

多项选择题将被标记如下。

(a) 预期答案和实际答案每匹配一次，你将得到1分。

你将得到1分。

(b) 如果预期的答案与你的答案不一致，你将得到1分的扣减。

你将被扣掉1分。

(c) 如果任务中的差异数高于差异数，或没有给出答案，将被罚1分。

(如果某项任务中的分歧数大于分歧数，或者你没有选择某个特定的答案选项。

Wintersemester 2020/21

你将得到0分。每项任务没有负分。

1 Prof. Dr.-Ing. S. Gudenkauf

在第二部分（练习7至10），公开问题的答案要写在每个练习的不同页面上。

在不同的页面上。指出每种情况下的任务编号。

每项任务都给出了最高分值。分数表示任务的难度。

分数表示任务的难度。

# ePrüfung Hausarbeit (eHA)

5.你可以在各自的答题纸上直接记下理解性问题。6.你可以使用两（2）张双面的、手写的、亲自写的笔记（Din-A4）。

(Din-A4)可作为辅助工具。

7) 不允许与第三方协商（除了审查员和出现技术问题时的联络人之外）。

在处理时间内与第三方（除审查员和技术问题时的联系人外）的讨论将被视为交流的尝试。

8) 请在相应的Moodle考试课程中，将你编辑好的答案以PDF格式的解决方案文件上传。

Moodle考试课程，并在那里交卷，完成电子考试。

Prüfer In: Prof. Dr.-Ing. S. Gudenkauf

Dauer: 120 Minuten

## Allgemeine Hinweise:

1. Erstellen Sie **Ihre Antworten handschriftlich und lesbar**. Beschriften Sie **jede Ihrer Antwortseiten** oben rechts deutlich mit Ihrem **Vor- und Nachnamen**, sowie Ihrer **Matrikelnummer**.
2. Der erste Teil der ePrüfung (Aufgaben 1 - 6) besteht aus **Multiple-Choice-Fragen**, bei denen eine oder mehrere Antworten richtig sein können. **Notieren Sie Ihre Antworten zu den Multiple-Choice-Fragen auf einer Antwortseite**. Geben Sie dazu jeweils die Nummer der Aufgabe sowie ihre gewählte Antwort an, z.B.

1. a), b)

Die Bewertung der Multiple-Choice-Fragen gestaltet sich wie folgt:
  - (a) Bei jeder **Übereinstimmung zwischen vorgesehener Antwort und tatsächlicher Antwort** erhalten Sie **1 Punkt**.
  - (b) Besteht **keine Übereinstimmung** zwischen der vorgesehenen Antwort und Ihrer Antwort, so erhalten Sie einen **Abzug von 1 Punkt**.
  - (c) Falls die Zahl der Nichtübereinstimmungen bei einer Aufgabe höher ist als die Zahl der Übereinstimmungen oder falls Sie keine vorgegebene Antwortmöglichkeit gewählt haben, so erhalten Sie 0 Punkte. Es gibt **keine negativen Punktzahlen** je Aufgabe.
3. Im zweiten Teil (Aufgaben 7 bis 10) sind die **Antworten zu offenen Fragen je Aufgabe auf jeweils separaten Seiten** zu schreiben. Geben Sie dazu jeweils die Nummer der Aufgabe an.
4. Zu jeder Aufgabe ist die **maximale Punktzahl** angegeben. Die Anzahl der Punkte gibt Auskunft über die **Schwierigkeit** der Aufgabe.
5. Verständnisfragen können Sie direkt an Ihren Antworten auf dem jeweiligen Antwortblatt notieren.
6. Als Hilfsmittel dürfen Sie **zwei (2) doppelseitig beschriebene, handschriftliche, und persönlich verfasste Aufzeichnung (Din-A4)** verwenden.
7. Die **Absprache** mit Dritten (mit Ausnahme von Prüfer\_in und Ansprechpartner bei technischen Störungen) während der Bearbeitungszeit wird als **Täuschungsversuch** gewertet.
8. Laden Sie bitte Ihre bearbeiteten Antwortblätter als **eine PDF-Lösungsdatei** im zugehörigen Moodle-Prüfungskurs hoch und geben Sie dort ab, um die ePrüfung abzuschließen.

