

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 1 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe	1	2	3	4	5	6	Summe	Note
Maximum	19	18	15	8	25	15	100	
Erreicht								

Organisatorische Hinweise:

- Dauer der Klausur: 90 Minuten
- Es darf ein DinA4-Blatt mit handschriftlichen Notizen verwendet werden (Vorder- und Rückseite beschrieben).
- Es dürfen keine elektronischen Geräte wie z.B. Taschenrechner, Notebook, Handy usw. benutzt werden.

Verhalten während der Klausur:

- Beschriften Sie sofort das Deckblatt mit Ihrem Namen und Ihrer Matrikel-Nr.
- Verwenden Sie die vorgegebenen Klausuraufgabenzettel für Ihre Lösungen (ggf. auch die Rückseite).
- Benutzen Sie keinen Bleistift und keine rote Tinte. Entsprechende Aufzeichnungen werden nicht gewertet.
- Nutzen Sie die anfängliche Fragestunde, um Unklarheiten zu klären. Während der Bearbeitungszeit werden keine Fragen mehr beantwortet!
- Legen Sie Ihren Studierendenausweis gut sichtbar auf den Tisch.
- Sofern Sie ein (max. DinA4-)Hilfsblatt vorgefertigt haben, legen Sie es zum Ausweis.
- Melden Sie sich bitte mit Handzeichen, wenn Sie zur Toilette müssen (zeitgleich darf nur ein Klausurteilnehmer den Raum verlassen)
- Vermeiden Sie Störungen!

Genereller Hinweis zur Klausur:

- Halten Sie sich an die vorgegebenen Namenskonventionen.
- Achten Sie bitte auf einen strukturierten und gut lesbaren Code (d.h. Coding-Style).
- Achten Sie auf Effizienz beim Programmieren.
- Standard Bibliotheken dürfen als Eingebunden vorausgesetzt werden, alle anderen Header-Dateien müssen von Ihnen eingebunden werden.

Viel Erfolg!

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 2 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 1: Grundlage OO Programmierung [19 Punkte]

Eine Klasse namens **Artikel** soll nach folgender Beschreibung entworfen und implementiert werden:

- Die Klasse **Artikel** hat zwei private Attribute:
 1. *Name* vom Typ *string*.
 2. *Status* vom Typ *boolean*.
- Die Klasse **Artikel** hat
 1. einen Default Konstruktor.
 2. einen Overload Konstruktor mit Parameter vom Typ *string* und *boolean*.
 3. einen Destruktor.
- Die Klasse **Artikel** hat vier public Methoden:
 1. *ausleihen()* liefert keinen Rückgabewert. Sie setzt den *Status* intern auf *true* und gibt folgende Nachricht auf der Konsole aus:
„<Name des Buchs> wurde ausgeliehen“
 2. *zurueckgeben()* liefert einen Rückgabewert vom Typ *Boolean*. Die Methode setzt den *Status* intern auf *false* und quittiert mit *true*, wenn ein Artikel erfolgreich zurückgegeben wurde.
 3. *getStatus()* gibt den aktuellen *Status* aus.
 4. *setStatus(bool)* setzt den neuen Wert für *Status*.

- a) Entwerfen Sie die Klasse **Artikel** mit Hilfe von UML Klassendiagramm. Halten Sie sich bitte an die Namensgebung in der Beschreibung. [5 Punkte]

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 3 / 17	Name, Vorname:	Matr.-Nr.:

b) Wie sieht die Deklaration der Klasse **Artikel** in C++ aus? *[4 Punkte]*

Antwort:

c) Beim Kreieren eines neuen Objektes der Klasse **Artikel** soll das Objekt standardmäßig den Namen „Harry Potter“ bekommen, und der Status soll auf *false* gesetzt sein. Implementieren Sie das in C++ *[2 Punkte]*:

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 4 / 17	Name, Vorname:	Matr.-Nr.:

- d) Dem Overload Konstruktor werden ***Name*** und ***Status*** als Parameter übergeben. Implementieren Sie das in C++ [2 Punkte]:

