibt das Programm mit dem Erreichen der Zeile N aus?*
- Konzeptionelle Fragen: *Nennen Sie drei Ganzahldatentypen in C++.*

A1: Korrektur eines Programms

Für die Auswertung der Messergebnisse wird eine Funktion benötigt, die den größten Wert und seine Position in einem Array ermittelt. Es kann dabei davon ausgegangen werden, dass der Wert im Array nur einmal vorkommt. Die vorliegende Implementierung der Funktion sowie die dazugehörige main-Funktion enthalten syntaktische und logische Fehler, die es zu korrigieren gilt: Schreiben Sie das korrigierte Programm auf.

Aufgabe1.cpp

```
1 #include <iostream>
2
3 int maximum(double values,double& max_value)
4 {
5     for (i=0;i<anz;i++)
6     {
7         &max_value=values[0];
8         if (values[i]<max_value)
9         {
10             value[i]=&max_value;
11             i=pos;
12         }
13     }
14     return pos;
15 }
16
17 int main(void)
18 {
19     double werte[100];max;
20     int I;
21     for (i=0;i<100,i++) cin<<werte[i];
22     std::cout<<"Position: "<<maximum(werte,100,max)>>"\n";
23     std::cout<<"Maximum: "<<max
24     return 0;
25 }
```

```
main.cpp:22:52: error: stray '\' in program
22 |     std::cout<<"Position: <<maximum(werte,100,max)>>"\n";
|         ^
main.cpp:22:54: warning: missing terminating " character
22 |     std::cout<<"Position: <<maximum(werte,100,max)>>"\n";
|         ^
main.cpp:22:54: error: missing terminating " character
22 |     std::cout<<"Position: <<maximum(werte,100,max)>>"\n";
|         ^
main.cpp: In function 'int maximum(double, double&)':
main.cpp:5:8: error: 'i' was not declared in this scope
5 |     for (i=0;i<anz;i++)
|         ^
main.cpp:5:14: error: 'anz' was not declared in this scope
5 |     for (i=0;i<anz;i++)
|         ^
main.cpp:7:22: error: invalid types 'double[int]' for array subscript
7 |     &max_value=values[0];
|         ^
main.cpp:10:7: error: 'value' was not declared in this scope; did you mean 'values'?
10 |     value[i]=&max_value;
|     ^
|     values
main.cpp:11:9: error: 'pos' was not declared in this scope
11 |     i=pos;
|     ^
main.cpp:14:10: error: 'pos' was not declared in this scope
14 |     return pos;
|     ^
main.cpp: In function 'int main()':
main.cpp:19:21: error: 'max' was not declared in this scope; did you mean 'std::max'?
19 |     double werte[100];max;
|         ^
|         std::max
In file included from /usr/include/c++/12/string:50,
                  from /usr/include/c++/12/bits/locale_classes.h:40,
                  from /usr/include/c++/12/bits/ios_base.h:41,
                  from /usr/include/c++/12/ios:42,
                  from /usr/include/c++/12/ostream:38,
                  from /usr/include/c++/12/iostream:39,
                  from main.cpp:1:
/usr/include/c++/12/bits/stl_algobase.h:300:5: note: 'std::max' declared here
300 |     max(const _Tp& __a, const _Tp& __b, _Compare __comp)
|     ^
main.cpp:21:8: error: 'i' was not declared in this scope
21 |     for (i=0;i<100,i++) cin<<werte[i];
|     ^
main.cpp:21:21: error: expected ';' before ')' token
21 |     for (i=0;i<100,i++) cin<<werte[i];
|     ^
|     ;
main.cpp:21:23: error: 'cin' was not declared in this scope; did you mean 'std::cin'?
21 |     for (i=0;i<100,i++) cin<<werte[i];
|     ^
|     std::cin
/usr/include/c++/12/iostream:60:18: note: 'std::cin' declared here
60 |     extern istream cin;           /// Linked to standard input
|     ^
main.cpp:22:52: error: expected ';' before 'n'
22 |     std::cout<<"Position: <<maximum(werte,100,max)>>"\n";
|         ^
|         ;
```

```
main.cpp:20:7: warning: unused variable 'I' [-Wunused-variable]
20 |     int I;
|     ^
```

Machen Sie sich bewusst, dass Sie in der Klausur keine Möglichkeit haben, den Compiler "zu befragen". Üben Sie daher das lesen von Code. Tauschen Sie dazu Programme aus den Übungen untereinander aus und evaluieren Sie diesen.

