Objektorientiertes Programmieren - SWB2 & TIB2
Labor 4

Aufgabe 1: Medien in einer Bibliothek

In einer Bibliothek können sich Personen Medien ausleihen, z.B. Bücher oder DVDs. An der Hochschule können die Personen Studierende und Dozenten sein.

Programmieren Sie daher die folgenden Klassen aus, so dass das nachfolgende Hauptprogramm ausgeführt werden kann. In den gezeigten Dateien fehlt jedoch noch die Vererbungshierarchie. Daher müssen Sie vier der hpp-Dateien noch leicht erweitern.

```
// Bibliothek.hpp
#pragma once

#include "Buch.hpp"
#include "DVD.hpp"
#include "Person.hpp"
#include "Datum.hpp"

// Die Klasse Bibliothek verweist auf alle Medien (Bücher und DVDs),
// die von Personen (Studierende und Dozenten) ausgeliehen werden
// können.
class Bibliothek {
    // Anzahl der in der im Bibliothekskatalog verzeichneten Medien
    int anz;
    // maximale Anzahl der Medien im Katalog
    const int maxAnz;
    // Zeiger auf das Array der Zeiger auf die Medien im Katalog
    // Deshalb wird hier ein Doppelzeiger genutzt.
    Medium ** medien;
public:
    // Standardkonstruktor
    Bibliothek(int maxAnz = 1000);
    // Destruktor
    ~Bibliothek();
    // Kopie eines Buches in den Katalog der Bibliothek eintragen
    void mediumBeschaffen(Buch &);
    // Kopie einer DVD in den Katalog der Bibliothek eintragen
    void mediumBeschaffen(DVD &);
    // alle Medien auf der Konsole ausgeben,
    // die im Titel das Suchwort enthalten
    void mediumSuchen(string suchwort);
    // im Medium mit der Nummer nr, die Person p als Ausleiher
    // eintragen mit von-Datum d und bis-Datum d+p.ausleihdauer
    void mediumAusleihen(int nr, Person & p, Datum d);
    // alle Medien in der Konsole ausgeben
    void print() const;
```

```
};
```

```
// Medium.hpp
#pragma once

#include <string>
#include "Datum.hpp"
#include "Person.hpp"
using namespace std;

// Klasse für die Medien, die in der Bibliothek ausgeliehen werden
// können
class Medium {
    // Titel des Mediums
    const string titel;
    // Verlag, der das Medium herausgibt
    const string verlag;
    // Jahr, in dem das Medium veröffentlicht wurde
    const int jahr;
    // Typ des Mediums (z.B. Buch, DVD, ...)
    // wird von den abgeleiteten Klassen festgelegt
    const string typ;
    // ausgeliehen von
    Person * ausleiher;
    // ausgeliehen am
    Datum von;
    // ausgeliehen bis
    Datum bis;

public:
    // Konstruktor
    Medium(string t = "", string v = "", int j = 0,
           string typ = "undef");
    // Titel zurückliefern
    string getTitel() const;
    // Typ zurückliefern
    string getTyp() const;
    // Ausleiher zurückliefern
    Person * getAusleiher() const;
    // das Mediuem "ausleihen", d.h. Person p, von und bis eintragen
    void ausleihen(Person & p, Datum von, Datum bis);
    // Medium in der Konsole ausgeben
    void print() const;

};
```

```
// Buch.hpp
#pragma once
```

```
#include "Medium.hpp"

// Klasse für die Bücher als Spezialisierung von Medium
class Buch {
    // Autor(en) des Buches
    string autor;
public:
    // Standardkonstruktor
    Buch(string t = "", string a = "", string v = "", int jahr = 0);
    // das Buch auf der Konsole ausgeben
    void print() const;
};
```

```
// DVD.hpp
#pragma once

#include "Medium.hpp"

// Klasse für die DVDs als Spezialisierung von Medium
class DVD {
    // Abspieldauer der DVD
    const int dauer;
public:
    // Standardkonstruktor
    DVD(string t = "", string v = "", int j = 0, int d = 0);
    // die DVD auf der Konsole ausgeben
    void print() const;
};
```

```
// Person.hpp
#pragma once

#include <string>
using namespace std;

// Klasse für alle Personen, die Medien ausleihen können
class Person {
    // Name der Person
    string name;
    // Dauer in Tagen, die die Person ein Medium ausleihen darf
    // wird von den abgelieteten Klassen festgelegt
    int ausleihdauer;
public:
    // Standardkonstruktor
    Person(string name, int dauer = 0);
    // den Namen zurückliefern
    string getName() const;
```

```
    // die Ausliehdauer zurückliefern
    int getAusleihdauer() const;
    // die Person auf der Konsole ausgeben
    void print() const;
};
```

```
// Student.hpp
#pragma once

#include "Person.hpp"

// Klasse Student als Spezialisierung von Person
class Student {
    // Matrikelnummer des Studenten/der Studentin
    int matNr;
public:
    // Standardkonstruktor
    Student(string name, int matNr);
    // Student auf der Konsole ausgeben
    void print() const;
};
```

```
// Dozent.hpp
#pragma once

#include "Person.hpp"

// Klasse Dozent als Spezialisierung von Person
class Dozent {
    // Prüfernummer des Dozenten
    int prfrNr;
public:
    // Standardkosntruktor
    Dozent(string name, int prfrNr);
    // Dozenten auf der Konsole ausgeben
    void print() const;
};
```

```
// Datum.hpp
#pragma once

#include <string>
using namespace std;

// Klasse Datum, ähnlich zu der in der Übung der Vorlesung
class Datum {
    // Elemente eines Datums
    int tag, monat, jahr;
```

```
public:
    // Standardkonstruktor
    Datum(int = 0, int = 0, int = 0);
    // Konvertierkonstruktor für String
    Datum(const string &);
    // Konvertierkonstruktor für C-String
    Datum(const char *);
    // Operator + addiert eine Anzahl von Tagen zum Datum hinzu
    Datum operator+(int tage);
    // Ausgabeoperator <<, Ausgabe in dem Format tt.mm.jjjj
    friend ostream & operator<<(ostream &, const Datum &);
};
```

