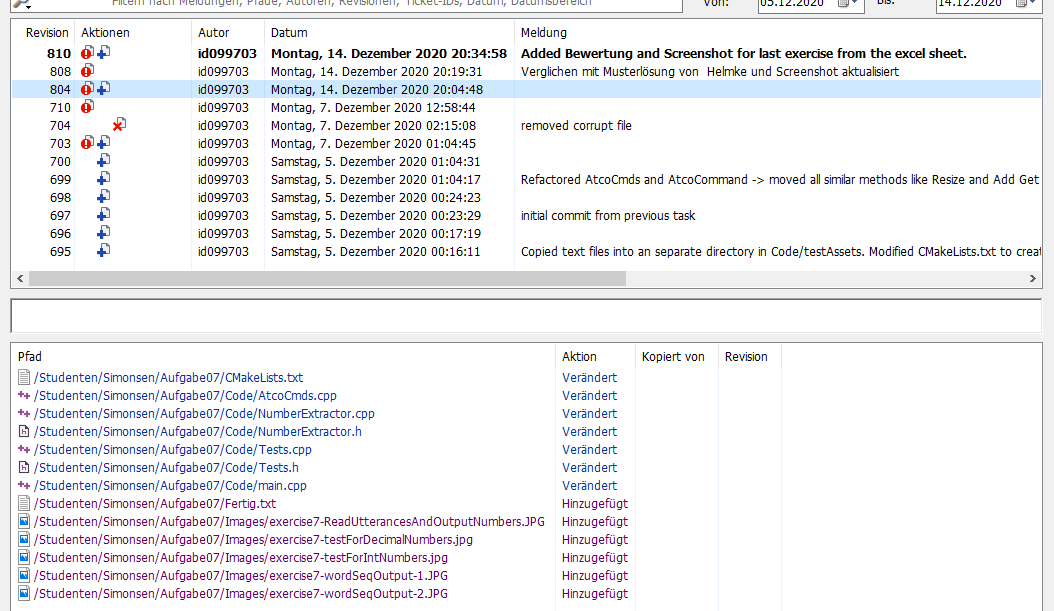
Feedback für Simonsen, 24.12.2020



Viele entscheidende Codedateien wurden nach dem Stichtag 13.12 noch hochgeladen.

Die Bewertungs-Datei im Csv Format fehlte. Vermutlich ist es die Datei mit der Endung txt.

Wurde aber nicht weiterverfolgt.

Projektdateien waren nicht vorhanden und auch keine CMake-Datei

|  |  |
| --- | --- |
| Operating System  (e.g. Windows, Linux) | Linux |
| IDE (e.g. Visual Studio, CLion …) | VS Code |

Es wurde ein Projekt von Helmke erstellt.

Programm nicht kompilierbar, da rekursives Include

DynArray.cpp

#include <stdexcept>

#include "DynArray.h"

Die Header-Datei enthält

};

#include "DynArray.cpp"

#endif //AUFGABE7\_DYNARRAY\_H

Include in cpp wurde entfernt und cpp vom Build ausgeschlossen.

Anschließend lässt sich das Projekt ohne Warnungen compilieren.

Das Programm sollte zwischen –test -Modus und normalen Modus unterscheiden.

Die Berechnung der Zahlen wird auch bei –test ausgeführt. Teil besser im else-Teil packen.

