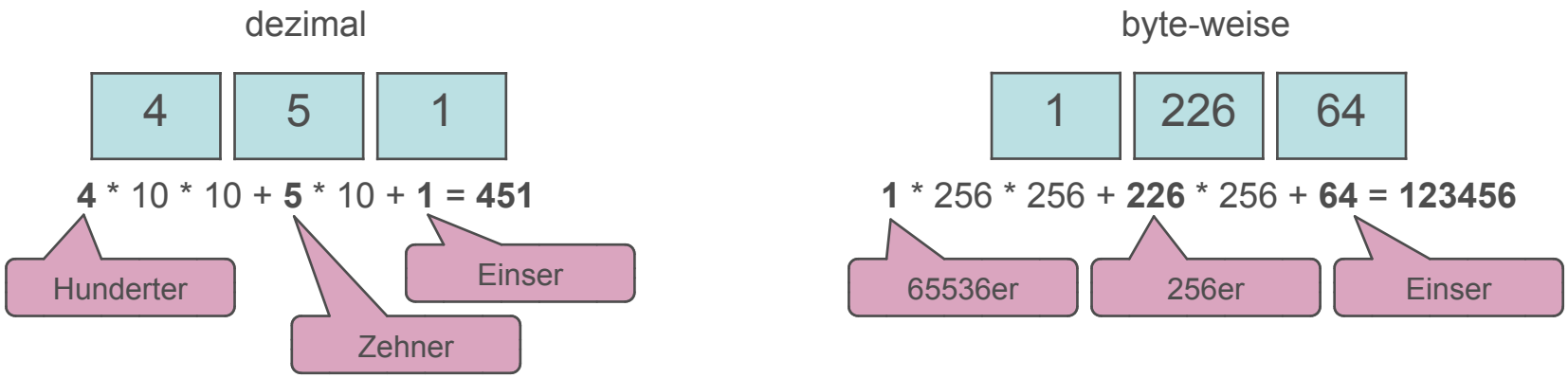
Teil 1
Grundlagen

- Arbeitsspeicher
- Programmaufbau
- IDE
- Variablen
- Datentypen
- Eingabe & Ausgabe
- Dateien
- Operatoren & Bedingungen

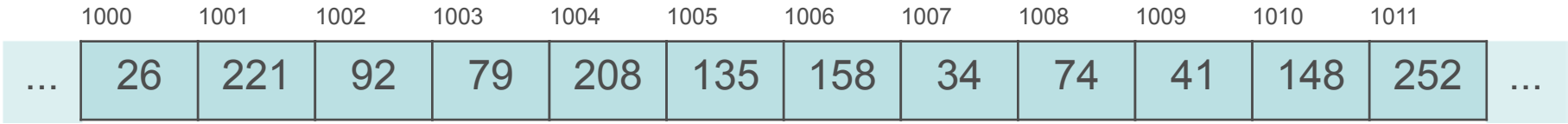
Bytes

- Bytes bestehen (normalerweise) aus 8 bit
- Bytes können 256 verschiedene Werte aufnehmen, z.B. 0 bis 255 oder -128 bis +127
- kleinste Einheit, mit der ein Computer umgehen kann
- mehrere Bytes lassen sich zusammenfassen, um größere Werte aufzunehmen, z.B. 256 x 256 = 65536 mit 2 Bytes (16 bit)



Arbeitsspeicher

- byteweise fortlaufend nummeriert (Adresse)
- anfangs im Prinzip zufällig initialisiert



- Bytes können alles mögliche bedeuten
- entscheidend ist, wie diese Bytes *interpretiert* werden (Kodierung)