Hauptprogramm zu Testen:

```
#include <clocale>
#include "Bibliothek.hpp"
#include "Buch.hpp"
#include "DVD.hpp"
#include "Student.hpp"
#include "Dozent.hpp"

int main() {
    // Ulaute etc. in der Konsole zulassen
    setlocale(LC_ALL, "");
    // Bibliothek mit 100 Plätzen initialisieren
    Bibliothek bib(100);
    // Bücher und DVDs erstellen
    Buch b1("C++: das umfassende Handbuch", "Jürgen Wolf",
            "Galileo Press", 2014);
    Buch b2("C++ - der Einstieg", "Arnold Willemer", "Wiley", 2014)
       ;
    Buch b3("Der C++-Programmierer", "Rainer Grimm", "O'Reilly",
            2014);
    Buch b4("C++ for Dummies", "Stephen R. Davies", "Wiley", 2012);
    Buch b5("C++ lernen und professionell anwenden",
            "Ulla Kirch und Peter Prinz", "mitp", 2012);
    Buch b6("BeagleBone für Einsteiger", "Matt Richardson",
            "O'Reilly", 2014);
    DVD d1("Die Rächer von C++", "DVD ex", 1984, 666);
    DVD d2("Ganz nah dabei - Raumgestaltung in Kitas \
für 0- bis 3-Jährige", "Cornelsen", 2013, 30);
    // Ein Buch und eine DVD ausgeben
    b1.print();
    d1.print();
    // Kopien der Bücher und DVDs (Medien) in die Bibliothek
    // einfügen
```

```
    bib.mediumBeschaffen(b1);
    bib.mediumBeschaffen(b2);
    bib.mediumBeschaffen(b3);
    bib.mediumBeschaffen(b4);
    bib.mediumBeschaffen(b5);
    bib.mediumBeschaffen(b6);
    bib.mediumBeschaffen(d1);
    bib.mediumBeschaffen(d2);
    // Bestand der Bibliothek ausgeben
    bib.print();
    // Personen anlegen
    Student p1("Hägar", 12345678);
    Student p2("Hilde", 87654321);
    Dozent p3("Prof A", 4711);
    // Suchen im Bibliotheksbestand durchführen
    bib.mediumSuchen("C++");
    bib.mediumSuchen("Kita");
    // Medien ausleihen
    bib.mediumAusleihen(2, p1, "25.04.2014");
    bib.mediumAusleihen(7, p2, "26.04.2014");
    bib.mediumAusleihen(6, p3, "21.04.2014");
    // Bestand der Bibliothek ausgeben
    bib.print();
}
```