- 关于标识符的命名，哪些说法是正确的？
- a) 识别码可以以字母或下划线开头。
  - b) 标识符只能由字母和下划线组成。
  - c) 标识符必须至少由两个字符组成。
  - d) 标识符不能是保留字（例如，公共或休息）。
  - e) Hello和Hello是不同的标识符。

## 1. Grundelemente von Java

(Multiple-Choice, 5 Punkte)

Welche Aussagen über die Benennung von Bezeichnern **treffen zu**?

- a) Bezeichner können mit einem Buchstaben oder einem Unterstrich beginnen.
- b) Bezeichner dürfen nur aus Buchstaben und Unterstrichen bestehen.
- c) Bezeichner müssen mindestens aus zwei Zeichen bestehen.
- d) Bezeichner dürfen keine reservierten Wörter (z.b. `public` oder `break`) sein.
- e) `Hallo` und `hallo` sind unterschiedliche Bezeichner.

## 2. Typkonvertierung

以下哪种类型转换是不必要的，因为它们在需要时是隐式执行的？

(Multiple-Choice, 5 Punkte)

Welche der folgenden Typkonvertierungen sind **unnötig**, da sie im Bedarfsfall implizit durchgeführt werden?

- a) `(double) 6.3f`
- b) `(long) 6.3`
- c) `(short) 6L`
- d) `(float) 6`
- e) `(double) 6L`

## 3. Wertzuweisungen

给出的是以下Java代码片段。

(Multiple-Choice, 5 Punkte)

Gegeben sei der folgende Java Codeausschnitt:

```
1  int x;  
2  x = 27;  
3  System.out.println(x+3);  
4  x = x+1;  
5  x = 3.7;
```

哪些语句代表有效的值分配？

Welche Anweisungen stellen **gültige Wertzuweisungen** dar?

- a) Die Anweisung in Zeile 1.
- b) Die Anweisung in Zeile 2.
- c) Die Anweisung in Zeile 3.
- d) Die Anweisung in Zeile 4.
- e) Die Anweisung in Zeile 5.

给出类Kreis和一个使用类Kreis的主程序KreisTest。

#### 4. Methoden

(Multiple-Choice, 5 Punkte)

Gegeben sei die Klasse Kreis sowie ein Hauptprogramm KreisTest, das die Klasse Kreis verwendet:

```
1 package pl.exams.methods;
2 public class Kreis {
3     private double r;
4     public Kreis (double r) {
5         this.r = r;
6     }
7     /* Methode umfang() hier einfüegen */
8 }
```

Kreis.java

```
1 package pl.exams.methods;
2 public class KreisTest {
3     public static void main(String[] args) {
4         Kreis k = new Kreis(5);
5         double umfang = k.umfang();
6         System.out.println(umfang);
7     }
8 }
```

KreisTest.java

当插入类Kreis中时，类Kreis的方法umfang()的下列哪种实现在给定的主程序KreisTest中工作时没有错误？

Welche der nachfolgenden Implementierungen der Methode umfang() der Klasse Kreis funktionieren **fehlerfrei mit dem gegebenen Hauptprogramm** KreisTest, wenn Sie in der Klasse Kreis eingefügt werden?

- a) public double umfang(double r){return 2\*r\*Math.PI;}
- b) public static double umfang() {return 2\*r\*Math.PI;}
- c) public float umfang()return 2\*r\*Math.PI;
- d) public static double umfang(double r) {return 2\*r\*Math.PI;}
- e) public double umfang() {return 2\*r\*Math.PI;}

## 5. Methoden

(Multiple-Choice, 5 Punkte)

Gegeben sei das folgende Programm:

```
1 public class Klausur {
2
3     // Methode (1)
4     public int aufgabe( String text ) {
5         // Aufgabe anzeigen ...
6     }
7
8     // Methode (2)
9     public int aufgabe( String text, String hilfe ) {
10        // Aufgabe mit Loesungshilfe anzeigen ...
11    }
12
13    // Methode (3)
14    public double note( int punkte ) {
15        // Benotung ermitteln ...
16    }
17
18    // Methode (4)
19    public String toString() {
20        return "Eine spannende Klausur!";
21    }
22
23    public static void main (String[] args){
24        Klausur k = new Klausur();
25        int p1 = k.aufgabe("Aufgabentext 1 ...");
26        int p2 = k.aufgabe("Aufgabentext 2 ...");
27        System.out.println( k.note(p1+p2) );
28    }
29 }
```

Welche Aussagen über die Methoden des Programms **treffen zu**?