Arbeitsspeicher

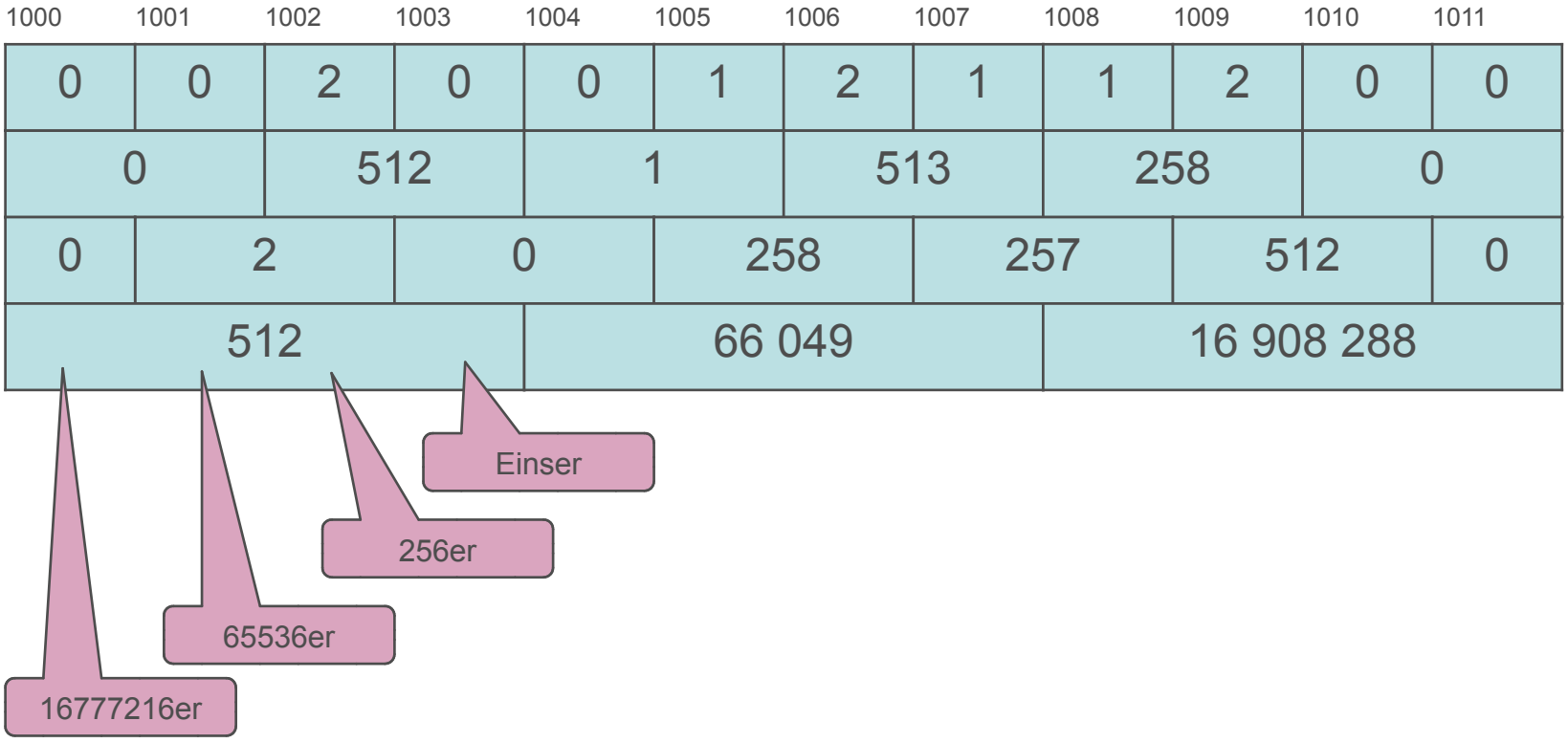
- Bytefolgen können z.B. Text repräsentieren (ASCII, UTF-8 etc.)

1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011
67	43	43	32	38	32	65	83	67	73	73	0
C	+	+		&		A	S	C	I	I	