Die Ausgabe der Hauptfunktion kann so aussehen:

```
----------------------------------------
Buch
Titel:     C++: das umfassende Handbuch
Verlag:    Galileo Press
Jahr:      2014
Ausleiher: kein
Autor:     Jürgen Wolf
----------------------------------------
DVD
Titel:     Die Rächer von C++
Verlag:    DVD ex
Jahr:      1984
Ausleiher: kein
Dauer:     666

Bibliothekskatalog:
----------------------------------------
Buch
Titel:     C++: das umfassende Handbuch
Verlag:    Galileo Press
```

```
Jahr:      2014
Ausleiher: kein
Autor:     Jürgen Wolf
----------------------------------------
Buch
Titel:     C++ - der Einstieg
Verlag:    Wiley
Jahr:      2014
Ausleiher: kein
Autor:     Arnold Willemer
----------------------------------------
Buch
Titel:     Der C++-Programmierer
Verlag:    O'Reilly
Jahr:      2014
Ausleiher: kein
Autor:     Rainer Grimm
----------------------------------------
Buch
Titel:     C++ for Dummies
Verlag:    Wiley
Jahr:      2012
Ausleiher: kein
Autor:     Stephen R. Davies
----------------------------------------
Buch
Titel:     C++ lernen und professionell anwenden
Verlag:    mitp
Jahr:      2012
Ausleiher: kein
Autor:     Ulla Kirch und Peter Prinz
----------------------------------------
Buch
Titel:     BeagleBone für Einsteiger
Verlag:    O'Reilly
Jahr:      2014
Ausleiher: kein
Autor:     Matt Richardson
----------------------------------------
DVD
Titel:     Die Rächer von C++
Verlag:    DVD ex
Jahr:      1984
Ausleiher: kein
Dauer:     666
----------------------------------------
DVD
Titel:     Ganz nah dabei - Raumgestaltung in Kitas für 0- bis 3-Jährige
```

```
Verlag:    Cornelsen
Jahr:      2013
Ausleiher: kein
Dauer:     30

Suche nach "C++". Ergebnis:

Medium 0:
----------------------------------------
Buch
Titel:     C++: das umfassende Handbuch
Verlag:    Galileo Press
Jahr:      2014
Ausleiher: kein

Medium 1:
----------------------------------------
Buch
Titel:     C++ - der Einstieg
Verlag:    Wiley
Jahr:      2014
Ausleiher: kein

Medium 2:
----------------------------------------
Buch
Titel:     Der C++-Programmierer
Verlag:    O'Reilly
Jahr:      2014
Ausleiher: kein

Medium 3:
----------------------------------------
Buch
Titel:     C++ for Dummies
Verlag:    Wiley
Jahr:      2012
Ausleiher: kein

Medium 4:
----------------------------------------
Buch
Titel:     C++ lernen und professionell anwenden
Verlag:    mitp
Jahr:      2012
Ausleiher: kein

Medium 6:
```

```
----------------------------------------
DVD
Titel:     Die Rächer von C++
Verlag:    DVD ex
Jahr:      1984
Ausleiher: kein

Suche nach "Kita". Ergebnis:

Medium 7:
----------------------------------------
DVD
Titel:     Ganz nah dabei - Raumgestaltung in Kitas für 0- bis 3-Jährige
Verlag:    Cornelsen
Jahr:      2013
Ausleiher: kein

Bibliothekskatalog:
----------------------------------------
Buch
Titel:     C++: das umfassende Handbuch
Verlag:    Galileo Press
Jahr:      2014
Ausleiher: kein
Autor:     Jürgen Wolf
----------------------------------------
Buch
Titel:     C++ - der Einstieg
Verlag:    Wiley
Jahr:      2014
Ausleiher: kein
Autor:     Arnold Willemer
----------------------------------------
Buch
Titel:     Der C++-Programmierer
Verlag:    O'Reilly
Jahr:      2014
Ausleiher: Hägar von: 25.04.2014 bis: 25.05.2014
Autor:     Rainer Grimm
----------------------------------------
Buch
Titel:     C++ for Dummies
Verlag:    Wiley
Jahr:      2012
Ausleiher: kein
Autor:     Stephen R. Davies
----------------------------------------
Buch
```

```
Titel:     C++ lernen und professionell anwenden
Verlag:    mitp
Jahr:      2012
Ausleiher: kein
Autor:     Ulla Kirch und Peter Prinz
----------------------------------------
Buch
Titel:     BeagleBone für Einsteiger
Verlag:    O'Reilly
Jahr:      2014
Ausleiher: kein
Autor:     Matt Richardson
----------------------------------------
DVD
Titel:     Die Rächer von C++
Verlag:    DVD ex
Jahr:      1984
Ausleiher: Prof A von: 21.04.2014 bis: 21.07.2014
Dauer:     666
----------------------------------------
DVD
Titel:     Ganz nah dabei - Raumgestaltung in Kitas für 0- bis 3-Jährige
Verlag:    Cornelsen
Jahr:      2013
Ausleiher: Hilde von: 26.04.2014 bis: 26.05.2014
Dauer:     30
```