Antwort:

- e) Implementieren Sie eine Getter-Methode und eine Setter-Methode für das Attribut ***Status***. Stellen Sie dabei sicher, dass die Getter-Methode wirklich keine Werte überschreiben kann. [3 Punkte]

Antwort:

- f) Implementieren Sie die Methode **`ausleihen()`**. [3 Punkte]

Antwort:

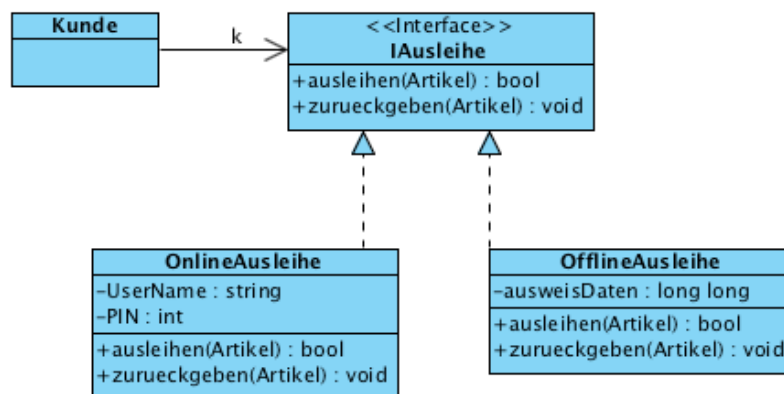
MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 5 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 2: Interface Polymorphismus [18 Punkte]

In einer Bibliothek kann ein Kunde bestimmte Artikel wie z.B. Bücher, CDs etc. ausleihen. Die Ausleihe geschieht entweder online über das Internet oder offline vor Ort. Dafür wurde ein Interface Design erstellt:

Hinweise:

- Achten Sie bei der Implementierung auf das Einbinden von benötigten Header-Dateien.
- Nutzen Sie die Implementierung der Klasse **Artikel** aus Aufgabe 1.
- Konstruktor und Destruktor müssen Sie nicht implementieren.



- a) Woran erkennen Sie eine Interface Klasse in C++? Nennen Sie drei wesentlichen Merkmale. [3 Punkte]

Antwort:

- b) Deklarieren Sie die Interface Klasse **IAusleihe** in C++. [3 Punkte]

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 6 / 17	Name, Vorname:	Matr.-Nr.:

c) Wie sieht die Deklaration der Provider Klasse ***OnlineAusleihe*** in C++ aus? *[4 Punkte]*

Antwort:

d) Implementieren Sie die Konsumer Klasse so, dass ein Artikel zuerst online, anschliessend offline ausgeliehen wird. *[5 Punkte]*

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 7 / 17	Name, Vorname:	Matr.-Nr.:

- e) Die Implementierung der Methode `ausleihen()` von den Klassen `OnlineAusleihe` und `OfflineAusleihe` sehen folgendermassen aus. Wie lautet die Konsolenausgabe nach dem Aufruf Ihres Konsumers in d)? [3 Punkte]

Hinweis: Überlegen Sie sich beliebige Werte für die Konstruktoren der Klassen.

```
bool OnlineAusleihe::ausleihen(Artikel artikel) {
    artikel.ausleihen(); // Methode der Klasse Artikel aus Aufgabe 1
    cout << "Online vom User: " << this->UserName << endl;
}
```

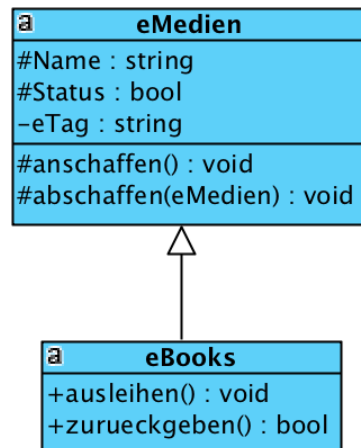
```
bool OfflineAusleihe::ausleihen(Artikel artikel) {
    artikel.ausleihen(); // Methode der Klasse Artikel aus Aufgabe 1
    cout << "Offline mit Ausweis: " << this->ausweisDaten << endl;
}
```

