

**1**

# **Programmieren**

## **.c-Dateien – hello.c**

```
# include <stdio.h>
# include "hello2.h"

int main (int argc, char *argv[])
{
    printf ("Hello World\n");
    printf ("Die Kommandozeilenargumente:\n");

    for (i = 0; i < argc; i++) {
        hello (i, argv[i]);
    }

    return 0;
}
```

## **.h-Dateien – hello2.h**

```
# ifndef hello2_H
# define hello2_H
# include <stdio.h>

typedef int Integer;
extern void hello (Integer nr, char *argument);
extern Integer i;

# ifdef DEBUG
# define DPRINTF(str) \
    fprintf (stderr, "Datei %s, Zeile %d: %s\n", \
        __FILE__, __LINE__, str);
    /* __FILE__ und __LINE__ sind vordefinierte Macros, die
     * den Namen der übersetzten Datei und die Zeilennummer
     * an der das Macro benutzt wird enthalten.
     */
# else
# define DPRINTF(str) /* ignored */
# endif
# endif
```

## **.c-Dateien – hello2.c**

```
# include <stdio.h>
# include "hello2.h"

Integer i = 0;

void hello (Integer nr, char *argument)
{
    DPRINTF ("Aufruf von hello");
    printf (" - %2d: %s\n", nr, argument);
}
```

## Kommandos

```
cc -c hello.c
cc -c hello2.c
cc -o hello hello.o hello2.o
hello
```

```
hello.c ⇒ hello.o
hello2.c ⇒ hello2.o
hello.o hello2.o libc.a ... ⇒ hello
Hello World....
```

---

```
cc -DDEBUG -c hello.c
cc -DDEBUG -c hello2.c
cc -DDEBUG -o hello hello.o hello2.o
```

## Makefile

```
SRC_H    =          hello2.h
SRC_C    = hello.c  hello2.c
OBJ      = hello.o  hello2.o
MAIN     = hello

CC        = gcc
X_DEBUG   = -g -Wall -pedantic
X_CFLAGS  = -ansi $(X_DEBUG) $(CFLAGS)

all: $(MAIN)

$(MAIN): $(OBJ)
⇒ $(CC) $(X_CFLAGS) -o $(MAIN) $(OBJ)

.SUFFIXES: .c .o
.c.o:
⇒ $(CC) $(X_CFLAGS) -c $*.c

clean:
⇒ rm -f $(OBJ) $(MAIN) core

hello.o:  hello.c hello2.h
hello2.o: hello2.c hello2.h
```

Das Zeichen ⇒ steht für ein Tabulator-Zeichen.