32		64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h

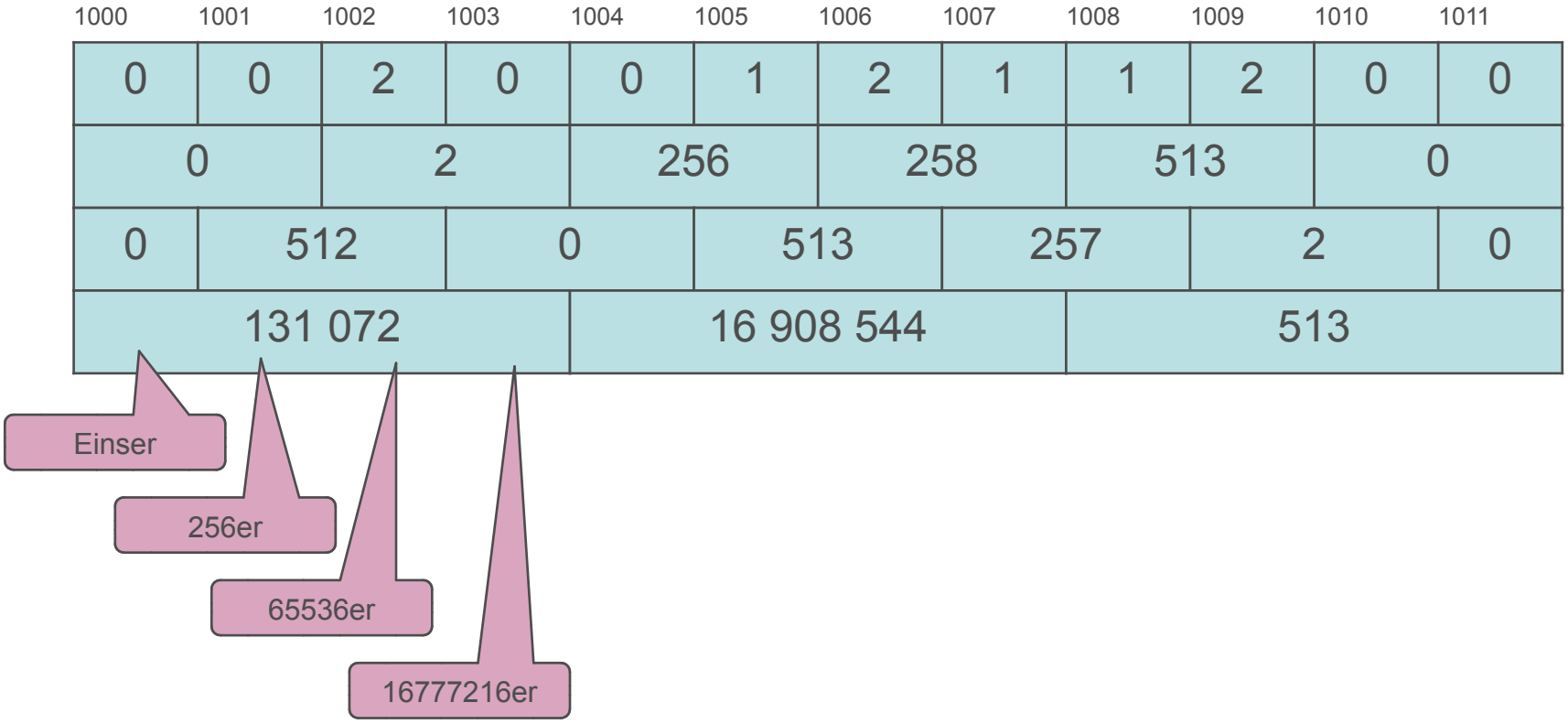
Arbeitsspeicher

- Bytefolgen lassen sich praktisch beliebig interpretieren



Arbeitsspeicher

- „little endian“ (Intel) statt „big endian“ (Motorola)



Programm

```
int main()
{
   return 0;
}
```

```
int main(int argc, char* argv[])
{
   return 0;
}
```

- Dateiendung: cpp (z.B. main.cpp)