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 8 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 3: Vererbung Implementierung [15 Punkte]

Finden Sie Fehler in der Implementierung des folgenden Designs. Markieren Sie die Fehler und schreiben Sie die richtige Lösung auf:



```

// eMedien.h
class eMedien {
private:
    string Name;
    bool Status;
    string eTag;
protected:
    eMedien();
    virtual ~eMedien();
    void anschaffen();
    void abschaffen();
};
  
```

```

// eMedien.cpp
#include "eMedien.h"

eMedien::eMedien() {
    this->Name = "Harry";
    this->Status = false;
    this->eTag = "12345";
}

eMedien::~~eMedien() {
}

void eMedien::anschaffen() {
    cout << "angeschafft" << endl;
}

bool eMedien::abschaffen() {
    cout << Name << " abgeschafft" << endl;
}
  
```


MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 9 / 17	Name, Vorname:	Matr.-Nr.:

```
// eBooks.h
#include "eMedien.h"

class eBooks : eMedien{
public:
    eBooks();
    virtual ~eBooks();
    void ausleihen();
    bool zurueckgeben();
};
```

```
// eBooks.cpp

#include "eBooks.h"

eBooks::eBooks() {
}

eBooks::~eBooks() {
}

void eBooks::ausleihen() {
    return ->Status=true;
}

bool eBooks::zurueckgeben(){
    cout << this->eTag << " wieder da" << endl;
}
```

```
// main.cpp

#include "eBooks.h"

int main(){
    eBooks b;
    b.anschaffen();
    b.ausleihen();
    b.zurueckgeben();
    b.abschaffen();

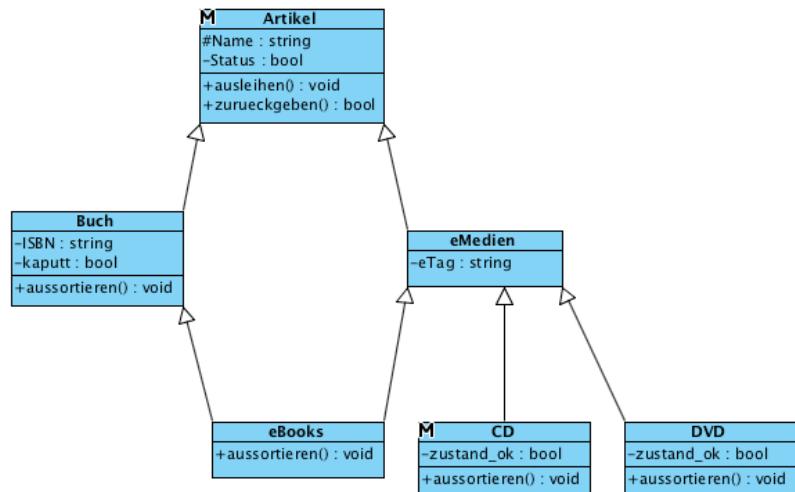
    eMedien m;
    m.anschaffen();
    m.abschaffen();
    cout << "Name: " << m.Name << endl;
    cout << "eTag: " << m.eTag << endl;
    m.ausleihen();
    m.zurueckgeben();

    return 0;
}
```

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 10 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 4: Vererbungshierarchie [8 Punkte]

Gegeben sei folgendes Design:



- a) In welcher Reihenfolge werden die Konstruktoren und Destruktoren der Basisklassen beim Erzeugen eines Objekts der Klasse **DVD** aufgerufen? [3 Punkte]

Antwort:

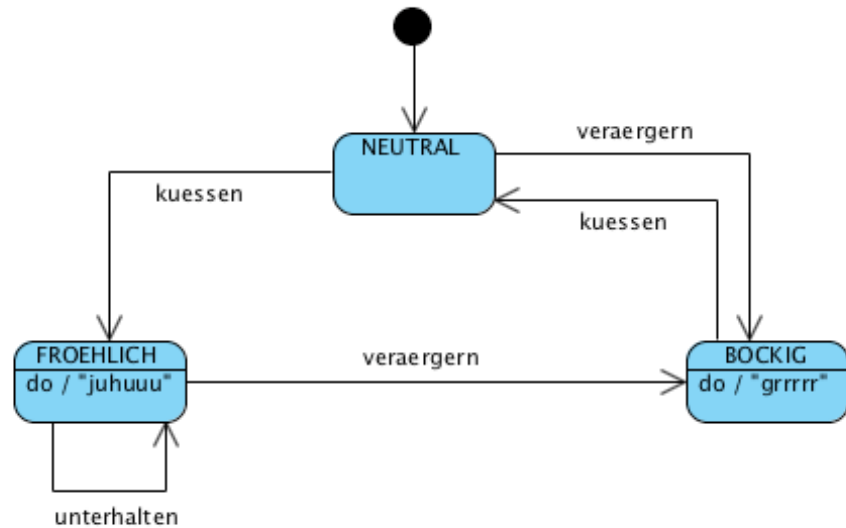
- b) In welcher Reihenfolge werden die Konstruktoren und Destruktoren der Basisklassen beim Erzeugen eines Objekts der Klasse **eBooks** aufgerufen? Welchen Namen trägt dieses wohlbekannte Problem? [5 Punkte]

Antwort:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 11 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 5: FSM Implementierung [25 Punkte]

Der Gemütszustand eines Kindes ist in folgender FSM beschrieben.



a) Implementieren Sie diese Zustandsmaschine in C++. [20 Punkte]

Antwort:

```
// FSMKind.h
```

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 12 / 17	Name, Vorname:	Matr.-Nr.:

Antwort:

```
// FSMKind.cpp
```

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 13 / 17	Name, Vorname:	Matr.-Nr.:

- b) Implementieren Sie einen Testfall in C++, der eine 100%-ige Transitionsüberdeckung der FSM erreicht. Benötigte Header-Dateien sind von Ihnen einzubinden. *[5 Punkte]*

Antwort:

```
// TestFSMKind.cpp
#include <assert.h>
```

```
}
```

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 14 / 17	Name, Vorname:	Matr.-Nr.:

Aufgabe 6: Grenzwertanalyse [15 Punkte]

Um Gebühren für zu spät zurückgegebene Artikel zu berechnen, ist die Methode ***double chargeFee(int)*** implementiert worden. Die Gebühren setzen sich folgendermassen zusammen:

<i>Verspätung</i>	<i>Gebühren</i>
1. bis 10. Tag	0.5 Euro / Tag
ab 11. Tag	1 Euro / Tag

Beispiel: chargeFee(3)=1.5 Euro

a) Bestimmen Sie die Äquivalenzklassen. [4 Punkte]

Antwort:

b) Führen Sie eine vollständige Grenzwertanalyse durch, d.h. wie sehen die Input-Daten und Output-Daten aus? Gehen Sie dabei von einer 16-Bit Integer aus. [5 Punkte]

Antwort:

<u>ÄK:</u>	<u>Input:</u>	<u>Output:</u>

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 15 / 17	Name, Vorname:	Matr.-Nr.:

- c) Suchen Sie sich drei Grenzwerte aus b) heraus und testen Sie damit die Methode ***chargeFee(int)***. Die Tests sollen in C++ implementiert sein. [6 Punkte]

Hinweis: Die Methode ***double chargeFee(int)*** befindet sich in der Klasse **Library**.

Antwort:

```
#include <assert.h>
#include "testLibrary.h"
#include "Library.h"           // implementiert die Methode chargeFee()
```

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 16 / 17	Name, Vorname:	Matr.-Nr.:

MT-PR2 SS2018	Klausur Programmieren 2 – 02.07.2018	DAI
Seite: 17 / 17	Name, Vorname:	Matr.-Nr.: