Leibniz Universität Hannover

Programmieren 1

Auditorium Exercise 12

Human-Computer Interaction Group

Tim Dünte

programmieren1@hci.uni-hannover.de

Reversi Winterchallenge

Alle Teilnehmer der Winterchallenge:

- Wir machen ein Doodle
- Der Doodle Link wird per Email verschickt.
- Appelstraße 9A, 9. Etage, Seminarraum

Ablauf:

- Turnier & Siegerehrung
- gemütliches Zusammensitzen mit ein paar Keksen und Getränken

Question & Answer Slots with your Tutors

Tutorsprechstunden enden zum 01.02.2019 !!!

Tutor	Day	Time	Room
Zena Obeidi	Monday	09:30 – 10:00	F111
Lukas Köhler	Monday	09:00 – 10:00	F111
Oguz Önbas	Tuesday	13:30 – 14:00	F411
Lars Wrenger	Tuesday	15:30 – 16:00	F411
Jan Meyer	Tuesday	12:00 – 12:30	F411
Kendall Ly	Tuesday	16:00 – 16:30	F411
Eklekta Kristo	Tuesday	13:30 – 14:00	F411
Tobias Schoon	Tuesday	08:00 – 09:00	F411
Polina Geisler	Tuesday	11:30 – 12:00	F411
Dominik Töllner	Wednesday	12:15 – 12:45	F111
Patric Plattner	Wednesday	12:15 – 12:45	F111
Hoang Kiet Ta	Wednesday	09:00 – 09:30	F411
Kerem Demir	Wednesday	09:00 – 10:00	F411
Nils Grünefeld	Wednesday	19:00 – 20:00	F411
Alexander Koch	Thursday	15:00 – 15:30	F111
Valeri Estin	Friday	14:00 – 14:30	F111

LernLounge

Unterstützung von Studierenden beim Lernen durch studentische Tutorinnen und Tutoren in der InfoLounge

- Die Tutorinnen und Tutoren sind zu festen Zeiten Ansprechperson für fachliche Fragen sowie für Fragen des selbstorganisierten Lernens
- Die Unterstützung erfolgt nach dem Prinzip der minimalen Hilfe

Fächer-Schwerpunkte

- **Programmieren**
- Grundlagen der Theoretischen Informatik
- Analysis
- Lineare Algebra
- Grundlagen digitaler System
- Elektrotechnische Grundlagen

Start ab Montag den 29.10.18 !!!

Koordination: Isabelle Kross, Tel.: 0511-762 3744, Email: isabelle.kross@et-inf.uni-hannover.de

LernLounge Termine

Stundenplan LernLOUNGE WS 2018/19

Zeit	Mo	Di	Mi	Do	Fr
9.00 - 10:00				Malte	
10:00 – 10:30		Alexander		Malte	
10:30 – 11:00	Malte	Alexander		Malte	
11:00 – 12:00	Malte	Alexander		Malte	
12:00 – 12:30	Malte, Jan	Alexander			Jan
12:30 – 13:00	Jan	Alexander			Jan
13:00 – 13:30	Jan, Jakob	Julian		Lennart: Voraus-sichtlich ab 15.01.2019	Jan
13:30 – 14:00	Jakob	Julian			Jan
14:00 – 15:00	Jakob	Julian			Jan
15:00 – 16:00	Jakob	Julian			

Schwerpunktfächer

Alexander
- Programmieren
- GTI

Jakob
- Elektro. Grundlagen
- GDS

Lennart
- Programmieren
- Lineare Algebra

Julian
- GTI
- Lineare Algebra

Jan
- Lineare Algebra
- Analysis

Malte
- Programmieren
- GTI
- GDS

Koordination: Isabelle Kross, Tel.: 0511-762 3744, Email: isabelle.kross@et-inf.uni-hannover.de

Programmieren 1

Where to go for slides, assignments, ...