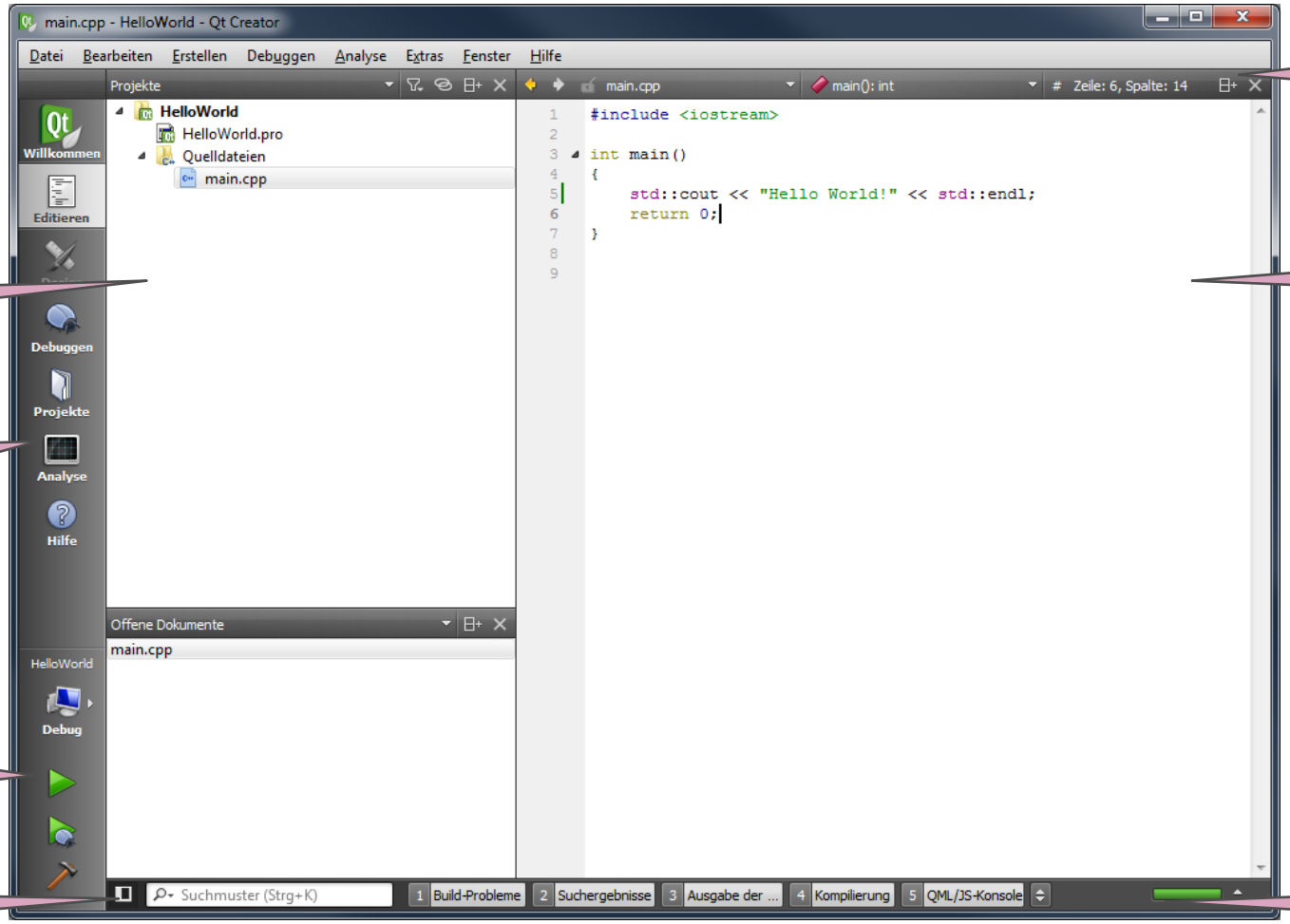
Kommentare

```
// bis zum Zeilenende

/* mehr-
zeilig */
```