- |   |                           |
|---|---------------------------|
| a) Methode (1) ist überschrieben.                         | 关于该方法的方法，哪些说法是正确的？        |
| b) Methode (1) ist überladen.                             | a) 方法 (1) 是超载的。           |
| c) Methode (2) ist überschrieben.                         | b) 方法 (1) 是超载的。           |
| d) Methode (2) ist überladen.                             | c) 方法 (2) 是超载的。           |
|   | d) 方法 (2) 是超载的。           |
|   | e) 方法 (1) 和方法 (2) 都没有被覆盖。 |
| e) Weder Methode (1) noch Methode (2) sind überschrieben. |                           |

## 6. Vererbung

(Multiple-Choice, 5 Punkte)

Welche **Zugriffsmodifikatoren** darf eine **Instanzvariable** in einer **Superklasse** (Oberklasse) in Java haben, damit mögliche Subklassen (Unterklassen) direkt auf diese geerbte Eigenschaft zugreifen können?

- a) public
- b) private
- c) package
- d) protected
- e) hidden

在Java中，一个实例变量在超类（superclass）中可以有哪些访问修改器，以便可能的子类（subclasses）可以直接访问这个继承的属性？  
可以直接访问这个继承的属性？

## 7. Operatoren

(Offene Aufgabe, 10 Punkte)

Gegeben seien die folgenden Variablen `i` und `x`:

```
1 int i = 2;  
2 float x = 4;
```

在两个变量*i*和*x*的基础上，确定以下表达式的结果值和结果类型。将每种情况下的值指定为相应类型的有效的Java字符符号。

Bestimmen Sie jeweils auf Basis der beiden Variablen `i` und `x` den **Ergebniswert** und den **Ergebnistyp** der folgenden Ausdrücke. Geben Sie die Werte jeweils als gültige Java-Literale des entsprechenden Typs an.

a) (2 Punkte)

```
1 (i + x) / 2.0 - (x + i / 2)
```

b) (2 Punkte)

```
1 i % 5 >= 2 ? x-- : --x
```

c) (2 Punkte)

```
1 "i + 3.0 * x" + i / 4
```

d) (2 Punkte)

```
1 (i > x) || (x >= i + 1)
```

e) (2 Punkte)

```
1 (int) (3 * x / 5) * i - 1
```

## 8. Gültigkeit, Sichtbarkeit und Existenz von Variablen

(Offene Aufgabe, 10 Punkte)

Gegeben sei folgender Programmcode:

```
1 public class Sichtbarkeit {
2
3     public static int multiplikator = 10;
4
5     public static int max(int a, int b) { /*3*/
6         if (a > b) {
7             return a;
8         } else {
9             return b;
10        }
11    }
12
13    public static int berechne(int x) { /*2*/
14        int y = multiplikator * max(x, 0); /*4*/
15        return y;
16    }
17
18    public static void main(String[] args) { /*1*/
19        int n = Tastatur.liesInt();
20        System.out.println(berechne(n));
21    }
22 }
```

针对给定的程序代码，回答下列任务。

Beantworten Sie folgende Aufgabenstellungen zum gegebenen Programmcode:

- Welche Variablen sind an der jeweiligen Stelle /\*1\*/ bis /\*4\*/ des Programmcodes **existent**? (4 Punkte)
- Welche Variablen sind an der jeweiligen Stelle /\*1\*/ bis /\*4\*/ des Programmcodes **sichtbar**? (4 Punkte)
- In welchem Fall wäre die Klassenvariable `multiplikator` in den Methoden der Klasse **gültig, aber nicht sichtbar**? (1 Punkt)
- Inwiefern stellt die Verwendung der Klassenvariable `multiplikator` im gegebenen Programmcode eine **mögliche Fehlerquelle** dar? Wie könnte das Programm **verbessert** werden, um diese Fehlerquelle zu beseitigen? (1 Punkt)

a) 哪些变量存在于程序代码的相应位置/\*1\*/至/\*4\*/? (4分)

b) 哪些变量在程序代码的相应位置/\*1\*/至/\*4\*/可见? (4分)

c) 在哪种情况下，类的变量乘数在类的方法中有效，但不可见? (1分)

d) 在给定的程序代码中，使用班级变量乘法器以何种方式代表了一个可能的错误来源? 如何改进该方案以消除这一错误来源? (1分)

