

Programmieren 1

Auditorium Exercise 11

Weiterer Verlauf Übungen

	Ausgabe	Abgabe	Besprechung
Ü10	16.12.	22.12.	09.01.–12.01.
Ü11	22.12.	12.01.	13.01.–19.01.
Ü12	13.01.	19.01.	23.01.–26.01.
Ü13 (optional, Klausurvorbereitung)	20.01.	-	-
letzte Vorlesung 27.01.			

Erinnerung: Programmieren I Winter Challenge

- Aufgabe
 - möglichst guten Reversi-Spieler programmieren
 - 1 sec pro Zug (sonst disqualifiziert, siehe Template)
 - C-Template & Details: `WinterChallenge.{zip|pdf}` auf Stud.IP
- Abgabe bis 15.1.
 - Quelltext (eine C-Datei)
- Preise
 - Urkunde und Buchpreis für den besten Computer-Reversi-Spieler
- Auswahl
 - Vorrunde: zufällige Paarung, spielen gegeneinander, sammeln Punkte
 - Top 16 (Achtelfinale) spielen gegeneinander beim Reversi-Turnier
 - Termin: letzte Semesterwoche (23.-27.1.), wird noch bekannt gegeben

Das Reversi-Programm darf nicht (z.B. mit einem Server) kommunizieren und darf nur selbst entwickelte Komponenten enthalten (außer der prog1lib und der C-Standardbibliothek).

Klausur

- Zeitraum: 14.03. – 16.03.
- Im Rechnerraum: Hauptgebäude F411
 - Hier finden auch die Präsenztutorien statt
- Zeitslots über die 3 Tage verteilt
 - Zuteilung erfolgt demnächst per StudIP

Assignment 11

BINÄRBÄUME IN DATEIEN

Write Tree

```
// Writes a tree to a file. Returns true if the operation was successful.
bool write_tree_rec(FILE* file, Tree* tree) {
    require_not_null(file);
    if (tree == NULL) {
        if (!write_byte(file, 0)) return false;
    } else {
        if (!write_byte(file, 1)) return false;
        if (!write_int(file, tree->value)) return false;
        if (!write_tree_rec(file, tree->left)) return false;
        if (!write_tree_rec(file, tree->right)) return false;
    }
    return true;
}
```

Read Tree

```
// Reads a tree from a file. Returns true if the operation was successful.
bool read_tree_rec(FILE* file, /*out*/Tree** tree) {
    require_not_null(file);
    require_not_null(tree);
    char c;
    if (!read_byte(file, &c)) return false;
    if (c == 0) {
        *tree = NULL;
        return true;
    }
    if (c != 1) return false;
    Tree* t = new_leaf(0);
    *tree = t;
    if (!read_int(file, &t->value)) return false;
    if (!read_tree_rec(file, &t->left)) return false;
    if (!read_tree_rec(file, &t->right)) return false;
    return true;
}
```

Assignment 12

- Last mandatory assignment
- Already available on StudIP
- We will have a brief look inside now

Void- and Function Pointers

LIVE SESSION