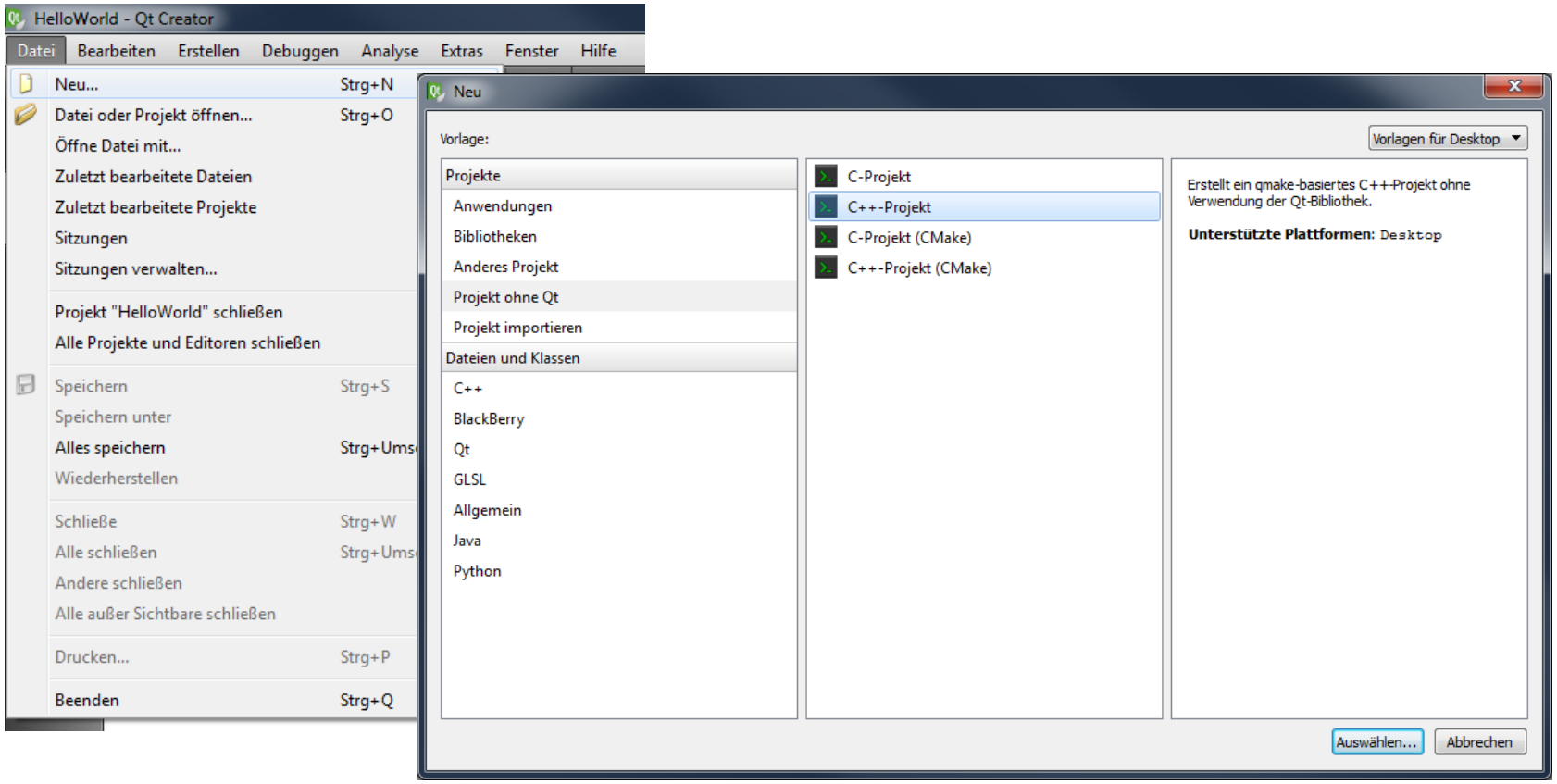
- Intension ausdrückend, d.h. erklärend, nicht beschreibend

Qt Creator


main.cpp - HelloWorld - Qt Creator
Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe
Projekte
HelloWorld
HelloWorld.pro
Quelldateien
main.cpp
Offene Dokumente
main.cpp
main.cpp
main(): int
Zeile: 6, Spalte: 14
#include <iostream>
int main()
{
std::cout << "Hello World!" << std::endl;
return 0;
}
Willkommen
Editieren
Debuggen
Projekte
Analyse
Hilfe
HelloWorld
Debug
Suchmuster (Strg+K)
1 Build-Probleme
2 Suchergebnisse
3 Ausgabe der ...
4 Kompilierung
5 QML/JS-Konsole
Navigation
Projekt
Quellcode
Modi
Aktionen
Suche
Konsolen