Website:

http://hci.uni-hannover.de/teaching/winter18-Prog1

Go here for general information

Stud.IP:

https://studip.uni-hannover.de

Go here for slides (e.g., this ones), assignments, and the forum

Handing in assignments:

https://assignments.hci.uni-hannover.de/WiSe2018/Programmieren1

Go here to submit your assignments

Prog1Lib

More information about the Prog1Lib:

- http://hci.uni-hannover.de/files/prog1lib/index.html

Lectures

#	Date	Topic
1	19.10.	Organization, computers, programming, algorithms, PostFix introduction (execution model, IDE, basic operators, booleans, naming things)
2	26.10.	PostFix (primitive types, functions, parameters, local variables, tests), recipe for atomic data
3	2.11.	PostFix (operators, array operations, string operations), recipes for enumerations, intervals, and itemizations
4	9.11.	Recipes for compound and variant data, iteration and recursion, PostFix (loops, association arrays, data definitions)
5	16.11.	C introduction (if, variables, functions), Programming I C library
6	23.11.	Data types, infix expressions, C language (enum, switch, while)
7	30.11.	Compound and variant data, C language (formatted output, loops, struct, union)
8	7.12.	C language (arrays, pointers) arrays: fixed-size collections, linear and binary search
9	14.12.	Dynamic memory (malloc, free), recursion (recursive data, recursive algorithms)
10	21.12.	Linked lists, binary trees, game trees, minimax algorithm
11	11.1.	C language (program structure, scope, lifetime, linkage), function pointers, pointer lists
12	18.1.	Objects, object lists, binary trees, search trees
13	25.1.	Dynamic data structures (stacks, queues, maps, sets), iterators, documentation tools
14	1.2.	This and that, C language (remaining C keywords)

Assignment 11

Leibniz Universität Hannover

Aufgabe 1a

```
// Student* -> Student*
// Create a copy of student.
void* copy_student(void* x) {
        Student* s = (Student*)x;
        return new_student(s->name, s->sex, s->mat_nr, s->debts);
}

// Student* -> void
// Releases memory of a student.
void free_student(void* x) {
        if (x != NULL) {
                Student* s = (Student*)x;
                free(s->name);
                free(s->sex);
                free(s);
        }
}
```

Assignment 11

Leibniz Universität Hannover

Aufgabe 1b

```
// Student* -> bool
// Returns true if the amount of debts is above 20000.
bool poor_student(void* element, int i, void* x) {
        Student* s = (Student*)element;
        return s->debts > 20000.0;
}

// find first poor student, use find_list
// print result, if any
printsln("first poor student:");
Student* found_student = find_list(list, poor_student, NULL);
if (found_student != NULL) {
        String s = student_to_string(found_student);
        printsln(s);
        free(s);
}
```

Assignment 11

Leibniz Universität Hannover

Aufgabe 1c

```
// Student* -> bool
// Returns true if the matriculation number is shorter than *x.
bool mat_nr_less_digits_than(void* element, int i, void* x) {
   Student* s = (Student*)element; // Cast of void* to Student*, to get
access to attributes of student
   int* a = (int*)x; // Cast of void* to double* to get the necessary
parameter for compare
   int counter = 0; // initialization of counter variable
   for(int i = s->mat_nr; i > 0; i = i / 10 )// i gets the value of mat_nr. i
is divided by 10 until i is zero.
      counter++; // counter is incremented
   return (counter < (*a)); // return true if the matriculation number
(counter) of student is shorter (smaller) than given value from pointer a. }
```

Assignment 11

Leibniz Universität Hannover

Aufgabe 1c

```
printsln("first student with mat_nr digit count less than max_digits:");
int max_digits = 8
found_student = find_list(list, mat_nr_less_digits_than, &max_digits);
if (found_student != NULL) {
        String s = student_to_string(found_student);
        printsln(s);
        free(s);
}
```

Assignment 11

Leibniz Universität Hannover

Aufgabe 1d

```
// Student* -> String
// Maps a student to its name.
void* student_name(void* element, int i, void* x) {
        Student* s = (Student*)element;
        return s->name;
}

// map the list of students to a list of student names
printsln("student names:");
Node* names = map_list(list, student_name, NULL);
println_list(names, string_to_string);
free_list(names, NULL);
```

Assignment 12

- Will be available around 4 pm
- Last necessary assignment
  - Submission of assignment 12 until 24.01.19
  - Tutorials of assignment 12 until 31.01.19
- We will have a brief look inside now
  - … and then continue with some live programming on similar challenges

Good luck and have fun

^_^

programmieren1@hci.uni-hannover.de