

Die verschiedenen Programmierparadigmen von C++

Rufzeichen aus Flugfunk-Wortsequenzen automatisch extrahieren

20% der Punkte für einfache aufeinander aufbauende Programmieraufgaben, die am Vorlesungsstoff orientiert sind

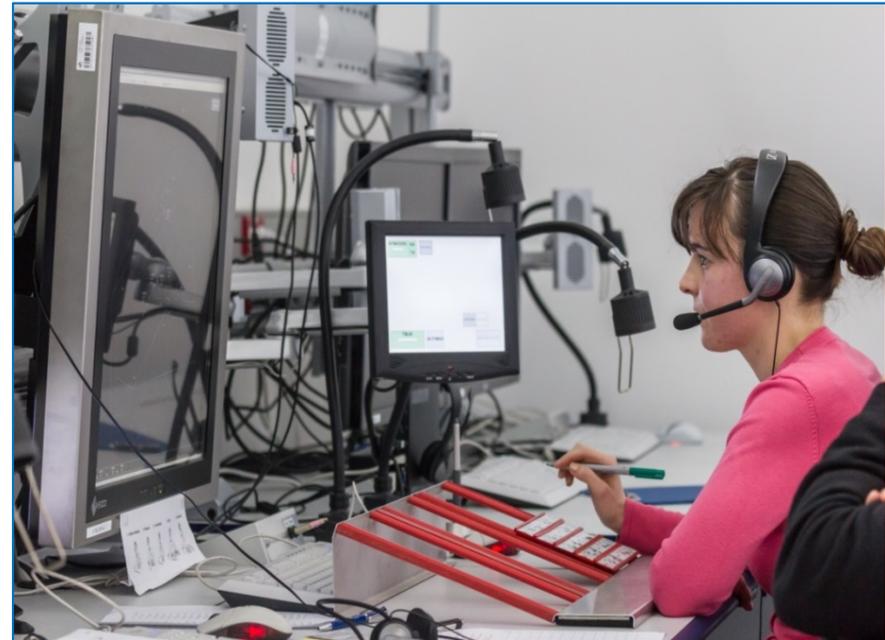
Nochmals bis zu 10 Zusatzpunkte für die Klausur für die letzte Aufgabe



Punktevergabe

Ich stelle in der Vorlesung Teile der Lösung vor. Sie probieren Ihren Code auf meinen bisher unbekanntem Testdaten aus und nennen mir Ihr Ergebnis per SVN.

Damit gibt es die Punkte. Bei einigen von Ihnen werde ich den Code aus dem SVN auf meinen Daten ausführen.



Was bedeutet automatische Semantik – Interpretation?

Wortsequenz:

austrian five nine eight zulu speed three hundred knots
direct whisky whisky nine eight one

Semantik:

AUA598Z SPEED 300 kt
AUA598Z DIRECT_TO WW981 none



Sie bekommen (über meine Homepage)

2. Vorlesung; Fr. 04.10.2024 oder Alternativtermin in Folgewoche

[Link zur Clicker](#)

Vorlesung

Software

[Wiederholung / Ankündigung \(21.09.2023\)](#)

[Funktionen \(12.08.2023\)](#) [benutzerdefinierte Datenstrukturen \(12.08.2023\)](#) [Aufteilung von Code auf Dateien \(12.08.2023\)](#)

[Ausgangscod f](#)
[Lösung für Stru](#)

Übungsaufgaben WS 2024/25; Levenshtein-Berechnung, Abgabe bis ??? über SVN

Lösungen und :

Einführung in die Aufgabenstellung der großen Übung [Air Traffic Controllers and challenge of manual input \(21.09.2024\)](#)

[Automatic Speech Recognition \(ASR\) \(21.09.2024\)](#)

[Literatur zu den Aufgaben \(21.09.2024\)](#)

[Exercise 1: Levenshtein Distance Calculation \(21.09.2024\)](#)

[Aufgabenübersicht \(nur grob, Details bei Aufgaben selbst\) \(21.09.2024\)](#)

[Kurzeinführung zur Installation und Nutzung von TortoiseSVN an der FH Ostfalia \(21.09.2023\)](#)

Übungsaufgaben WS 2020/21 (immer noch aktuell)

Lösungen

Aufgabe 1

[Einführung in die Aufgabenstellung der großen Übung Air Traffic Controllers and challenge of man Automatic Speech Recognition \(ASR\) \(21.09.2024\)](#)

[Literatur zu den Aufgaben](#) (21.09.2024)

[Exercise 1: Levenshtein Distance Calculation](#) (21.09.2024)

[Aufgabenübersicht \(nur grob, Details bei Aufgaben selbst\)](#) (21.09.2024)

