

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt

1.2. Entwicklungsumgebung Qt Creator

Anlegen eines neuen C++-Projekts, zunächst noch ohne Qt:

The screenshot shows the Qt Creator interface. The 'Datei' menu is open, with 'Neu...' selected. The 'Neue Datei oder neues Projekt' dialog box is displayed, showing the 'Projekt ohne Qt' option selected in the left pane and 'Reine C++-Anwendung' selected in the right pane. A red arrow points to the 'Reine C++-Anwendung' option.

Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster

Neu... Strg+N

Datei oder Projekt öffnen... Strg+O

Öffne Datei mit...

Zuletzt bearbeitet

Zuletzt bearbeitet

Sitzungen

Sitzungen verwalt

Projekt schließen

Alle Projekte und

Speichern

Speichern unter...

Alles speichern

Wiederherstellen

Schließen Strg+W

Alle schließen Strg+Umschalt+W

Andere schließen

Alle außer Sichtbare schließen

Drucken Strg+P

Neue Datei oder neues Projekt

Vorlage: Alle Vorlagen

Projekte

Anwendung

Bibliothek

Anderes Projekt

Projekt ohne Qt

Projekt importieren

Reine C-Anwendung

Reine C++-Anwendung

Erzeugt eine einfache C++-Anwendung, die qmake, CMake oder Qbs zum Erstellen verwendet.

Unterstützte Plattformen: Desktop

Auswählen... Abbrechen

1.3. Entwicklungsumgebung Qt Creator

Reine C++-Anwendung

Projektverzeichnis

Erzeugt eine einfache C++-Anwendung, die qmake, CMake oder Qbs zum Erstellen verwendet.

Name: Prog1

Erzeugen in: U:\CaX\ Auswählen...

Als Vorgabe für Projektordner verwenden

Pfad
Build-System
Kits
Zusammenfassung

Reine C++-Anwendung

Build-System auswählen

Build-System: qmake

Weiter Abbrechen

Pfad
Build-System
Kits
Zusammenfassung

Reine C++-Anwendung

Kit-Auswahl

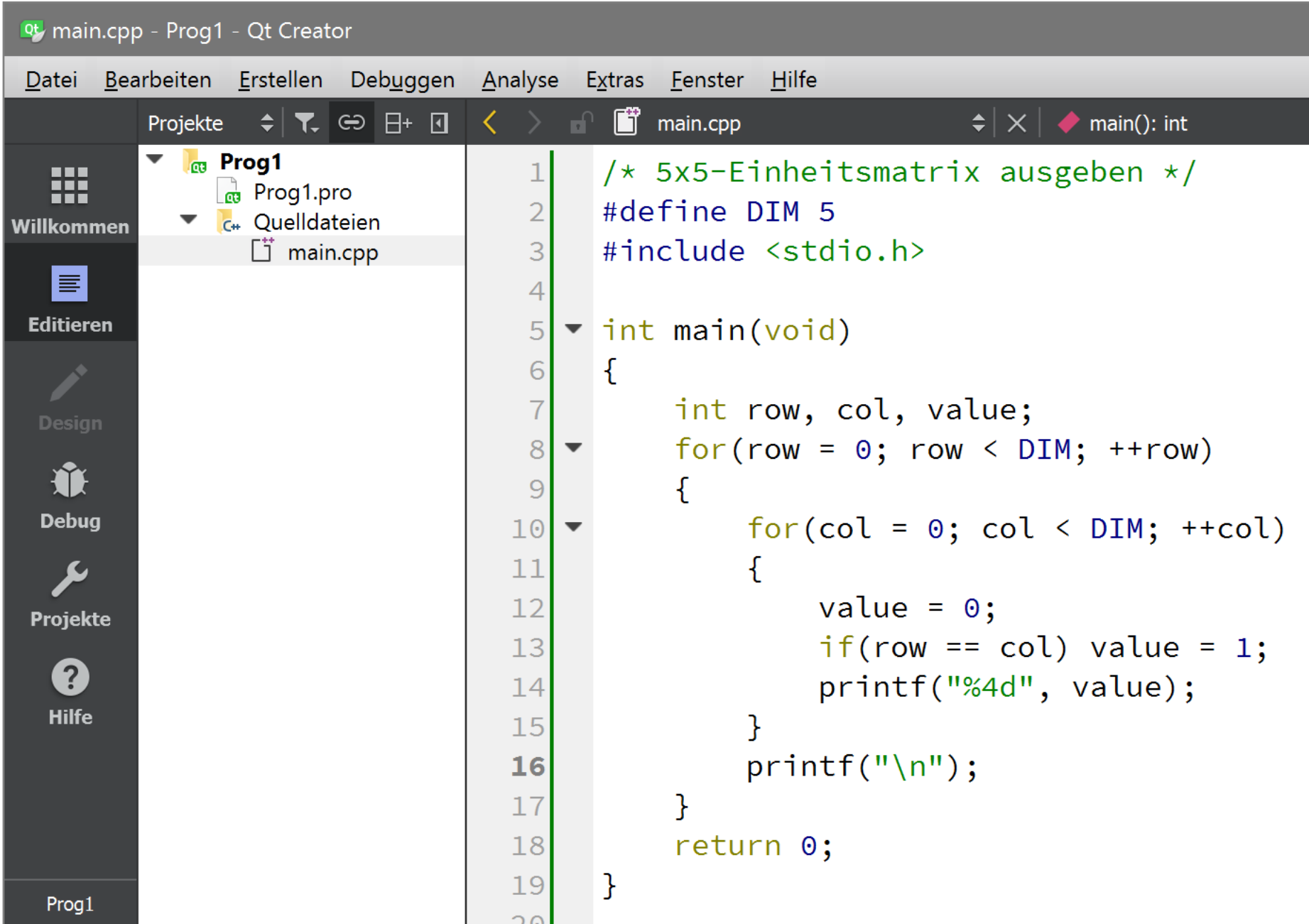
Qt Creator kann für das Projekt **Prog1** die folgenden Kits verwenden:

Alle Kits auswählen

Desktop Qt 5.8.0 MinGW 32bit Details ▼

Weiter Abbrechen

1.4. Entwicklungsumgebung Qt Creator



The screenshot displays the Qt Creator IDE interface. The title bar shows 'Qt main.cpp - Prog1 - Qt Creator'. The menu bar includes 'Datei', 'Bearbeiten', 'Erstellen', 'Debuggen', 'Analyse', 'Extras', 'Fenster', and 'Hilfe'. The toolbar shows 'Projekte', navigation icons, and the current file 'main.cpp' with a function signature 'main(): int'. The left sidebar contains icons for 'Willkommen', 'Editieren', 'Design', 'Debug', 'Projekte', and 'Hilfe'. The main editor area shows the following C++ code:

```
1  /* 5x5-Einheitsmatrix ausgeben */
2  #define DIM 5
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int row, col, value;
8      for(row = 0; row < DIM; ++row)
9      {
10         for(col = 0; col < DIM; ++col)
11         {
12             value = 0;
13             if(row == col) value = 1;
14             printf("%4d", value);
15         }
16         printf("\n");
17     }
18     return 0;
19 }
```

1.5. Entwicklungsumgebung Qt Creator

The screenshot shows the Qt Creator IDE interface. The title bar reads "Qt main.cpp - Prog1 - Qt Creator". The menu bar includes "Datei", "Bearbeiten", "Erstellen", "Debuggen", "Analyse", "Extras", "Fenster", and "Hilfe". The "Erstellen" menu is open, displaying the following options:

- Alles erstellen (Strg+Umschalt+B)
- Projekt "Prog1" erstellen (Strg+B)
- qmake ausführen
- Datei "main.cpp" erstellen (Strg+Alt+B)
- Vollständiges Deployment durchführen
- Deployment des Projekts "Prog1" durchführen
- Alles neu erstellen
- Projekt "Prog1" neu erstellen** (highlighted with a yellow mouse cursor)
- Alles bereinigen
- Projekt "Prog1" bereinigen
- Erstellen abbrechen
- Ausführen (Strg+R)
- Ausführung ohne Deployment
- Kit-Auswahl für Erstellung und Ausführung öffnen...

The left sidebar contains icons for "Willkommen", "Editieren", "Design", "Debug", "Projekte", and "Hilfe". The main editor area shows a code snippet:

```
main(): i  
en >  
  
++rc  
IM;  
  
alue  
alue)
```

The status bar at the bottom shows line numbers 16 and 17, and the code `printf("\n");`.

1.6. Entwicklungsumgebung Qt Creator

The screenshot shows the Qt Creator interface with a context menu open over the 'Projekte' view. A yellow box highlights the text *(rechte Maustaste)*. The context menu includes options like 'Projekte', 'Erstellen', 'qmake ausführen', 'Deployment', 'Ausführen', and 'Neu erstellen'. A second yellow box contains the text: *Wenn die Bearbeitung eines Projekts beendet ist, sollte das Projekt „geschlossen“ werden. Es wird dann nicht mehr in der Projektliste im Qt Creator angezeigt.*

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt

1.8. Ein- und Ausgabe

```
/* Ein- und Ausgabe in C */
```

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    float f;
```

```
    double d;
```

```
    char str[100];
```

```
    printf("int eingeben: " ); scanf("%d" , &i );
```

```
    printf("float eingeben: " ); scanf("%f" , &f );
```

```
    printf("double eingeben: "); scanf("%lf" , &d );
```

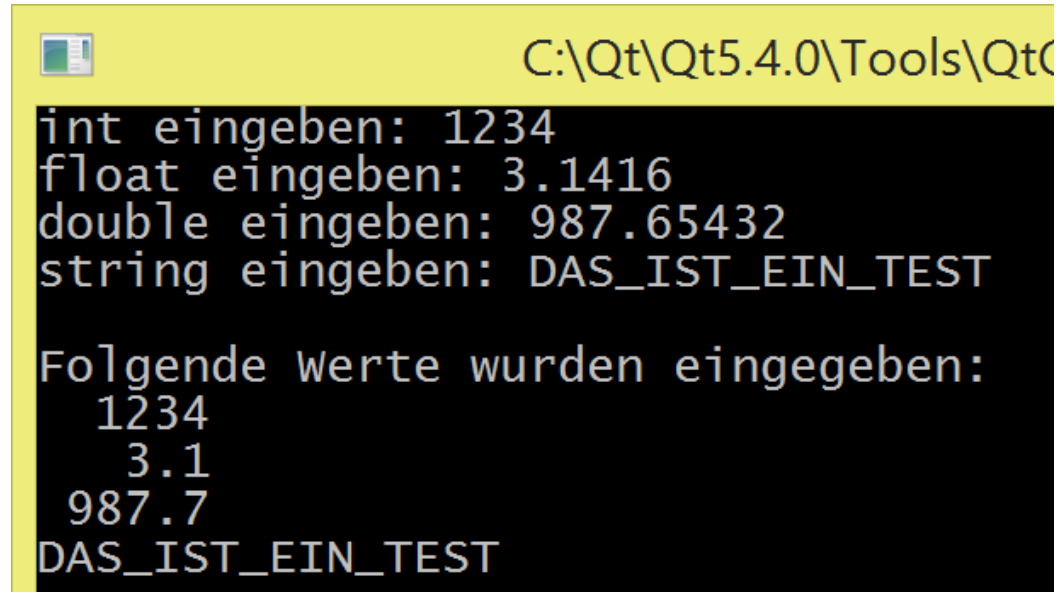
```
    printf("string eingeben: "); scanf("%99s", str);
```

```
    printf("\nFolgende Werte wurden eingegeben:\n");
```

```
    printf("%6d \n%6.1f \n%6.1f\n%s\n\n", i, f, d, str);
```

```
    return 0;
```

```
}
```



```
C:\Qt\Qt5.4.0\Tools\QtC
int eingeben: 1234
float eingeben: 3.1416
double eingeben: 987.65432
string eingeben: DAS_IST_EIN_TEST

Folgende Werte wurden eingegeben:
 1234
  3.1
 987.7
DAS_IST_EIN_TEST
```

*Tipp: Hilfsblatt zum printf-Befehl
auf <http://kuepper.userweb.mwn.de/>*

Ausgabeformat festlegen, Nachkommastellen, tabellarische Ausgabe:

`printf(" ...Formatbeschreibung... ", Liste, auszugebender, Variablen);`

```
/* Kosinustabelle ausgeben */
```

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

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    double x, y;
```

```
    printf("Kosinus-Tabelle\n\n");
```

```
    printf( ... (1) ... ?? );
```

```
    for(x = -M_PI; x <= M_PI; x += M_PI/4)
```

```
    {
```

```
        y = cos(x);
```

```
        printf( ... (2) ... ?? );
```

```
    }
```

```
    return 0;
```

```
}
```

Hausaufgabe!

Kosinus-Tabelle

x	y
-3.14	-1.00
-2.36	-0.71
-1.57	0.00
-0.79	0.71
0.00	1.00
0.79	0.71
1.57	0.00
2.36	-0.71
3.14	-1.00

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt

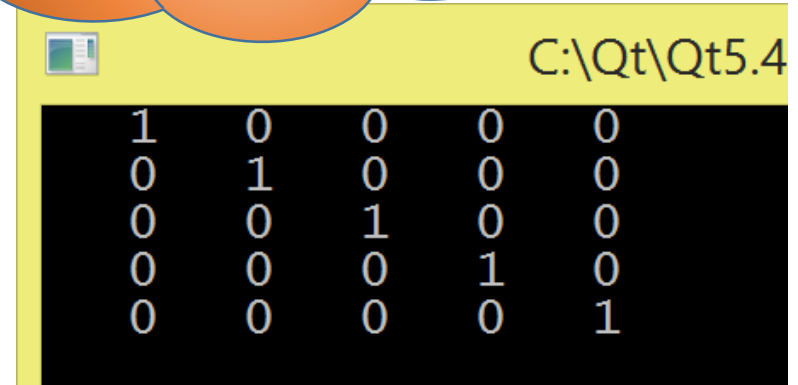
1.11. Kontrollstrukturen

Aufgabe: Zeichnen Sie ein Struktogramm, welches den genauen Ablauf der Funktion **main** beschreibt.

```
/* 5x5-Einheitsmatrix ausgeben */
#define DIM 5
#include <stdio.h>

int main(void)
{
    int row, col, value;
    for(row = 0; row < DIM; ++row)
    {
        for(col = 0; col < DIM; ++col)
        {
            value = 0;
            if(row == col) value = 1;
            printf("%4d", value);
        }
        printf("\n");
    }
    return 0;
}
```

Hausaufgabe!



```
C:\Qt\Qt5.4
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

Tipp 1: Wikipedia-Artikel zu Struktogrammen

Tipp 2: Kontrollstrukturen in C und MATLAB auf <http://kuepper.userweb.mwn.de/>

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt

Aufgabe: Definieren Sie die Funktion **qsolve** zum Lösen quadratischer Gleichungen.

```
/* Quadratische Gleichung */
#include <stdio.h>
#include <math.h>

double x1, x2; /* glob. Variablen! */
int qsolve(double p, double q);

int main(void)
{
    double p, q;
    printf("x^2 + px + q = 0\n\n");
    printf("p eingeben: "); scanf("%lf", &p);
    printf("q eingeben: "); scanf("%lf", &q);

    if(qsolve(p, q) != 0)
        printf("Reelle Nullstellen bei %.2f und %.2f\n\n", x1, x2);
    else
        printf("Keine reellen Nullstellen!\n\n");
    return 0;
}
```



Hausaufgabe!



*Tipp: Ingenieurinformatik 1,
Einleitung zum Praktikum Nr. 4*

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt



*...siehe
Praktikum!*

Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

Modulare Programmierung

Mehrere Quelldateien in einem Projekt

```
/* Rückgabe mehrerer Ergebnisse (ohne globale Variablen) */
```

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

```
#include <math.h>
```

```
int qsolve(double p, double q,  
           Ort einer double-Variablen o1, Ort einer double-Variablen o2);
```

```
int main(void)
```

```
{  
    double p, q, nst1, nst2;
```

```
    printf("x^2 + px + q = 0\n\n");
```

```
    printf("p eingeben: "); scanf("%lf", &p);
```

```
    printf("q eingeben: "); scanf("%lf", &q);
```

```
    if(qsolve(p, q, Ort von nst1, Ort von nst2) != 0)
```

```
        printf("Reelle Nullstellen bei %.2f und %.2f\n\n", nst1, nst2);
```

```
    else
```

```
        printf("Keine reellen Nullstellen!\n\n");
```

```
    return 0;
```

```
}
```

```
int qsolve(double p, double q,  
           Ort einer double-Variablen o1, Ort einer double-Variablen o2)
```

```
{
```

```
    double w = p*p/4.0 - q;
```

```
    if(w < 0) return 0;
```

```
    Wert am Ort o1 = -p/2.0 - sqrt(w);
```

```
    Wert am Ort o2 = -p/2.0 + sqrt(w);
```

```
    return 1;
```

```
}
```


Kapitel 1

Wiederholung C-Programmierung

Entwicklungsumgebung Qt Creator

Ein- und Ausgabe

Kontrollstrukturen, Verzweigungen, Schleifen

Funktionen, lokale und globale Variablen

Felder und Matrizen

Zeiger und Adressen

*...siehe
Praktikum!*

Modulare Programmierung

Mehrere Quelldateien in einem Projekt