Ostfalia
Hochschule für angewandte Wissenschaften

Institut für Fahrzeugsystem- und Servicetechnologien
IFST

Laborscript

Labor 6
Informatik II

Vorlesung : Prof. Dr. B. Lichte
Labor : B.Eng. Rafael Shah
Tutor : Alexander Tank
Version : 1.1 vom 2. Februar 2021

Inhaltsverzeichnis

Labor 6

Inhalt:

- Abschlussaufgabe

Abschlussaufgabe

In der Abschlussaufgaben sollen Sie exemplarisch die Daten einer 100m-Sprint-Veranstaltung einer Schule auswerten. Hierfür ist es zunächst notwendig, dass Sie die Daten, welche Sie in der Datei *resources/Input.txt* finden, einlesen, um anschließend die Berechnungen durchführen zu können. Jede Zeile der Datei steht für die Leistung einer Schülerin bzw. eines Schülers. Im Folgenden sehen Sie die erste Zeile der Datei, welche die Interpretation der Daten aufzeigt:

Daten:	1.	M	17	160	13.24
Interpretation:	Laufende Nummer	Geschlecht	Alter	Körpergröße	100m-Zeit

Die aus der Datei *resources/Input.txt* einzulesenden Daten sind in einem `struct` mit dem Namen `Student_t` zu speichern. Das `struct` soll dabei aus den folgenden Komponenten bestehen:

- **Gender** (Geschlecht, M: Male oder F: Female)
- **Age** (Alter, Angabe in Jahren)
- **Height** (Größe, Angabe in cm)
- **Time** (100m-Zeit, Angabe in s)

Die laufende Nummer muss nicht mit abgespeichert werden. Die Namensgebung sollte auf Englisch erfolgen.

Aufbau:
Um die anschließende Bewertung zu vereinfachen und vergleichbarer zu machen, wird eine gewisse Grundstruktur des Programms in Form von Funktionen vorgegeben. Darüber hinaus können somit auch einfacher Teilpunkte vergeben werden, wenn das Gesamtprogramm nicht korrekt funktioniert, einzelne Programmteile jedoch korrekt implementiert sind. Die Funktionalität Ihres Programms soll daher auf die folgenden Funktionen aufgeteilt werden.

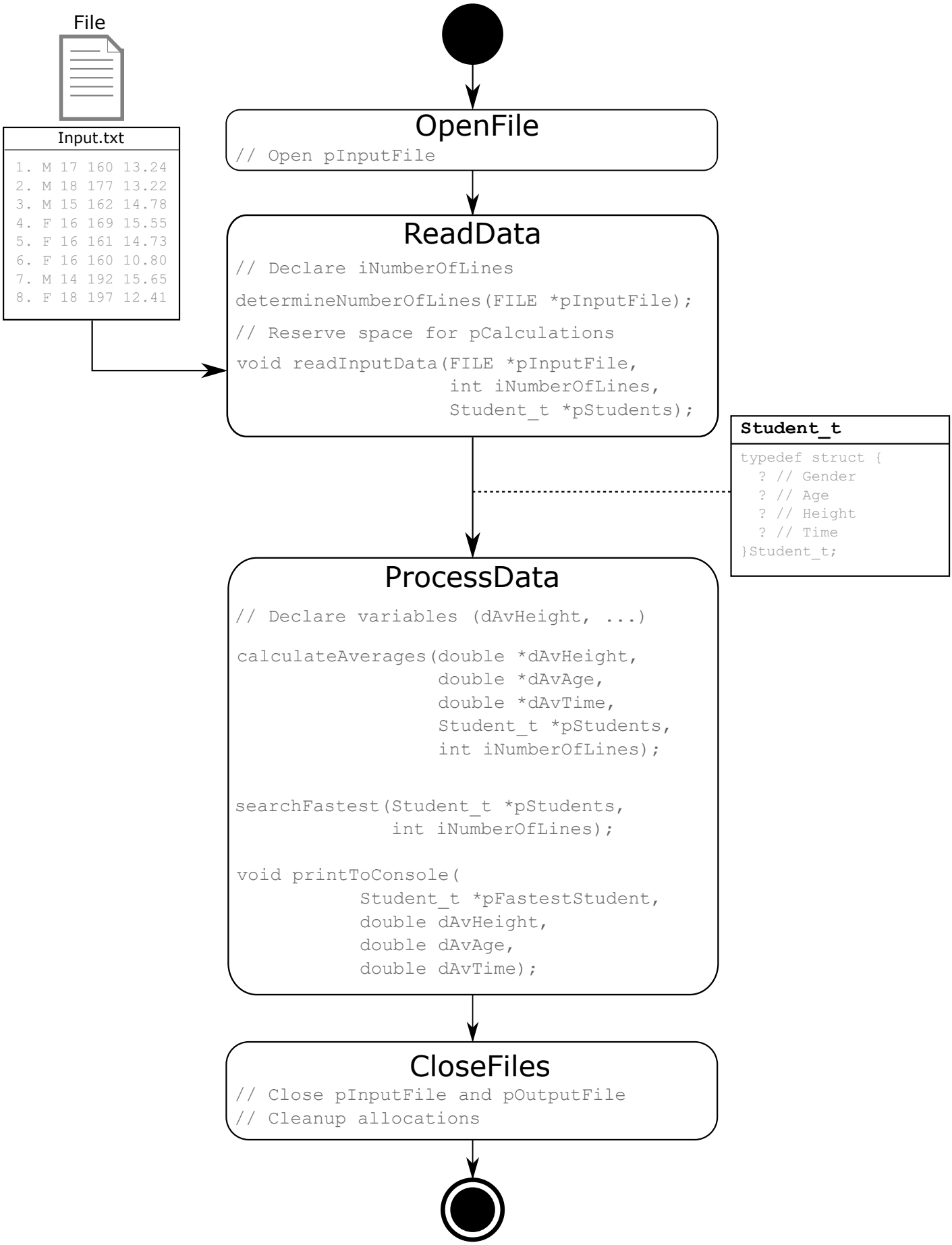
- `int determineNumberOfLines(FILE *pInputFile);`
  Der Funktion soll ein Filepointer `pInputFile` übergeben werden, über den die Funktion die Eingabedatei einließt und analysiert, wie viele Zeilen bzw. Leistungen von Schülern darin gespeichert sind. Dieser Wert soll als Ergebnis anschließend zurückgegeben werden.
- `void readInputData(FILE *pInputFile, int iNumberOfLines, Student_t *pStudents);`
  Der Funktion soll ein Filepointer `pInputFile` übergeben werden, aus der die Funktion insgesamt eine Anzahl von `iNumberOfLines` Datensätzen einließt und in ein Feld `*pStudents` des dafür zu erstellenden Datentyps `Student_t` abspeichert.