Aufgabe 2: Funktionen überladen, überdecken und redefinieren

Schauen Sie sich folgendes Programm an.

a) Sagen Sie vorher, in welchen Zeilen ein Kompiler eine Fehlermeldung meldet.
b) Sagen Sie die Ausgabe des Hauptprogramms ohne die fehlerhaften Zeilen voraus.
c) Dokumentieren Sie, welche Funktion welche andere Funktion überlädt, verdeckt oder redefiniert.
d) Kommentieren Sie die fehlerverursachenden Zeilen aus und überprüfen Sie Ihre Vorhersagen mit einem Kompiler.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class A {
6  public:
7      void f() {
8          cout << "A::f()" << endl;
9      }
10     void g(double d) {
11         cout << "A::g(double)" << endl;
```

```
12      }
13      void g(string s) {
14          cout << "A::g(string)" << endl;
15      }
16      void h(char c) {
17          cout << "A::h(char)" << endl;
18      }
19      void h(string s) {
20          cout << "A::h(string)" << endl;
21      }
22  };
23
24  class B : public A {
25  public:
26      void g(int i) {
27          cout << "B::g(int)" << endl;
28      }
29      void h(int i) {
30          cout << "B::h(int)" << endl;
31      }
32      void h(string s) {
33          cout << "B::h(string)" << endl;
34      }
35  };
36
37  int main() {
38      A a;
39      B b;
40      cout << "a.f() ";
41      a.f();    // Überlädt ... Überdeckt ... Redefiniert ...
42      cout << "b.f() ";
43      b.f();    // Überlädt ... Überdeckt ... Redefiniert ...
44      cout << "a.g(1.2) ";
45      a.g(1.2); // Überlädt ... Überdeckt ... Redefiniert ...
46      cout << "b.g(1.2) ";
47      b.g(1.2); // Überlädt ... Überdeckt ... Redefiniert ...
48      cout << "a.g(1) ";
49      a.g(1);   // Überlädt ... Überdeckt ... Redefiniert ...
50      cout << "b.g(1) ";
51      b.g(1);   // Überlädt ... Überdeckt ... Redefiniert ...
52      cout << "a.g('a') ";
53      a.g('a'); // Überlädt ... Überdeckt ... Redefiniert ...
54      cout << "b.g('a') ";
55      b.g('a'); // Überlädt ... Überdeckt ... Redefiniert ...
56      cout << "a.g(\"a\") ";
57      a.g("a"); // Überlädt ... Überdeckt ... Redefiniert ...
58      cout << "b.g(\"a\") ";
59      b.g("a"); // Überlädt ... Überdeckt ... Redefiniert ...
```

```
60      cout << "a.h(1) ";
61      a.h(1);   // Überlädt ... Überdeckt ... Redefiniert ...
62      cout << "b.h(1) ";
63      b.h(1);   // Überlädt ... Überdeckt ... Redefiniert ...
64      cout << "a.h('a') ";
65      a.h('a'); // Überlädt ... Überdeckt ... Redefiniert ...
66      cout << "b.h('a') ";
67      b.h('a'); // Überlädt ... Überdeckt ... Redefiniert ...
68      cout << "a.h(\"a\") ";
69      a.h("a"); // Überlädt ... Überdeckt ... Redefiniert ...
70      cout << "b.h(\"a\") ";
71      b.h("a"); // Überlädt ... Überdeckt ... Redefiniert ...
72      return 0;
73  }
```

Aufgabe 3: Mehrfache Vererbung und Konstruktoren

Das am Ende der Aufgabe gegebene Programm erzeugt die folgende Ausgabe:

```
D1::print() ...
--------------------
C1::a = D1C1A
--------------------
C2::a = D1C2A
--------------------
C1::b = D1C1B
--------------------
C2::b = D1C2B
--------------------

D2::print() ...
--------------------
C1::a = D2C1A
--------------------
C2::a = D2C2A
--------------------
C3::a = D2C3A
--------------------
C1::b = D2C1B
--------------------
C2::b = D2C2B
--------------------
C3::b = D2C3B
--------------------
```

Im Programm wurde **`struct`** statt **`class`** lediglich genutzt, um das Programm kürzer zu gestalten. Die Funktionen `suche` und `speicherzelle` zusammen geben die Instanzvariablen der Objekte `d1obj` und `d2obj` aus. Instanzvariablen, die bei der Ausgabe in getrennten Zeilen

ausgegeben werden, sind eigenständige Instanzvariablen, die eigenen Speicherplatz in Anspruch nehmen. In der Ausgabe oben werden daher die Instanzvariablen `a` und `b` mehrfach an `D1` und `D2` vererbt.

Ändern Sie die Klassen im Programm so ab, dass die folgende Ausgabe erzeugt wird, d.h. dass `a` nicht mehrfach und `b` nur teilweise mehrfach vererbt wird. Beachten Sie auch die Werte der Variablen.

```
D1::print() ...
--------------------
C1::a C2::a = D1A
--------------------
C1::b C2::b = B
--------------------

D2::print() ...
--------------------
C1::a C2::a C3::a = A
--------------------
C1::b C2::b = D2B
--------------------
C3::b = D2C3B
--------------------
```

```
#include <iostream>
#include <string>
using namespace std;

// Name s und Wert val einer Instanzvariable ausgeben
void speicherzelle(string s, int val) {
    cout << "--------------------" << endl;
    cout << s << "= " << val << endl;
}

// Die Instanzvariablen raussuchen und ausgeben,
// die wegen virtuellen Basisklassen zusammenfallen
void suche(string s[], int * vptr[], int n) {
    // Zustand von cout zwischenspeichern
    ios::fmtflags cflags = cout.flags();
    // Hilfsvariablen initialisieren
    int i = 0;
    int k = 0;
    string str = "";
    // cout für Ausgabe der Instanzvariablen konfigurieren
    cout << hex << uppercase;
    // Schauen, ob Instanzvariable i (vptr[i])
    // evtl. mit Variable k (vptr[k]) zusammenfällt
    while (i < n) {
```

```
        // Wenn Variable i+k und i+k+1 identisch ...
        if ((i+k < n-1) && (vptr[i+k] == vptr[i+k+1])) {
            // erhöhe k
            k++;
        } else {
        // andernfalls gib die zusammengefallenen Variablen
        // auf einer Zeile aus
            // Text für Ausgabe zusammenstellen
            for (int m = i; m<=i+k; m++) {
                str += s[m]+" ";
            }
            // Text und Wert ausgeben
            speicherzelle(str, *vptr[i]);
            // Hilfsvariablen für nächste Prüfung zurücksetzen
            str = "";
            i += k+1;
            k = 0;
        }
    }
    cout << "---------------------" << endl << endl;
    // Ursprünglicher Zustand von cout wieder herstellen
    cout.flags(cflags);
}

// Klassenhierarchie
class A {
public:
    int a;
    A(int a = 0xA) : a(a) {}
};

class B : public A {
public:
    int b;
    B(int a = 0xBA, int b = 0xB) : A(a), b(b) {}
};

class C1 : public B {
public:
    C1(int a = 0xC1A, int b = 0xC1B)
        : B(a, b) {}
};

class C2 : public B {
public:
    C2(int a = 0xC2A, int b = 0xC2B)
        : B(a, b) {}
};
```