[Kurzeinführung zur Installation und Nutzung von TortoiseSVN an der FH Ostfalia](#) (21.09.2023)

Implement a class Levenshtein to calculate the Levenshtein distance

- Word sequence 1: swiss one [redacted] nine two mike
- Word sequence 2: swiss one two nine [redacted] mike

Levenshtein distance is 2

```
sky travel five seven juliett dobry den praha radar radar contact descend one hundred  
[redacted] [redacted] ryan air juliett dobry den praha radar radar contact descend one hundred  
[redacted] [redacted]
```

Levenshtein distance is 4

```

class Levenshtein
{
public:
    // astr1
    Levenshtein(
        const std::vector<std::string>& astr1, // first string compared to astr2
        const std::vector<std::string>& astr2) // i.e. LD between
                                                // astr1 and astr2 is calculated

    //! returning the Levenshtein distance between astr1 and astr2
    int CalcLevenshteinDistance();
    std::string backtrace() const;
    std::string GetPrettyPrint(int ai_ld, std::string astr_goldText="",
        std::string astr_recognText="") const;
    /* maybe other methods */

private:
    // first index runs over size of mstr1
    int Get(int st1Ind, int st2Ind) const {
        return (mpi_mat[st1Ind * mi_spCnt_n2 + st2Ind]);
    }
    std::vector<std::string> mstr1;
    std::vector<std::string> mstr2;
    int mi_zCnt_m1; // size of mstr1
    int mi_spCnt_n2; // size of mstr2
    // contains the matrix, used for LD calculation as a vector
    int mpi_mat[1000]; // later we create a dynamic matrix, currently sizes
                        // of astr1 and astr2 are limited
    /* maybe other members */
};2

```

Für die Matrix einen
1-dim Vektor verwenden

Aufgabe 1a: CalcLevenshteinDistance bauen

// We calculate the Levenshtein distance of Tier and Tor
which should be 2

```
bool LevenshteinDistTierTor()
{
    vector<string> s1{ "T", "i", "e", "r" };
    vector<string> s2{ "T", "o", "r" };
    Levenshtein dist(s1, s2);
    return 2 == dist.CalcLevenshteinDistance();
}
```

Aufgabe 1b: Methode backtrace bauen

```
bool LevenshteinDistRenTierTiere()
{
    vector<string> s1{ "R", "e", "n", "T", "i", "e", "r" };
    vector<string> s2{ "T", "i", "e", "r", "e" };
    Levenshtein dist(s1, s2);
    cout << "Needed steps: " << dist.backtrace();
    return (4 == dist.CalcLevenshteinDistance());
}
```

The expected output of the `cout` command to the screen should be:

Needed steps: Del Del Del Equ Equ Equ Equ Ins

We first have three deletions to transform "Rentier" to "Tiere". Then we have four equal strings and then we have a insertion.

Aufgabe 1c: GetPrettyPrint bauen

```
bool LevenshteinDistWords()
{
    vector<string> s1{ "Tango", "ind", "echo", "romeo" };
    vector<string> s2{ "Ind", "echo", "romeo", "abba" };
    Levenshtein dist(s1, s2);
    auto ld = dist.CalcLevenshteinDistance();
    cout << dist.GetPrettyPrint(ld, "s1", "and s2") << "\n";
    return 3 == ld;
}
```

```
s1       : Tango ind echo romeo
and s2     : Ind      echo romeo abba // LD: 3, gold words: 4
```

Der Code für GetPrettyPrint ist schon angegeben.

Regeln

Sie dürfen die Aufgabe auch zu zweit bearbeiten.

Stellen Sie selber sicher, dass jeder aus der Gruppe etwas von der Bearbeitung der Aufgabe hat.

Mögliche weitere Aufgaben

Aufgabe 2:

Verwendung von Arrays dynamischer Größe und Code auf verschiedene Dateien aufteilen

Aufgabe 3:

Klassen anstatt Strukturen nutzen und Kopier-Konstruktor und Zuweisungs-Operator (tiefe und flache Kopie in C++)

Weitere Aufgaben (so der Plan)

Aufgabe 4:

Extraktion von Rufzeichen aus Wortfolgen

Aufgabe 5:

Effizientere Extraktion von Rufzeichen mit der schnellsten und der besten Implementierung

Es ist ausdrücklich erwünscht, dass Sie die Aufgabe in einer größeren Gruppe (z.B. 4 bis 5) GEMEINSAM bearbeiten. Das macht die Sache nicht unbedingt einfacher.

Ich werde (zusammen mit Ihnen) dafür Sorge tragen, dass JEDE(R) aus der Gruppe/Team einen Anteil zur Aufgabelösung beiträgt bzw. nur die Aktiven mit Punkte „versorgen“.