

Polymorphismus II

AUFGABE 6.1 (Buchhändler) - Abgabe

Ein Buchhändler möchte seine Buchdatenbank mit einem Programm (von Ihnen in C++ geschrieben) verwalten.

- (a) (Niveau Objektorientierung) Erstellen Sie eine Datei Buchhaendler.cpp, die eine Klasse `Buch` mit folgenden Attributen enthält: Titel (String), Autor (String), Anzahl der Seiten (int) sowie einem Attribut, das angibt, ob das Buch ein Bestseller ist oder nicht.

Rüsten Sie Ihre Klasse mit folgendem aus:

- einer Methode `berechne_preis`, mit der der Preis eines Buches berechnet werden kann. Dieser Preis errechnet sich aus der Seitenanzahl multipliziert mit 0.3, zuzüglich 50 Euro, wenn es sich um einen Bestseller handelt;
- einen Konstruktor, der Parameter als Argumente verwendet, um alle Attribute der Klasse zu initialisieren;
- einem leeren virtuellen Destruktor;
- einer Anzeigemethode, die den Wert der Attribute eines Buches anzeigt. Der Preis des Buches wird bei dieser Methode nicht angezeigt. Lassen Sie sich bei der Gestaltung Ihrer Ausgabe von dem unten aufgeführten Ausführungsbeispiel inspirieren.

Die Attribute Ihrer Klasse sind privat. Testen Sie Ihre Klasse mit einer kleinen `main`, um etwa Folgendes zu erstellen:

Ausführungsbeispiel:

```
Titel: Harry Potter und der Stein der Weisen
Autor: J.K. Rowling
Anzahl Seiten: 308
Bestseller: ja
Preis: 82.4
```

```
Titel: Aufzeichnungen eines Wahnsinnigen
Autor: Gogol
Anzahl Seiten: 252
Bestseller: nein
Preis: 45.6
```

- (b) (Niveau Vererbung) Fahren Sie fort, indem Sie die Zugriffsrechte der privaten Attribute der `Buch`-Klasse ändern, sodass sie direkt in den Unterklassen von `Buch` zugänglich sind. Erstellen Sie dann die folgenden Unterklassen der `Buch`-klasse:

- `Roman`: gekennzeichnet durch ein Attribut, das angibt, ob es sich um eine Biografie handelt oder nicht. Dieses Attribut wird geschützt.

- Krimi: Unterklasse von Roman. Diese Unterklasse hat keine spezifischen Attribute.
- CoffeeTableBuch (illustrierte Werke): Unterklasse der Klasse Buch. Hat keine spezifischen Attribute.

Unterklassen sollten die Zugriffsrechte von Attributen und Methoden, die von den Klassen Buch und Roman geerbt wurden, nicht einschränken.

- Die in der Buch-Klasse implementierte Methode `berechne_preis` gibt den Standardpreis des Buches an.

Statten Sie Ihre Klassen des Weiteren mit folgenden `public` Methoden aus:

- Klasse Roman:
 - einem Konstruktor, der als Argumente die Parameter verwendet, die zum Initialisieren aller Attribute der Roman-Klasse verwendet werden
 - einem leeren virtuellen Destruktor
 - einer Anzeigemethode, die es ermöglicht, den Wert aller Attribute eines Romans anzuzeigen. Der Preis des Romans wird bei dieser Methode nicht angezeigt. Lassen Sie sich von dem folgenden Ausführungsbeispiel inspirieren, um Ihre Ausgabe zu erstellen.
- Klasse CoffeeTableBuch
 - einen Konstruktor der als Argumente die Parameter verwendet, die es ermöglichen, alle von Buch geerbten Attribute zu initialisieren.
 - ein leerer virtueller Destruktor
 - eine `berechne_preis`-Methode, die den Preis eines CoffeeTableBuchs berechnet, indem der Standardpreis um 30 Euro erhöht wird. z.B. ein CoffeeTableBuch von 100 Seiten (Standardpreis 30 Euro), welches ein Bestseller ist (Standardpreis $30+50 = 80$ Euro) wird für 110 Euro verkauft ($= 80+30$ Euro (die Erhöhung)).
- Klasse Krimi
 - einem Konstruktor, der als Argumente die Parameter verwendet, die zum Initialisieren aller Attribute der Roman-Klasse verwendet werden
 - eine `berechne_preis`-Methode, die den Preis eines Krimis berechnet, indem vom Standardpreis 10 Euro abgezogen werden. Ist der so ermittelte Preis negativ, beträgt der Arbeitspreis 1 Euro.

Die Preisberechnung in den Unterklassen Krimi und CoffeeTableBuch neu definierten Methoden `berechne_preis` muss die von der Klasse Buch geerbte Methode `berechne_preis` (Standardpreis) aufrufen.