- `void calculateAverages(double *dAvHeight, double *dAvAge, double *dAvTime, Student_t *pStudents, int iNumberOfLines);`
  Nachdem die Daten der Schüler von Ihnen eingelesen wurde, sollen nun zunächst einige Mittelwerte der Daten berechnet werden. Zu berechnen sind die Durchschnittswerte der Größe, des Alters und der 100m-Zeit. Um diese Werte nachfolgend in Verlauf des Programmes verwenden zu können, werden sie der Funktion als Pointer übergeben (`*dAvHeight, *dAvAge & *dAvTime`). Als Informationsquelle werden mit dem Pointer `*pStudents` die Daten der Schüler und mit `iNumberOfLines` die Anzahl an Zeilen bzw. Leistungen von Schülern übergeben.

- `Student_t* searchFastest(Student_t *pStudents, int iNumberOfLines);`
  Der Funktion sollen die in `*pStudents` gespeicherten Daten und die Anzahl an Zeilen bzw. Leistungen von Schülern (`iNumberOfLines`) übergeben werden. Aus diesen Daten wird der schnellste Schüler bzw. die schnellste Schülerin ermittelt und ein Pointer auf diese(n) von der Funktion zurückgegeben. Der Rückgabewert soll in der `main`-Funktion entgegengenommen werden.

- `void printToConsole(Student_t *pFastestStudent, double dAvHeight, double dAvAge, double dAvTime);`
  Der Funktion werden die zuvor ermittelten Durchschnittswerte (`dAvHeight, dAvAge & dAvTime`) als auch ein Pointer auf den schnellsten Schüler bzw. die schnellste Schülerin `Student_t *pFastestStudent` übergeben. Die Durchschnittswerte als auch die gesamten Eigenschaften des schnellsten Schülers bzw. der schnellsten Schülerin sollen in der Konsole ausgegeben werden.

Diese Funktionen sollen aus der `main()`-Funktion nach folgendem Schema aufgerufen werden:



Ein- und Ausgabedateien:
Die Eingabedateie, die Ihr Programm verarbeiten soll, befindet sich im *resources/* Verzeichnis Ihrer Programmiervorlage.

Vorgehen:
Das strategische Vorgehen bei der Implementierung ist grundsätzlich Ihnen persönlich überlassen. An dieser Stelle sei aber erwähnt, dass es sinnvoll sein kann, die Funktion in der gegebenen Reihenfolge zu implementieren und ihre Implementierungen stets in kleinen Schritten zu überprüfen. Falls Sie die Funktion `void readInputData(FILE *pInputFile, int iNumberOfLines, Student_t *pStudents);` nicht implementieren können, finden Sie in dem bereitgestellten Dokument eine händisch erzeugte Datenstruktur, mit welcher Sie die weiteren Funktionen entwickeln und/oder testen können.

Hinweise:

1. Verwenden Sie in Ihrem Programm relative Pfade zu den Dateien.
2. Beachten Sie, dass Ihr Programm für eine beliebige Anzahl an Zeilen in *resources/Input.txt* funktionieren muss.
3. Beachten Sie die CodingConventions.
4. Zur Abgabe der Aufgabe exportieren Sie bitte die Datei wie im Labor gezeigt als ZIP-Archive-File. Bitte beachten Sie die Namensgebung der exportierten Datei:

   `WS2020_Name_Vorname_Matrikelnummer.zip`
5. Laden Sie das erzeugte Archive-File in den für Sie vorgesehenen Ordner im Stud.IP hoch.

Viel Erfolg!