

Programmieren in C

Wiederholungen

Dr. Adrian Koller, Büro E307, adrian.koller@hslu.ch

Was genau sind «argc», «argv»?

«count» /Anzahl

«values» /Werte

```
int main(int argc, char *argv[] )
```

→ Commandline arguments!
Beim Aufruf des Programms mitgeben

```
prgc@prgcvm: ~/PRGCMo.../2021FS.../40_Exercise.../04_CLA_Operators
prgc@prgcvm:~/PRGCMo.../2021FS.../40_Exercise.../04_CLA_Operators$ ./logicOperators
Usage:
logicOperators Input1 Input2 Input3
prgc@prgcvm:~/PRGCMo.../2021FS.../40_Exercise.../04_CLA_Operators$ ./logicOperators 12 14 23
12, 14, 23
23 is the greatest
12 is the smallest
The sum is: 49
The average is: 16.33
prgc@prgcvm:~/PRGCMo.../2021FS.../40_Exercise.../04_CLA_Operators$
```

Switch?

Lucerne University of
Applied Sciences and Arts

**HOCHSCHULE
LUZERN**

Technik & Architektur
FH Zentralschweiz



- `40_Exercises/11_switch/coffeeMaker.c`

Deklaration vs. Implementierung

- Deklaration:
→ Information für den Compiler

“Es gibt da etwas, das “heisst so” (→ Name) und/oder benötigt diese und jene Art von Input (→ Signatur)

- Implementierung:
→ Festlegung der «Funktion» des Objekts

«Detaillierte Anleitung, wie eine gewisse Funktionalität umgesetzt werden soll.»

- `40_Exercises/08_modular/greetings.c`

Instanzieren??

```
1 // *declaration* of a struct
2 struct Car {
3     char make[20];
4     char model[20];
5     float ps;
6     int year;
7 };
8
9 // *instances* with initializations
10 struct Car myPrivatCar = { .make= "Fiat", .model = "Panda", .ps = 75.5, .year = 1998};
11 struct Car companyCar = { .make= "VW", .model = "Transporter", .ps = 175.5, .year = 2008};
12
13
14 int main() {
15     return 0;
16 }
17 }
```

Fehlersuche in einem Programm?

- Compiler Error:
 - Compiler Output lesen!!
 - wichtig: evtl. eine Zeile vorher schauen
- Linker Error:
 - Linker Output genau lesen!
- Funktionsfehler (macht nicht, was es soll...)
 - Schritt für Schritt, auskommentieren.
 - mit gdb debuggen

Bit Funktionen

- Wahrheitstabelle → händische Beispiele

A	B	$A \& B$	$A B$	$A \wedge B$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0