Die `anzeige`-Methode der Roman-Klasse sollte auch die der Buch-Klasse aufrufen, um unnötige Duplizierung des allgemeinen Codes zu vermeiden.

Testen Sie Ihr Programm mit einer kleinen `main`, um etwa Folgendes zu erstellen:

```
Titel: Der Hund von Baskerville
Autor: A.C.Doyle
Anzahl Seiten: 221
```

Bestseller: nein
Dieser Roman ist keine Biographie
Preis: 56.3

Titel: Der Pate
Autor: A.Cuso
Anzahl Seiten: 367
Bestseller: ja
Dieser Roman ist keine Biographie
Preis: 150.1

Titel: Bretonischer Ruhm
Autor: J.L. Bannalec
Anzahl Seiten: 283
Bestseller: nein
Dieser Roman ist keine Biographie
Preis: 84.9

Titel: Geronimo: Erinnerungen
Autor: S.M. Barrett
Anzahl Seiten: 173
Bestseller: nein
Dieser Roman ist keine Biographie
Preis: 51.9

Titel: Europas Flüsse
Autor: C. Osborne
Anzahl Seiten: 150
Bestseller: nein
Preis: 75

(c) (Niveau Polymorphismus) Programmieren Sie nun eine Buchladen-Klasse, die über ein privates Attribut mit dem Namen „buecher_sammlung“ verfügt. Dieses Attribut ist eine Sammlung von Büchern, die als dynamisches Array dargestellt wird. Erstellen Sie ein Prototyp und definieren Sie dann die folgenden Methoden der Buchladen-Klasse:

- `void buch_hinzufuegen(Buch*)` , fügt der Sammlung einen (Zeiger auf) ein Buch hinzu;
- `void anzeigen() const`, das die `anzeigen()`-Methode (der Buch-Klasse oder ihrer abgeleiteten Klassen) aufruft;
- `void buchsammlung_leeren()` ermöglicht das Löschen aller in der Sammlung gespeicherten Bücher und gibt gleichzeitig deren Speicherplatz frei.

Ändern Sie die mit der Buch-Klasse verknüpfte Anzeigefunktion so, dass sie auch den Preis des Buchs anzeigt. Die Berechnung des anzuzeigenden Preises erfolgt über die Methode `berechne_preis()`. In der Buchladen-Klasse wird die Anzeige-Methode der Buch-Klasse polymorph für die Elemente in einer Büchersammlung aufgerufen.

Ändern Sie die Deklarationen der Anzeigemethoden für Bücher, sodass sie einer dynamischen Linkauflösung unterliegen können. Machen Sie dasselbe mit der `berechne_preis`-Methode (da `berechne_preis` von der Anzeige aufgerufen wird).

Im Prinzip enthält Ihre Datei „Buchladen.cpp“ nun alle notwendigen Werkzeuge, um die Objekte der heterogenen Sammlung polymorph zu verarbeiten. Testen Sie Ihren Code mit einer main, um Folgendes zu erhalten:

```
Titel: Der Hund von Baskerville
Autor: A.C.Doyle
Anzahl Seiten: 221
Bestseller: nein
Preis: 56.3
Dieser Roman ist keine Biographie

Titel: Der Pate
Autor: A.Cuso
Anzahl Seiten: 367
Bestseller: ja
Preis: 150.1
Dieser Roman ist keine Biographie

Titel: Bretonischer Ruhm
Autor: J.L. Bannalec
Anzahl Seiten: 283
Bestseller: nein
Preis: 84.9
Dieser Roman ist keine Biographie

Titel: Geronimo: Erinnerungen
Autor: S.M. Barrett
Anzahl Seiten: 173
Bestseller: nein
Preis: 51.9
Dieser Roman ist keine Biographie

Titel: Europas Flüsse
Autor: C. Osborne
Anzahl Seiten: 150
Bestseller: nein
Preis: 75
```

Ausgehend von folgendem Code:

```
int main(){
    Buchladen l;
    l.buch_hinzufuegen(new Krimi("Der Hund von Baskerville", "A.C.Doyle",
    221, false, false));
    l.buch_hinzufuegen(new Krimi("Der Pate", "A.Cuso", 367, true, false));
    l.buch_hinzufuegen(new Roman("Bretonischer Ruhm", "J.L. Bannalec", 283,
    false, false));
    l.buch_hinzufuegen(new Roman ("Geronimo: Erinnerungen", "S.M. Barrett",
    173, false, true));
    l.buch_hinzufuegen(new CoffeeTableBuch ("Europas Flüsse", "C. Osborne",
    150, false));
    l.anzeigen();
    l.lager_leeren();
    return 0;
}
```