A2: Erweitern eines C++-Programms

Die Klasse `Akte` enthält als Datenfelder die Angaben zum Aktenzeichen und der Laufzeit:

Aufgabe2.cpp

```
1 class Akte
2 {
3     private:
4     std::string aktenzeichen;
5     int laufzeit;
6 };
```

```
main.cpp:4:8: error: 'string' in namespace 'std' does not name a type
4 |     std::string aktenzeichen;
|     ~~~~~
main.cpp:1:1: note: 'std::string' is defined in header '<string>'; did you forget to '#include <string>'?
+++ |+#include <string>
1 | class Akte
```

Eine weitere von der Klasse `Akte` abgeleitete Klasse `AkteX` soll neben den bereits in der Klasse `Akte` enthaltenen Angaben einen Aktencode (eine dreistellige ganze Zahl) enthalten. Definieren Sie die Klasse `AkteX` und vervollständigen Sie die Klasse `Akte` um die Member, die folgende Anweisungen in der main-Funktion ermöglichen würden:

```
int main() {
    Akte akte1("08-15",2),
    akte2;
    std::cout << "Voraussichtliche Laufzeit: "<<akte1.getLaufzeit()<<"\n";
    akte1.setLaufzeit(4);
    akte2.ausgabe();
    AkteX aktex("08-16",99,816);
    aktex.ausgabe();
    std::cout << aktex.getAktencode()<<"\n";
}
```

- Die Funktion `ausgabe()` beider Klassen soll alle in einem Objekt gespeicherten Angaben auf dem Standard-Output ausgeben.
- In der Klasse `AkteX` sollen allerdings sowohl die Funktion `ausgabe` als auch die Funktion `getAktencode` statt des gespeicherten Wertes dessen Quersumme ausgeben bzw. liefern.
- Hinweis: bei den Objekten der Klasse `std::string` ist (anders als bei `char[]`) die Zuweisung mit `=operator` möglich.

A3: Kurze Fragestellungen

1. Markieren Sie alle Ganzahldatentypen in folgender Aufstellung.

- std::string
- short
- float
- bool

2. Welchen Inhalt gibt das folgenden Codefragment aus?

```
char text [] = "Dieser lange Text ergibt keinen Sinn";
for (int i=0;i<4;i=i+2) std::cout<<text[i];
std::cout<<text[5];
```

3. Schreiben Sie den Code in 2) als Python code.

```
1 text = "Dieser lange Text ergibt keinen Sinn"
2
```

4. Welches Konzept nutzt C++ um die Member einer Klasse vor dem Zugriff aus der anderen Klassen zu schützen.

5. Mit Hilfe welcher Methoden werden C++-Objekte erstellt?
- Destruktoren
 - Konstruktoren
 - private Methoden
 - new

Fragebögen

Für die Analyse des Erfolgs der Lehrveranstaltung bitten wir Sie um Ihr Feedback. Der gleich Fragebogen wurde bereit 2019 genutzt, um die Erwartungen und die Motivation der Studierenden mit dem Erreichten in Relation zu setzen.

Befragung zur Lehrveranstaltung „Prozedurale Programmierung“ / „Einführung in die Informatik“

Vielen Dank, dass Sie an der Befragung teilnehmen! Ihre Daten nutzen wir ausschließlich anonymisiert. Die Daten werden nur im Rahmen des Forschungsvorhabens genutzt und nicht an Dritte weitergegeben. Die Teilnahme an der Studie ist selbstverständlich freiwillig. Wenn Sie teilnehmen, gehen wir davon aus, dass Sie mit der Speicherung und Beforschung der erhobenen Daten einverstanden sind. Die Beantwortung des Fragebogens dauert etwa 10-15 Minuten.

Rahmendaten Studierende

1.) Bitte machen Sie zunächst ein paar Angaben zu Ihrer Person.

Geschlecht weiblich männlich divers

2.) Geburtsjahrgang _____ Bitte Jahreszahl eintragen.

3.) Welchen Studiengang studieren Sie? _____

4.) Im wievielten Semester studieren Sie? _____ Hochschulsemester _____ Fachsemester

Studium. Nachfolgend finden Sie eine Reihe von Aussagen, die sich auf Ihr Studium beziehen. Schätzen Sie bei jeder Aussage ein, inwieweit Sie dieser zustimmen.

5.) Ich lerne im Studium, ...

	stimme gar nicht zu	stimme voll zu			
... weil mir die Arbeit mit den Inhalten Spaß macht.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5
... weil ich mein Studium erfolgreich abschließen möchte.	<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5
... weil ich zu den Besten gehören möchte.	<input type="checkbox"/> A	<input type="checkbox"/> B	<input type="checkbox"/> C	<input type="checkbox"/> D	<input type="checkbox"/> E

Vielen Dank für Ihre Mitarbeit am Fragebogen!

Vielen Dank für Ihre Beteiligung während der Übungen und der Vorlesungen!