

Allgemeine Angaben

Modulprüfung	Klausur	Datum: 03.02.2022
Modulname: Objektorientierte Programmierung	Modulnummer: 40050200	
Prüfungsdauer: 60 min + 15 min	Prüfer: Welp	

Name: _____ Vorname: _____

Matrikelnummer: _____

Bewertung

Aufgabe	1	2	3	4						Summe
Erreichbare Punkte	8	13	9	19						49
Erzielte Punkte										
Unterschrift Prüfer				Erzielte Punkte in %					Note	

Aufgabe 1: Von C nach C++

1. Schreiben Sie eine Funktion `wert` zur Berechnung des Wertes eines Baugrundstücks. Der Wert berechnet sich aus der Fläche des Grundstücks x 235,-EUR und der Wohnfläche der Immobilie x 2530,-EUR. Der Funktion soll als Parameter die Grundstücksfläche und optional (falls das Grundstück bereits bebaut ist) die Wohnfläche übergeben werden.

Lösen Sie dieses Problem, indem Sie

1. `wert` überladen
2. `wert` als Funktion mit Defaultparametern definieren.

Geben Sie für beide Lösungswege die Funktions**definitionen** an. (5 Punkte)

2. Was gibt folgendes Programmfragment auf dem Bildschirm aus? (3 Punkte)

```
long n[3] = {2000,3000,4000};  
long& r = n[1];  
long* p = n;  
r = r/10;  
p=p+2;  
*p = r + n[0]/2;  
cout << n[2] << " " << r << " " << *p << endl;
```

Aufgabe 2: Klassen und Objekte

Definieren Sie eine Klasse `Bruch` für die Darstellung von Brüchen. Der Zähler und der Nenner sollen als ganzzahlige Attribute in der Klasse gespeichert werden. Die Klasse soll über folgenden Eigenschaften verfügen:

- Der direkte Zugriff auf den Zähler und den Nenner eines Bruches soll nicht möglich sein. Der Zugriff soll über entsprechende setter- und getter-Methoden erfolgen. **Implementieren** Sie die Methoden als inline-Funktionen.
- Die Klasse soll über geeignete Konstruktoren zur Initialisierung von `Bruch`-Objekten verfügen. Ein `Bruch`-Objekt soll dabei sowohl über die Angabe von Zähler und Nenner als auch über eine ganze Zahl initialisiert werden können. Eine Objektinstanziierung ohne Parameter soll auch möglich sein. In diesem Fall soll das Objekt mit dem Wert 0 initialisiert werden (siehe auch Hinweis). **Implementieren** Sie die Methoden als inline-Funktionen.
- Die Klasse soll eine Methode `getDouble()` zur Verfügung stellen, die den Bruch als double-Wert zurückliefert. **Implementieren** Sie die Methode ausserhalb der Klasse.
- Ferner soll der einstellige `~`-Operator für die Klasse `Bruch` überladen werden, sodaß der Bruch mit dem größten gemeinsamen Teiler von Zähler und Nenner gekürzt werden kann. Es steht die Funktion `int ggt(int a, int b);` zur Verfügung. **Implementieren** Sie die Methode ausserhalb der Klasse. (13 Punkte)

Hinweis

Die Klasse `Bruch` soll z.B. folgendermaßen verwendet werden können

```
int main()
{
    Bruch b1(12,9);
    Bruch b2(5);
    Bruch b3;

    cout << "b1 = " << b1.getZaehler() << "/" << b1.getNenner()
         << " = " << b1.getDouble() << endl;
    cout << "b2 = " << b2.getZaehler() << "/" << b2.getNenner()
         << " = " << b2.getDouble() << endl;
    cout << "b3 = " << b3.getZaehler() << "/" << b3.getNenner()
         << " = " << b3.getDouble() << endl;

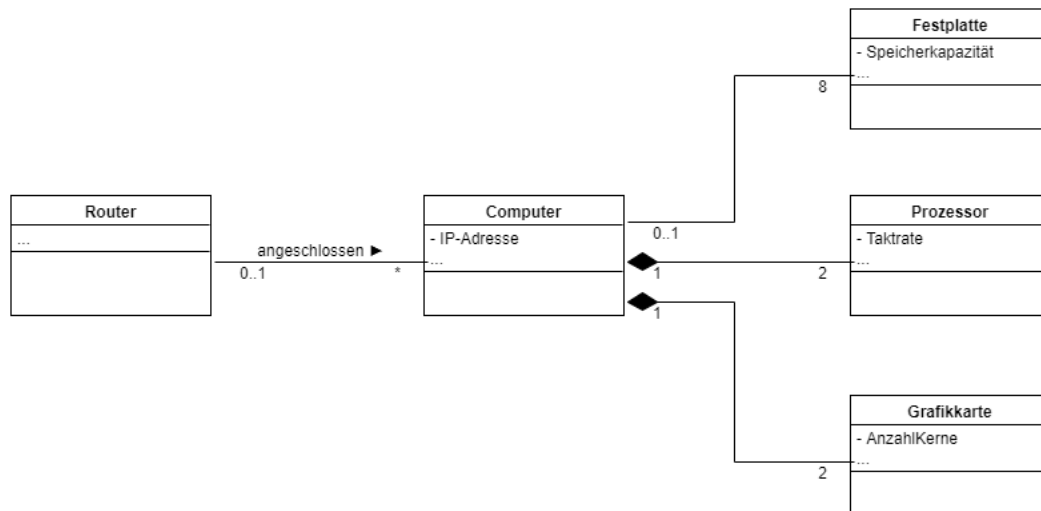
    b3 = ~b1;
    cout << "b1 = " << b1.getZaehler() << "/" << b1.getNenner()
         << " = " << b1.getDouble() << endl;
    cout << "b3 = " << b3.getZaehler() << "/" << b3.getNenner() << " = "
         << b3.getDouble() << endl;
}
```

