

Klausur
Einführung in die Programmierung
27.08.2021 (SoSe 2021)

Hinweise

1. Bearbeitungszeit: 90 Minuten
2. Zum Bestehen der Klausur sind 50 Punkte erforderlich.
3. Erlaubte Hilfsmittel: keine
4. Notieren Sie Ihre Lösungen auf eigenem Papier. Versehen Sie jedes Blatt mit Namen, Matrikelnummer, Studiengang und Seitenzahl. **Geben Sie bei jeder Lösung die Aufgabennummer und Teilaufgabe an.** Bitte geben Sie nicht die Lösungen von zwei verschiedenen Aufgaben auf einem Blatt an.
5. Bitte schreiben Sie deutlich mit einem schwarzen Stift.
6. Alle vorgegebenen und zu erstellenden Programmtexte beziehen sich auf die Programmiersprache Java.

Viel Erfolg!

Aufgabe	Max. Punktzahl	Erreichte Punktzahl
1 (Zahlendarstellung)	6	
2 (Programmieraufgabe)	30	
3 (Programmieraufgabe)	20	
4 (UML nach Java übertragen)	22	
5 (Speicherabbild)	22	
<i>Bonuspunkte</i>	<i>10</i>	
Gesamtpunktzahl	100+10=110	

Aufgabe 1 (Zahlendarstellung)**[6 Punkte]**

Beschreiben Sie am Beispiel der Zahl $n = -47$, wie man die Zweierkomplementdarstellung (8 Bit) einer Zahl berechnen kann.

Aufgabe 2 (Programmieraufgabe)**[30 Punkte]**

Implementieren Sie eine Klasse **ZeichenWiederholung** mit den folgenden Methoden:

i) **String wiederhole(String w, int n)**

Es wird ein Wort erzeugt, in dem **w** **n** mal wiederholt hintereinandergeschrieben wird.

Bspl.: Wird für **w** das Wort „abc“ und für **n** die Zahl 3 übergeben, so soll das Wort „abcabcabc“ zurückgegeben werden.

ii) **String berechnePräfix(String w, int i)**

Es wird das Wort zurückgegeben, das aus den ersten **i** Zeichen von **w** besteht.

Bspl.: Wird für **w** das Wort „abac“ und für **i** die Zahl 2 gewählt, so soll "ab" zurückgegeben werden.

iii) **boolean istWiederholt(String w)**

Es wird **true** zurückgegeben, wenn das Wort **w** von der Form **v v** ist, also aus dem zweifachen Hintereinanderschreiben eines Wortes **v** besteht.

Bspl.: Wird für **w** das Wort „abab“ gewählt, so soll **true** zurückgegeben werden ("abab" = "ab" "ab"), wenn **w** das Wort "abba" ist, so soll **false** zurückgegeben werden.

iv) **boolean istMehrfachWiederholt(String w)**

Es wird **true** zurückgegeben, wenn das Wort **w** durch das beliebig häufige (mindestens zweimalige) Hintereinanderschreiben eines Wortes **v** gebildet werden kann, d.h. $w = v v \dots v$.

Bspl.: Wird für **w** das Wort „abab“ oder "ababab" gewählt, so soll **true** zurückgegeben werden ("abab" = "ab" "ab", "ababab" = "ab" "ab" "ab"), wenn **w** das Wort "aba" ist, so soll **false** zurückgegeben werden.

Sie dürfen aus der Java-Bibliothek nur die Methoden *equals()*, *length()* und *charAt()* auf Strings benutzen. Für einen String **w** gibt die Methode **w.length()** die Länge von **w** und die Methode **w.charAt(i)** das **i**-te Zeichen von **w** zurück.

Aufgabe 3 (Programmieraufgabe)**[20 Punkte]**

In dieser Aufgabe soll eine Klasse bereitgestellt werden, die die Häufigkeit verschiedener Werte in einem zweidimensionalen Feld berechnet. Alle Werte in dem zweidimensionalen Feld können als nicht-negativ (d.h. ≥ 0) angenommen werden.

