

30.10.2019 17:24:05

Makefile

Page 1/1

```

1
2 # Makefile für Übung "Klasse Bruch"
3
4 BIN      = appl.exe
5 OBJS     = Bruch.o BruchAppl.o
6
7 CC       = g++
8 CFLAGS  = -g
9
10 all: $(BIN)
11
12
13 $(BIN): $(OBJS)
14 $(CC) $(CFLAGS) -o $(BIN) $(OBJS)
15
16 clean:
17 rm -f $(OBJS) $(BIN)
18
19 %.o: %.cpp %.h
20 $(CC) $(CFLAGS) -c $<
21
22 %.o: %.cpp
23 $(CC) $(CFLAGS) -c $<
24

```

30.10.2019 17:23:27

Bruch.h

Page 1/1

```

1 //=====
2 //      * Letsch Informatik *      www.LetsInfo.ch      CH-8636 Wald
3 //      Beratung, Ausbildung und Realisation in Software-Engineering
4 //=====
5 // Project   : Master of Advanced Studies in Software-Engineering MAS-SE 2019
6 // Title     : Klasse "Bruch"
7 // Author    : Thomas Letsch
8 // Tab-Width : 4
9 //=====
10 * Description: Header-File zu Beispiel-Klasse "Bruch" (Musterlösung).
11 * $Revision : 1.30 $ $Date: 2019/10/30 17:23:07 $
12 //=====
13 //      1      2      3      4      5      6      7      8
14 //3456789012345678901234567890123456789012345678901234567890
15 //=====
16
17 #include <iostream>
18
19 using std::cout;
20 using std::ostream;
21
22 class Bruch {
23
24     // Bruch ausgeben (via Friend):
25     friend ostream& operator<<(ostream&, const Bruch&);
26
27     public:
28         Bruch(); // Default-Konstruktor
29         Bruch(int pZahl); // Konstruktor mit ganzer Zahl
30         Bruch(int pZaehler, int pNenner); // Konstruktor: Zaehler, Nenner
31         void drucke(); // Bruch ausgeben (via Methode)
32         Bruch operator*(Bruch); // Operator Multiplikation
33         Bruch operator/(Bruch); // " Divison
34         Bruch operator+(Bruch); // " Addition
35         Bruch operator-(Bruch); // " Subtraktion
36         Bruch operator!(); // Operator Kehrwert
37         Bruch kuerze(); // Bruch kuerzen
38
39     protected:
40         int ggt(int i1, int i2); // Groesster gemeinsamer Teiler
41
42     private:
43         int mZaehler; // Zaehler als Ganzzahl
44         int mNenner; // Nenner als Ganzzahl
45
46 };

```

30.10.2019 17:23:53

Bruch.cpp

Page 1/2

```

1 //=====
2 //      * Letsch Informatik *      www.LetsInfo.ch      CH-8636 Wald
3 //      Beratung, Ausbildung und Realisation in Software-Engineering
4 //=====
5 // Project   : Master of Advanced Studies in Software-Engineering MAS-SE 2019
6 // Title     : Klasse "Bruch"
7 // Author    : Thomas Letsch
8 // Tab-Width : 4
9 //=====
10 * Description: Implementations-File zu Beispiel-Klasse "Bruch" (Musterlösung).
11 * $Revision : 1.30 $   $Date: 2019/10/30 17:23:07 $
12 //=====
13 //      1          2          3          4          5          6          7          8
14 //34567890123456789012345678901234567890123456789012345678901234567890
15 //=====
16
17 #include "Bruch.h"
18
19 Bruch::Bruch() : mZaehler(0), mNenner(1) {
20     ;
21 }
22
23 Bruch::Bruch(int pZahl) : mZaehler(pZahl), mNenner(1) {
24     ;
25 }
26
27
28 Bruch::Bruch(int pZaehler, int pNenner) : mZaehler(pZaehler), mNenner(pNenner) {
29     ;
30 }
31
32
33 void Bruch::drucke() {
34     cout << "Bruch (" << mZaehler << "/" << mNenner << ")" << "\n";
35 }
36
37
38 Bruch Bruch::operator*(Bruch b) {
39     return Bruch(mZaehler * b.mZaehler, mNenner * b.mNenner);
40
41     // Wir sehen spaeter das dieser Operator besser mit einem Friend-Operator
42     // realisiert wuerde (wie nachfolgender operator<<).
43 }
44
45
46 Bruch Bruch::operator/(Bruch b) {
47     return Bruch(mZaehler * b.mNenner, mNenner * b.mZaehler);
48 }
49
50
51 Bruch Bruch::operator+(Bruch b) {
52     return Bruch(mZaehler * b.mNenner + mNenner * b.mZaehler, mNenner * b.mNenner);
53 }
54
55
56
57

```

30.10.2019 17:23:53

Bruch.cpp

Page 2/2

```

57 Bruch Bruch::operator-(Bruch b) {
58     return Bruch(mZaehler * b.mNenner - mNenner * b.mZaehler, mNenner * b.mNenner);
59 }
60
61
62 Bruch Bruch::operator!() {
63     return Bruch(mNenner, mZaehler);
64 }
65
66
67 Bruch Bruch::kuerze() {
68     int theGGT = ggt(mZaehler, mNenner);
69     mZaehler = mZaehler / theGGT;
70     mNenner = mNenner / theGGT;
71     return *this;
72 }
73
74
75 int Bruch::ggt(int i1, int i2) {
76     if (i2 == 0)
77         return i1;
78     return ggt(i2, i1 % i2);
79 }
80
81
82 ostream& operator<<(ostream& os, const Bruch& b) {
83     os << "Bruch (" << b.mZaehler << "/" << b.mNenner << ")" << "\n";
84     return os;
85 }

```

30.10.2019 17:22:45

BruchAppl.cpp

Page 1/2

```

1 //=====
2 //      * Letsch Informatik *      www.LetsInfo.ch      CH-8636 Wald
3 //      Beratung, Ausbildung und Realisation in Software-Engineering
4 //=====
5 // Project   : Master of Advanced Studies in Software-Engineering MAS-SE 2019
6 // Title     : Klasse "Bruch"
7 // Author    : Thomas Letsch
8 // Tab-Width : 4
9 //=====
10 * Description: Test-Applikation zu Beispiel-Klasse "Bruch" (Musterlösung).
11 * $Revision : 1.30 $   $Date: 2019/10/30 17:22:07 $
12 //=====
13 //      1      2      3      4      5      6      7      8
14 //3456789012345678901234567890123456789012345678901234567890
15 //=====
16
17 #include <iostream>
18
19 using std::cout;
20
21 #include "Bruch.h"
22
23
24 int main() {
25
26     Bruch bruch1;
27     Bruch bruch2 = Bruch(1, 2); // Definition und Initialisierung
28     Bruch bruch3(2, 5);        // andere Variante der Initialisierung
29
30     bruch1.drucke(); // Ausgabe via Funktion
31     cout << "bruch1 = " << bruch1; // Ausgabe via Operator
32     cout << "bruch2 = " << bruch2;
33     cout << "bruch3 = " << bruch3;
34
35     // verschiedene Beispiele mit dem Multiplikator-Operator
36     cout << "bruch2 * bruch3" = " << bruch2 * bruch3;
37     cout << "bruch1 * bruch2 * bruch3" = " << bruch1 * bruch2 * bruch3;
38     cout << "bruch2 * 4" = " << bruch2 * 4;
39
40     cout << "bruch2 / bruch3" = " << bruch2 / bruch3;
41     cout << "Bruch(3, 4) / Bruch(2, 3)" = " << Bruch(3, 4) / Bruch(2, 3);
42     cout << "bruch2 + bruch3" = " << bruch2 + bruch3;
43     cout << "bruch2 - bruch3" = " << bruch2 - bruch3;
44     cout << "!bruch3" = " << !bruch3;
45     cout << "Bruch(12, 27).kuerze()" = " << Bruch(12, 27).kuerze();
46
47     return 0;
48
49 }
50
51
52

```

30.10.2019 17:22:45

BruchAppl.cpp

Page 2/2

```

52 /* Session-Log:
53 $
54 $ g++ -o appl.exe Bruch.cpp BruchAppl.cpp
55 $
56 $ appl.exe
57 Bruch (0/1)
58 bruch1 = Bruch (0/1)
59 bruch2 = Bruch (1/2)
60 bruch3 = Bruch (2/5)
61 bruch2 * bruch3 = Bruch (2/10)
62 bruch1 * bruch2 * bruch3 = Bruch (0/10)
63 bruch2 * 4 = Bruch (4/2)
64 bruch2 / bruch3 = Bruch (5/4)
65 Bruch(3, 4) / Bruch(2, 3) = Bruch (9/8)
66 bruch2 + bruch3 = Bruch (9/10)
67 bruch2 - bruch3 = Bruch (1/10)
68 !bruch3 = Bruch (5/2)
69 Bruch(12, 27).kuerze() = Bruch (4/9)
70 $
71
72 */

```