Ausgabe:

```
b1 = 12/9 = 1.33333
b2 = 5/1 = 5
b3 = 0/1 = 0
b1 = 4/3 = 1.33333
b3 = 4/3 = 1.33333
```

Aufgabe 3: Objektbeziehungen und Templates

In einer Netzwerkmanagement-Software gibt es die Klassen Router, Computer, Festplatte, Prozessor und Grafikkarte, die entsprechend dem folgendem Klassendiagramm miteinander in Beziehung stehen.



1. Welche Attribute werden in der Klasse `Computer` benötigt um die Beziehungen zu den Klassen `Router`, `Festplatte`, `Prozessor` und `Grafikkarte` zu implementieren? Beantworten Sie die Frage, indem Sie die Klassendefinition für die Klasse `Computer` angeben (nur Attribute, Methoden sind nicht erforderlich). (4 Punkte)
2. Definieren Sie eine **Template-Funktion** `maximum`, die den größten Wert eines Arrays zurückliefert. Die Elemente des Arrays sollen von beliebigem Typ sein. (5 Pkt.)

mögliche Anwendung:

```
long a[4] = {2,5,3,4};
string s[3] = {"Bochum", "Dortmund", "Essen"};
cout << maximum(a,4) << endl;
cout << maximum(s,3) << endl;
```

Ausgabe:

```
5
Essen
```

Aufgabe 4: Vererbung und Polymorphismus

Gegeben seien folgende beiden Klassen zur Verwaltung von Wohngrundstücken und Gewerbegrundstücken. Auf beide Arten von Grundstücken wird eine Grundsteuer erhoben, die sich für Wohn- und Gewerbegrundstücke unterschiedlich berechnet. Z.B.:

$$\text{grundsteuer}(\text{wohngrundstück}) = (\text{flaeche} * 10 + \text{wohnflaeche} * 300) * \frac{\text{zinssatz}}{100}$$

$$\text{grundsteuer}(\text{gewerbegrundstueck}) = \text{flaeche} * \text{nutzungsgrad} * \frac{\text{zinssatz}}{100}$$

Der Zinssatz ist für Wohn- bzw. Gewerbegrundstücke unterschiedlich.

Wohngrundstueck	
- flaeche	/* in m^2 */
- wohnflaeche	/* in m^2 */
- zinssatz	/* in % */
+ berechneGrundsteuer	
+ ausgabe	
+ <u>setZinssatz</u>	

Gewerbegrundstueck	
- flaeche	/* in m^2 */
- nutzungsgrad	/* [1...100] */
- zinssatz	/* in % */
+ berechneGrundsteuer	
+ ausgabe	
+ <u>setZinssatz</u>	

1. Führen Sie ein Redesign mit der Zielsetzung durch, Gemeinsamkeiten unter Verwendung von Vererbung in einer Basisklasse auszulagern. Skizzieren Sie das neue Klassendiagramm. *Überlegen Sie genau, was wirklich gemeinsam ist.* (4 Punkte)
2. Geben Sie Klassendefinition für mindestens eine abgeleitete Klasse an. (5 Punkte)
3. Wie machen Sie die Methode `berechneGrundsteuer` zu einer rein virtuellen Methode? (2 Punkte)
4. Über welche Attribute verfügen Objekte der abgeleiteten Klasse? (3 Punkte)
5. Was wäre an folgender Implementierung der Methode `setZinssatz` falsch? (2 Punkte)

```
void Wohngrundstueck::setZinssatz(double zs)
{
    if(flaeche>1000)
        zinssatz=zs*2;
    else
        zinssatz=zs;
}
```

6. **Qt:** Angenommen Sie möchten, dass bei einem Klick mit der Maustaste im Fenster Ihres Programms ein Ton ausgegeben wird. Welche Vorkehrungen müssen Sie in Ihrer Fenster-(Widget-)klasse treffen um dies zu realisieren? (3 Punkte)
Mehrere Antworten möglich.

- ☐ Ihre Klasse muss von `QSound` abgeleitet sein.
- ☐ Ihre Klasse benötigt ein Attribut vom Typ einer Qt-Widget-Klasse (z.B. `QWidget`). Dieses verwenden Sie zum Aufruf der Methode `mousePressEvent`.
- ☐ Ihre Klasse muss von einer Qt-Widget-Klasse (z.B. `QWidget`) abgeleitet sein.

- ☐ Sie müssen eine Methode `onMouseClicked` implementieren.
- ☐ Sie müssen die virtuelle Methode `mousePressEvent` redefinieren.
- ☐ Ihre Klasse muss einen Slot implementieren, der auf ein Maus-Event reagiert.