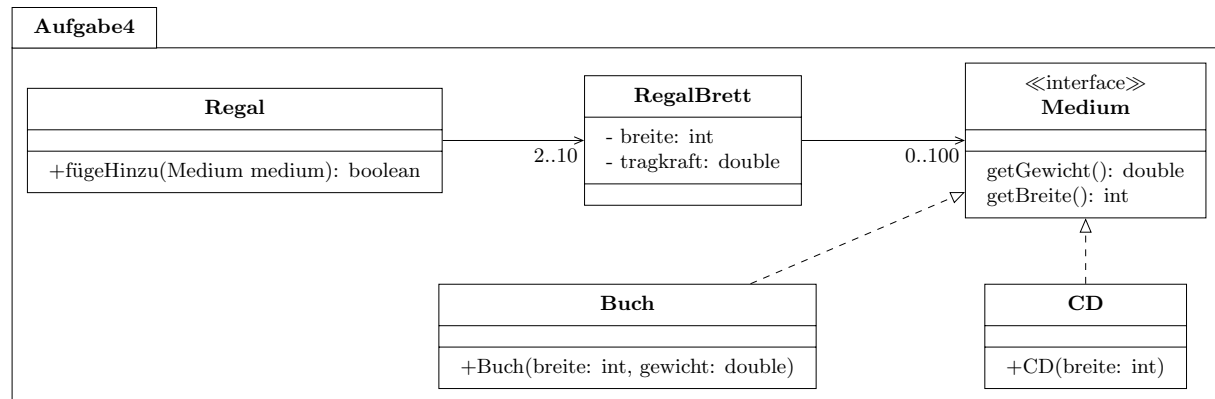
Implementieren Sie die folgenden Methoden:

- i) **`int max(int[][] field)`**: Es soll das Maximum der Werte in dem zweidimensionalen Feld `field` zurückgegeben werden.
- ii) **`int[] berechneHaeufigkeiten(int[][] field)`**: Es soll ein eindimensionales Feld zurückgegeben werden, in dem im *i*-ten Element steht, wie häufig die Zahl *i* in dem zweidimensionalen Feld `field` vorkommt. Der Inhalt des letzten Elements in dem zurückgegebenen Feld darf dabei nicht 0 sein. Sie dürfen annehmen, dass das Feld `field` wenigstens ein Element enthält.

Sie dürfen **keine** Klassen- oder Objektattribute oder Methoden aus der Java-Klassenbibliothek benutzen. Sie können aber weitere eigene (Hilfs-)Methoden hinzufügen.

Aufgabe 4 (UML nach Java übertragen)**[22 Punkte]**

Gegeben sei das folgende UML-Klassendiagramm.

Hinweis: Es dürfen **keine** Klassen der Java-Klassenbibliothek verwendet werden!

Programmieren Sie die angegebenen Klassen und Schnittstellen wie folgt:

1. Implementieren Sie in der Klasse **Regal** **nur** die **Attribute**, die zur Verwaltung der Assoziation zwischen den Klassen **Regal** und **RegalBrett** notwendig sind.
2. Implementieren Sie die Klassen **RegalBrett**, **Buch**, **CD** und das Interface **Medium** **vollständig** in Java. Ergänzen Sie notwendige Attribute und Methoden, insbesondere sinnvolle Konstruktoren und alle Attribute und Methoden zur Verwaltung der Assoziation zwischen den Klassen **RegalBrett** und **Medium**. Ein **Medium** darf hierbei nur dann in ein Objekt vom Typ **RegalBrett** aufgenommen werden, wenn die Summe der Breiten der Medien des Objekts nicht den Wert des Attributes **breite** überschreiten. Gleiches soll auch für die Summe der Gewichte und das Attribut **tragkraft** gelten. Eine **CD** wiegt immer 1,5.
3. **Implementieren** Sie nun die Funktion **fügeHinzu** der Klasse **Regal**, die das als Parameter übergebene Objekt vom Typ **Medium** in eines der zugeordneten Regalbretter einfügt. Es muss hierbei darauf geachtet werden, dass die Breite und Tragkraft des aufnehmenden Regalbretts nicht überschritten wird. Sollte das nicht möglich sein, so soll **false** zurückgegeben werden.

Aufgabe 5 (Speicherabbild)**[22 Punkte]**

Zeichnen Sie das Speicherabbild zu folgendem Programmcode zum [Zeitpunkt 1](#), d.h. vor der return-Anweisung der Methode **gleich**. Nehmen Sie hierbei an, dass **ihre_Matrikelnummer** ein String-Literal ist, dass **Ihre** Matrikelnummer enthält (wenn Ihre Matrikelnummer 0000000 ist, so soll hier also der String "0000000" eingetragen werden).

```
public class Test {
    public static void main(String[] args) {
        boolean erg;
        A b1 = new B(1);
        A b2 = new B(b1, b1.m);
        erg = b1.gleich(b2);
    }
}

public class A {
    protected static int x=1;
    public int m;

    public A(int m){
        this.m = m;
        x++;
    }
    public int get(){ return getM(); }
    public int getM(){ return m; }
    public boolean gleich(A a) {
        return a.m == this.m;
    }
}

public class B extends A {
    protected int m;
    protected String matrNr = ihre_Matrikelnummer;

    public B(int m){
        super(m + 2);
        this.m = m+1;
        x++;
    }
    public B(A a, int m){
        super(a.m + m);
        this.m = a.m;
        x++;
    }
    public int getM(){ return m; }
    public boolean gleich(A a) {
        boolean erg = getM() == a.getM();

        // Zeitpunkt 1
        return erg;
    }
}
```