

Aufgabe	A1	A2	A3	Σ
Punkte				

Aufgabe 1. Lesen und Schreiben von Dateien

```

1 #include <fstream>
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 int main(int argc, char* argv[]) {
8     // check if correct argument count
9     if (argc < 2) {
10         printf("Dateiname erwartet\n");
11         return 1;
12     }
13
14     // get filename from argument list
15     string filename = argv[1];
16     // open input file
17     ifstream fileIn;
18     fileIn.open(filename + ".txt");
19     // open output file
20     ofstream fileOut;
21     fileOut.open(filename + "-a.txt");
22     // check if we could open the files
23     if (!fileIn.good() || !fileOut.good()) {
24         // could not open files
25         cout << "Konnte Datei " + filename + ".txt nicht lesen" << endl;
26         return 1;
27     }
28     string input; // variable containing line
29     int lineNo = 1; // count lines
30     while(getline(fileIn, input)) { // while there is still a line left
31         // write out line with prepended line number to output file
32         fileOut << to_string(lineNo) + ": " + input << endl;
33         // increment line counter
34         lineNo += 1;
35     }
36     // close files
37     fileIn.close();
38     fileOut.close();
39 }

```

readfiles.cpp

Aufgabe 2. Fehler sind im Quellcode markiert

```

1 #include <iostream>
2 class A
3 {
4     public:
5         int ap;
6         void X();
7     private:
8         int aq;
9         void aX();
10 };
11
12 class B : public A
13 {
14     public:
15         int bp;
16         void Y();
17     private:
18         int bq;
19         void bY();
20 };
21
22 class C : public B
23 {
24     public:
25         int cp;

```

```

26     void Z();
27     private:
28         int cq;
29         void cZ();
30 };
31
32 void A::X() { };
33 void A::aX() { };
34 void B::bY() { };
35 void C::Z() { };
36
37 void B::Y()
38 {
39     bq = bp;
40     aq = ap; // Fehler: A::aq ist privat, also in B nicht sichtbar
41     bY();
42 }
43
44 void C::cZ()
45 {
46     ap = 1;
47     bp = 2;
48     cq = 3;
49     X();
50     Y();
51     aX(); // Fehler: A::aX() ist privat, also in C nicht sichtbar
52 }
53
54 int main()
55 {
56     A a; B b; C c;
57
58     a.X();
59     b.bY(); // Fehler: B::bY() ist privat, also hier nicht sichtbar
60     c.cp = 4;
61     c.bp = 1;
62     c.ap = 2;
63     c.aq = 5; // Fehler: A::aq ist privat, in C nicht sichtbar und erst recht nicht
64     hier
65
66     b.ap = c.ap;
67
68     return 0;
69 }

```

Fehlersuche

Aufgabe 3. Fehler wurden weggelassen mit Begründung dahinter, hinter den korrekten Aufrufen steht jeweils, welche Implementation aufgerufen wird.

```

1 // A a; A abstrakte Klasse
2 B b;
3 C c;
4 A* pa=&b;
5 B* pb=&c;
6 float x = 1.2;
7 pa->a(); // A::a()
8 pa->va(); // B::va()
9 pa->a(1); // B::a(int)
10 // pa->c(); A::c() private, hier nicht sichtbar
11 // pa->b(); pa ist vom Typ A* und kennt deswegen b() nicht
12 // pa->vb(); pa ist vom Typ A* und kennt deswegen vb() nicht
13 pa->a(x); // B::a(int), hier wird double x implizit zu int gecastet
14 // pb->a(); pb ist vom Typ B*, hier ist A::a() ueberladen durch
15 // entweder B::a(double) bzw. B::a(int), deswegen existiert B::a() nicht
16 pb->va(); // C::va()
17 pb->a(1); // B::a(int)
18 // pb->c(); A::c() private, deswegen hier nicht sichtbar
19 pb->b(); // B::b()
20 pb->vb(); // B::vb()
21 pb->a(x); // B::a(double)
22 pa = &c;
23 pa->a(); // A::a()
24 pa->va(); // C::va()
25 pa->a(1); // B::a(int)
26 // pa->c(); A::c() private, hier nicht sichtbar

```

```
27 // pa->b(); pa ist vom Typ A*, kennt deswegen b() nicht
28 // pa->vb(); pa ist vom Typ A*, kennt deswegen vb() nicht
29 pa->a(x); // B::a(int), hier wird double x implizit zu int gecastet
```

Virtuelle Methoden