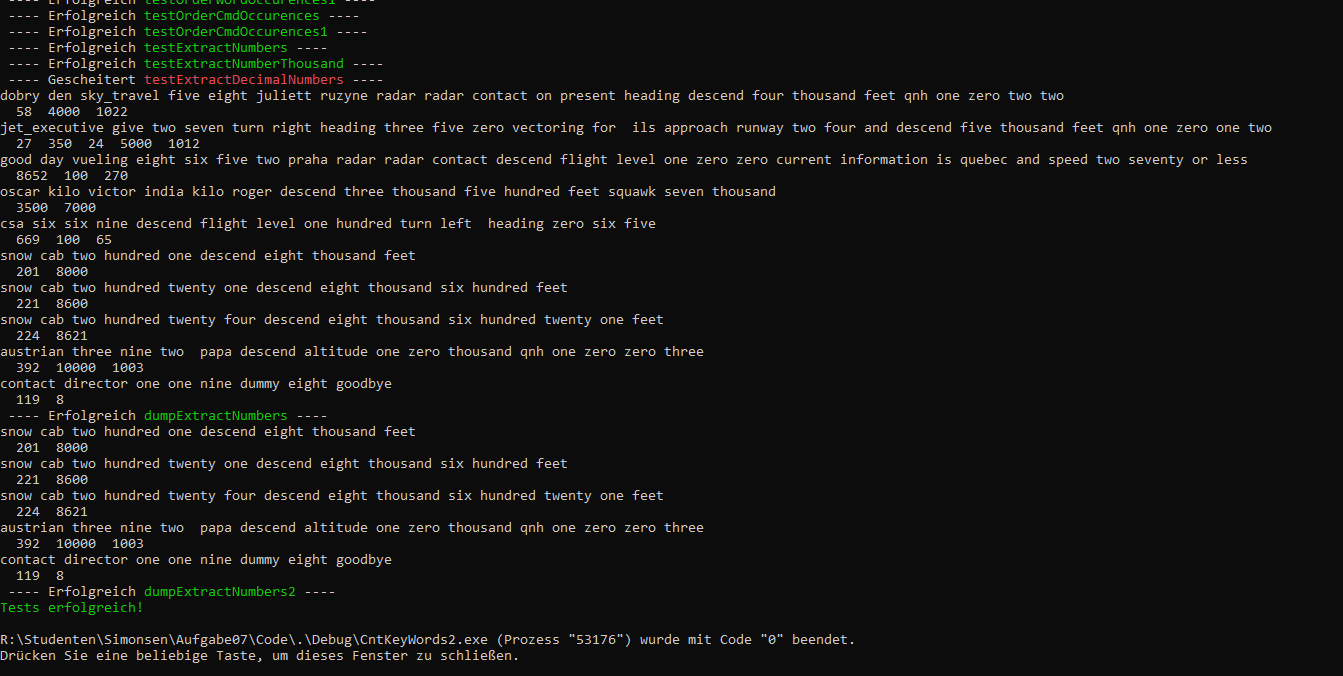
else

{

ReadUtterancesAndOutputNumbers();

}

Test-Mode funktioniert nicht wie gewünscht.



Es gibt einen kaputten Test und trotzdem wird am Ende Erfolg vermeldet????

Test scheitern an sich nicht, aber durch meine Erweiterung

std::string wordSeq3 = "contact director one one nine point zero two zero";

std::vector<double> expNumbers3 = { 119.020 };

bool b\_success3 = true;

numEx = testForDecimalNumbers(wordSeq3, expNumbers3, b\_success3);

return b\_success && b\_success2 && b\_success3;

Die Ausgabe ist hier 119.20. Schön wäre, wenn in diesem Fall auch eine Bildschirmausgabe erfolgen würde, was denn falsch ist, damit man schneller ins Debugging kommt

Keine Punkte dafür abgezogen.

Die Tests sollten kommentiert werden, damit man sieht, was sie machen (sollen). Das gilt auch für andere Code-Teile. Die Zahlen-Extraktion ist ausreichend in den Funktionen kommentiert

Programm stützt ab, wenn weniger Zahlen geliefert werden als erwartet.

Beweis, habe, die 180 ergänzt. Dann Buff.

std::string wordSeq4 = "on qnh one thousand twenty eight reduce one six zero knots";

std::vector<int> expNumbers4 = {1028, 160, 180};

bool b\_success4 = true;

numEx = testForIntNumbers(wordSeq4, expNumbers4, b\_success4);

Ich habe mir die Funktionalität nicht im Detail angesehen.

Im Prinzip gute Lösung, natürlich mit vielen Detailproblemen.

Es wurden nur 5 Punkte, statt 10 Punkte, wie eigentlich erforderlich, für die fehlende csv-Datei abgezogen.