neues Projekt

- Qt Creator
- Datei ⇨ Neu ⇨ Projekt ohne Qt ⇨ C++-Projekt

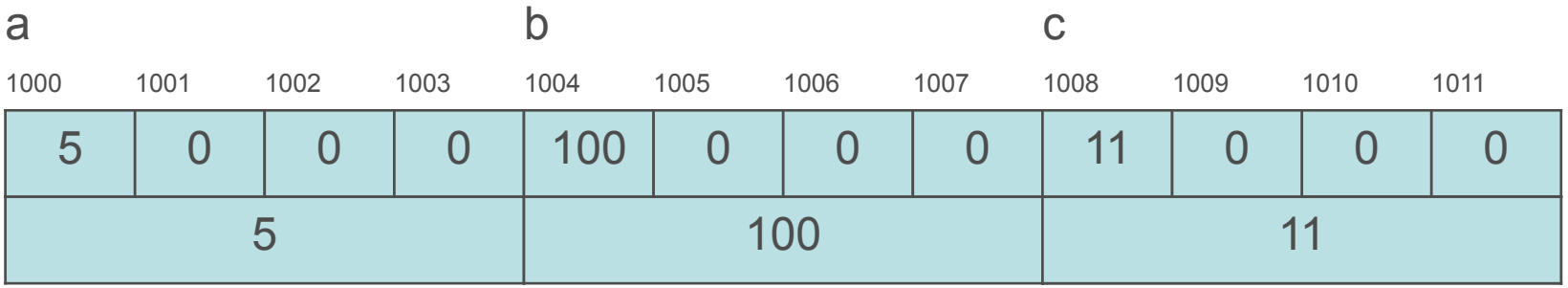


Variablen

- Variablen repräsentieren Arbeitsspeicheradressen

```
int a = 5;      // a: 1000
int b = 100;    // b: 1004
int c;          // c: 1008
c = 11;         // c: 1008
```

zufälliger Wert



Datentypen

```
int count = 12;
double area = 17.2;
std::string text = "hallo"; // C++
bool correct = true;
char letter = 'X';

count = count + 1;
area = area * 2.0;
correct = false;
```

```
char text[] = "hallo"; // C array
char* text = "hallo"; // C pointer
```

Schreibweise

```
int textLen = 12; // üblich für Var. / Fkt.
int TextLen = 13; // üblich für eigene Typen
int TEXT_LEN = 14; // Konstanten (veraltet)
int text_len = 15; // Standardbibliothek
```

- *Identifier* bestehen aus Buchstaben (A-Z bzw. a-z), Unterstrichen und Ziffern (außer als erstes Zeichen)
- Groß- / Kleinschreibung beachten
- keine Typinformation im Namen (iCount, dArea)
- Konsistenz ist erstrebenswert

- siehe auch *C++ Core Guidelines*

Ausgabe

```
#include <iostream>
#include <string>

int main()
{
   std::string text = "Ergebnis: ";
   int result = 2 + 3;

   std::cout << text << result << std::endl;

   return 0;
}
```

