

PROGRAMMIEREN II

DHBW Stuttgart Campus Horb INF2017

(HAUS)AUFGABE ZU DIESER ÜBERNÄCHSTE WOCHE

- Programmiert ein Schachspiel bei denen Abwechselnd beide Spieler ihre Züge machen
- Modelliert dabei die Figuren als Klassen, die ihr eigenes Move-Set haben
- Nutzt die bekannten Techniken zur Modellierung
- Eingabe der Züge erfolgt über die Kommandozeile (Beispiel: Erstes Kommando selektiert Figur, zweites gibt das Zielfeld an)
- Die Kommandozeile gibt dabei das Log für das Spiel an

- Wie hat es geklappt?
- Wer möchte Vorstellen? (2 Personen)

AGENDA

- File-Streams
- Programmieraufgaben zur Vertiefung

KEINE SORGE: NÄCHSTE WOCHE
GIBT ES WIEDER THEORETISCHE
DRUCKBETANKUNG



FILE STREAMS

- Package `<fstream>` (Standardlib)
 - `ofstream` —> Output-Operationen
 - `ifstream` —> Input-Operationen
 - `fstream` —> Beides :-)

OFSTREAM EXAMPLE

```
// basic file operations
#include <iostream>
#include <fstream>
using namespace std;
```

```
int main () {
    ofstream myfile;
    myfile.open ("example.txt");
    myfile << "Writing this to a file.\n";
    myfile.close();
    return 0;
}
```

OFSTREAM

- Vergleichbar zum geschrieben in den Outputstream
- Gleiche Stream-Operatoren
- Ofstream wird automatisch mit schreibendem Modus geöffnet

LESEN VON EINEM (ZEICHEN- STREAM)

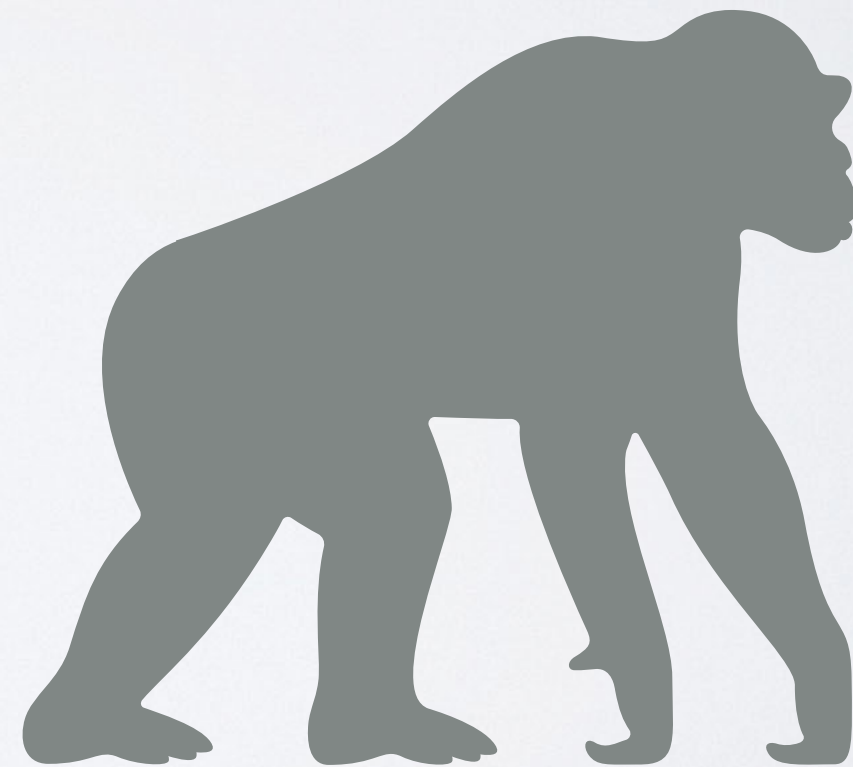
```
// reading a text file
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
```

```
int main () {
    string line;
    ifstream myfile ("example.txt");
    if (myfile.is_open())
    {
        while ( getline (myfile,line) )
        {
            cout << line << '\n';
        }
        myfile.close();
    }
```

```
    else cout << "Unable to open file";
```

```
    return 0;
}
```

- Close nie vergessen!



MODI BEIM ÖFFNEN VON STREAMS

ios::in

→ Input mode

ios::out

→ Output mode

ios::binary

→ Binary Mode (write, read instead of getLine...)

ios::ate

→ at the end

ios::app

→ appending

ios::trunc

→ truncating

AUFGABE I

Aufgabenstellung:

1. Erweitere deine Klassen- / Struct-basierte Hero App so, dass du die angelegten Helden in einer Datei speichern kannst.
2. Bei Start der Applikation sollen diese Helden wieder geladen werden.

**** Achtet darauf alle Streams zu schließen und eventuellen Speicher wieder frei zu geben! ****

KONVERSION STRING ZU INT ODER FLOAT

```
std::stof(myString)  
std::stoi(myString)
```

AUFGABE 2

Aufgabe 2

- * Erstelle ein Objekt, welches auf die Zeilen aus der beigefügten CSV und lese diese ein
- * Beantworte die folgenden Fragen:
 - * Wie viele Verbrechen gab es in Sacramento in 2006?
 - * Was war das häufigste Verbrechen?
 - * Welches ist der gefährlichste District? (gemessen an der Menge der Vorfälle?)
 - * Welche Adressen kamen mehr als einmal vor?