```
class C3 : public B {
public:
    C3(int a = 0xC2A, int b = 0xC2B)
        : B(a, b) {}
};

class D1 : public C1, public C2 {
public:
    D1(int C1_a = 0xD1C1A, int C2_a = 0xD1C2A,
       int C1_b = 0xD1C1B, int C2_b = 0xD1C2B)
        : C1(C1_a, C1_b), C2(C2_a, C2_b) {}
    void print() {
        cout << "D1::print() ..." << endl;
        string strArr[] =
            { "C1::a", "C2::a", "C1::b", "C2::b" };
        int * intPtrArr[] = { &(C1::a), &(C2::a),
                              &(C1::b), &(C2::b) };
        suche(strArr, intPtrArr, 4);
    }
};

class D2 : public C1, public C2, public C3 {
public:
    D2(int C1_a = 0xD2C1A, int C2_a = 0xD2C2A, int C3_a = 0xD2C3A,
       int C1_b = 0xD2C1B, int C2_b = 0xD2C2B, int C3_b = 0xD2C3B)
        : C1(C1_a, C1_b), C2(C2_a, C2_b), C3(C3_a, C3_b) {}
    void print() {
        cout << "D2::print() ..." << endl;
        string strArr[] = { "C1::a", "C2::a", "C3::a",
                            "C1::b", "C2::b", "C3::b" };
        int * intPtrArr[] = { &(C1::a), &(C2::a), &(C3::a),
                              &(C1::b), &(C2::b), &(C3::b) };
        suche(strArr, intPtrArr, 6);
    }
};

int main() {
    D1 d1obj;
    D2 d2obj;
    d1obj.print();
    d2obj.print();
    return 0;
}
```

Aufgabe 4: Grafische Objekte - Teil 4 - Klassenhierarchie

Diese Aufgabe des Labors wird am sechsten Labortermin als Teil einer Gesamtaufgabe testiert.

Sie müssen in den folgenden Laboren die Lösung von diesem Labor erweitern.

Ergänzen Sie das Programm zu Punkten, Kreisen und Linienzügen aus Labor 3 folgendermaßen:

a) Schreiben Sie eine Klasse `DrawingObject` für Zeichenobjekte und leiten Sie sie von der Klasse `ObjectCounter` aus Hausaufgabe 3 ab.
b) Definieren Sie eine Klasse `OneDimObject` als abgeleitete Klasse der Klasse `DrawingObject`.
c) Machen Sie die Klasse `Point` zu einer abgeleiteten Klasse von `DrawingObject` und machen Sie die Klassen `Circle` und `Polygonline` zu abgeleiteten Klassen der Klasse `OneDimObject`.
d) Führen Sie für jede Klasse Konstruktoren und Destruktoren ein, so dass mit einer globalen Variablen `debugConstructor` die Textausgabe ein- und ausgeschaltet werden kann. Die Textausgabe soll nun auch die Objekt-ID mit ausgeben:

   `Konstruktor der Klasse <Klassenname>, Objekt: <id>`

   bzw.

   `Destruktor der Klasse <Klassenname>, Objekt: <id>`

e) Passen Sie Ihren Code aus dem vorherigen Labor an, wo notwendig.

Testen Sie Ihr Programm mit dem folgenden Hauptprogramm und verfolgen Sie die Aufrufe von Konstruktoren und Destruktoren. Wenn Sie dabei beobachten, dass Objekte scheinbar vorzeitig oder mehrmals zerstört werden, führen Sie einen Kopierkonstruktor mit passendem Ausgabetext ein. Wenn Ihr Programm erst gar nicht kompiliert werden kann, liegt das eventuell an einem fehlenden Zuweisungsoperator. Da die `id` eines Objektes vererbt von `ObjectCounter` konstant ist, kann der Standard-Zuweisungsoperator nicht mehr angewandt werden.

```
#include <iostream>
#include "Circle.hpp"
#include "Polygonline.hpp"
using namespace std;

bool debugConstructor = true;

unsigned int ObjectCounter::maxId = 0;
unsigned int ObjectCounter::number = 0;

// Hauptprogramm
int main(void)
{
    cout << "Anzahl der Objekte: " << DrawingObject::getNumber();
    cout << endl;
    Point p1;
    cout << "p1 ObjectId: " << p1.getId() << " " << p1 << endl;
    Point p2(1,1);
    cout << "p2 ObjectId: " << p2.getId() << " " << p2 << endl;
    Circle c1(p1,3);
    cout << "c1 ObjectId: " << c1.getId() << " " << c1 << endl;
```

```
    Polygonline l1(p1);
    cout << "l1 ObjectId: " << l1.getId() << " " << l1 << endl;
    cout << "Anzahl der Objekte: " << c1.getNumber() << endl;
}
```