Zeilenumbruch
+ „flush“

Aufgaben

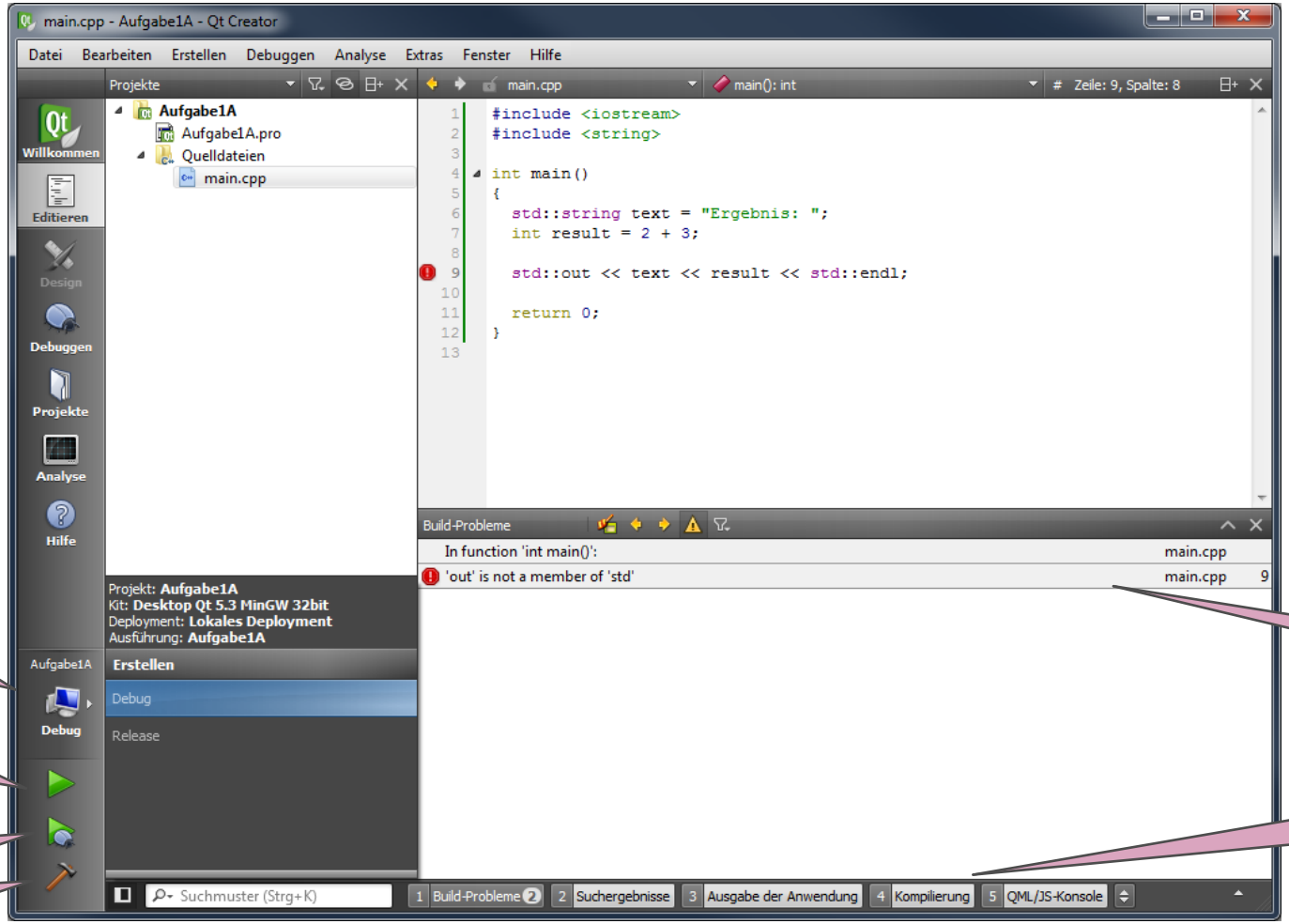
Aufgabe 1A

- anstatt einer Ganzzahl soll eine Gleitkommazahl berechnet und ausgegeben werden

Aufgabe 1B

- eine Zufallszahl soll ausgegeben werden
- ermittelt werden kann diese mit `rand()` (zunächst einmal nicht sonderlich zufällig)
- dazu braucht es außerdem `#include <cstdlib>`

Compiling


main.cpp - Aufgabe1A - Qt Creator
Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe
Projekte
Aufgabe1A
Aufgabe1A.pro
Quelldateien
main.cpp
main.cpp
main(): int
Zeile: 9, Spalte: 8
#include <iostream>
#include <string>
int main()
{
std::string text = "Ergebnis: ";
int result = 2 + 3;
std::out << text << result << std::endl;
return 0;
}
Willkommen
Editieren
Design
Debuggen
Projekte
Analyse
Hilfe
Projekt: Aufgabe1A
Kit: Desktop Qt 5.3 MinGW 32bit
Deployment: Lokales Deployment
Ausführung: Aufgabe1A
Aufgabe1A
Debug
Erstellen
Debug
Release
Build-Probleme
In function 'int main()': main.cpp
'out' is not a member of 'std' main.cpp 9
Suchmuster (Strg+K)
1 Build-Probleme 2
2 Suchergebnisse
3 Ausgabe der Anwendung
4 Kompilierung
5 QML/JS-Konsole
Konfiguration
Ausführen
Debuggen
Erstellen
Probleme
Compiler-Ausgaben

String-Literale

```
#include <string>

int main()
{
   std::string text = "eins\nzwei";
   std::string crlf = "\r\n";
   std::string path = "C:\\Windows\\";
   std::string quote = "\"super\"";
   std::string space = "\t\x20\0";
   std::string arrow = "\u2192";
   return 0;
}
```

nicht immer nötig

Exkurs: using

```
#include <iostream>

// using namespace std; // gar nicht gut!

using std::cout;
using std::endl;

int main()
{
  cout << "hallo Welt" << endl;
  return 0;
}
```

enthält z.B. count, get, set, copy, move, swap, begin, end, min, max, prev, next, find, search, distance...

Eingabe

```
#include <iostream>

using std::cin;
using std::cout;
using std::endl;

int main()
{
  double x;
  cin >> x;
  cout << x * x << endl;

  return 0;
}
```

Aufgaben

Aufgabe 2A

- es sollen zwei Zahlen eingegeben werden, die miteinander addiert werden

Aufgabe 2B

- vor der Ausgabe einer Zufallszahl soll der Anfangswert des Zufallszahlengenerators eingegeben werden
- initialisieren lässt sich der Zufallszahlengenerator mittels `srand(n)` (wobei normalerweise z.B. die aktuelle Uhrzeit verwendet wird)

Eingabe

```
#include <string>
#include <iostream>

using std::string;
using std::cin;
using std::cout;
using std::endl;

int main()
{
  string text;
  std::getline(cin, text);
  cout << text << endl;

  return 0;
}
```

Dateien

```
#include <fstream>

using std::ofstream;
using std::endl;

int main()
{
  ofstream stream("test.txt",
                  ofstream::out | ofstream::trunc);
  stream << "The quick brown fox is lazy today" << endl;

  return 0;
}
```

output file stream

mathematische Operatoren

+	Addition	a + b
–	Subtraktion	a – b
*	Multiplikation	a * b
/	Division	a / b
%	Modulo	a % b

```
double y = b + m * x;
int pos = (i + j) % 10;
std::cout << 100.0 * a / sum << std::endl;
```

Aufgaben

Aufgabe 3A

- zwei eingegebene Zahlen sollen dividiert werden und das Ergebnis inklusive Rest ausgegeben

Aufgabe 3B

- es soll eine Zufallszahl zwischen 1 und 6 „erwürfelt“ werden

Vergleichsoperatoren

==	gleich	`a == b`
!=	ungleich	`a != b`
<	kleiner als	`a < b`
>	größer als	`a > b`
<=	kleiner oder gleich	`a <= b`
>=	größer oder gleich	`a >= b`

Bedingungen

```
int main()
{
  int n;
  cin >> n;

  if (n < 0)
  {
    cout << "negativ" << endl;
  }

  return 0;
}
```

Bedingungen

```
int main()
{
  int n;
  cin >> n;
  bool negative = n < 0;

  if (negative)
  {
    cout << "negativ" << endl;
  }

  return 0;
}
```

true oder false
bzw. 0 oder 1

ungleich null

Bedingungen

```
int main()
{
  int n;
  cin >> n;

  if (n < 0)
  {
    cout << "negativ" << endl;
  }
  else
  {
    cout << "positiv?" << endl;
  }

  return 0;
}
```

Bedingungen

```
int main()
{
  int n;
  cin >> n;

  if (n < 0)
  {
    cout << "negativ" << endl;
  }
  else if (n > 0)
  {
    cout << "positiv" << endl;
  }

  return 0;
}
```

Aufgaben

Aufgabe 4A

- der Betrag einer eingegebenen Zahl soll ausgegeben werden

Aufgabe 4B

- das Maximum zweier eingegebener Zahlen soll ausgegeben werden
- optional: Minimum und Maximum dreier Zahlen soll ausgegeben werden

Bedingter Ausdruck
(ternärer Operator)

```
int main()
{
  int n;
  cin >> n;

  string text = n == 0 ? "null" : "nicht null";
  cout << text << endl;

  return 0;
}
```

```
int max = a > b ? a : b;
```

logische Operatoren

!	nicht	!(a > b)
&&	und	(a < b) && (a < c)
||	oder	(a >= b) || (a >= c)

```
int main()
{
  int n;
  cin >> n;

  if (n >= 1 && n <= 6)
  {
    cout << "Würfelst du?" << endl;
  }

  return 0;
}
```

Zuweisungsoperatoren

`=`	Zuweisung	`a = 0;`
`+=`	Inkrement	`a += 10;`
`-=`	Dekrement	`a -= 5;`
`*=`	Multiplikation	`a *= 2;`
`/=`	Division	`a /= 2;`
`++`	Inkrement	`++a; a++;`
`--`	Dekrement	`--a; a--;`

```
b = a = 5;
c = ++a;
d = a++;
```