## 9. Verschiedenes

(Offene Aufgabe, 15 Punkte)

Gegeben sei folgendes Codebeispiel:

```
1 Integer i = new Integer(123);
2 String str = "123";
3 int k = Integer.parseInt(str);
4 int n = str.length() * k++;
5 if ( !(k<10) )
6 n += (-k);
```

注意事项。Integer类包含在Java类库中，它封装了一个基本类型int的单一值。方法parseInt将一个String转换为相应的int值。

**Hinweise:** Die Klasse Integer ist in der Java-Klassenbibliothek enthalten und kapselt einen einzelnen Wert vom Basistyp int. Die Methode parseInt wandelt einen String in einen entsprechenden int-Wert um.

Beantworten Sie die folgenden **Fragen** zum gezeigten Codebeispiel.

- Ist die Methode parseInt eine **Instanzmethode** oder eine **Klassenmethode**?
  - Welche **Signatur** hat die Methode parseInt?
  - Welche **Signatur** hat der Konstruktor der Klasse Integer?
  - Zu welcher Klasse **gehört** die Methode length?
  - Ist die Methode length eine **Instanzmethode** oder eine **Klassenmethode**?
  - Welche der verwendeten Variablen sind **Referenzvariablen**?
  - Welche **unäre Operatoren** werden verwendet?
  - Welche **binäre Operatoren** werden verwendet?
  - Wie könnte die Anweisung k++ **alternativ ausgedrückt** werden?
  - Meldet der Compiler einen Fehler, wenn der Variablenname str in allen Zeilen des Codebeispiels in **Großbuchstaben** geschrieben ist?
  - Meldet der Compiler einen Fehler, wenn der Klassenname Integer in allen Zeilen des Codebeispiels in **Großbuchstaben** geschrieben ist?
- parseInt方法是一个实例方法还是一个类方法?
  - parseInt方法的签名是什么?
  - 整数类的构造函数的签名是什么?
  - 该方法的长度属于哪个类别?
  - 该方法的长度是一个实例方法还是一个类方法?
  - 使用的哪些变量是参考变量?
  - 使用哪些单数运算符?
  - 使用了哪些二进制运算符?
  - 语句k++可以如何替代表达?
  - 如果变量名称str在代码示例的所有行中都用大写字母书写，编译器是否报告错误?
  - 如果类名integer在代码示例的所有行中都用大写字母书写，编译器是否报告错误?



将实现一个学生注册和输出的方案。

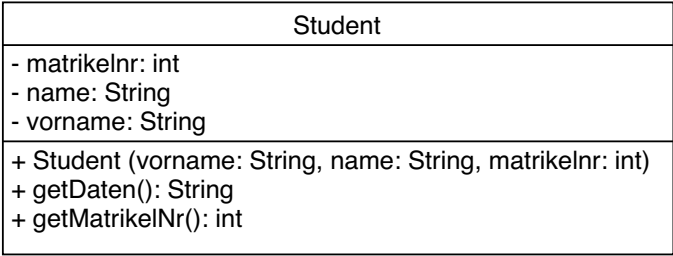
10. Programmierung

(Offene Aufgabe, 20 Punkte)

Es soll ein Programm zur Erfassung und Ausgabe von Studierenden realisiert werden.

考虑到以下子任务，对应用程序进行编程。  
**Programmieren** Sie die Anwendung unter Berücksichtigung der folgenden Teilaufgaben.

- a) Erstellen Sie eine **Java-Klasse** `Student`: Studierende haben einen Vornamen, einen Nachnamen und eine Matrikelnummer. Berücksichtigen Sie die in Abbildung 1 dargestellten Variablen und Methoden der Klasse: (10 Punkte)
- a) 创建一个Java类Student：学生有一个名字、一个姓氏和一个预科生编号。和一个预科生编号。考虑一下图1所示的类的变量和方法。(10分)



- 方法Student是该类的构造函数，允许创建Student类型的新对象。传递的参数firstname、name和matrikelnr将类的实例变量初始化为初始值。

- 方法getData以下列形式返回学生的数据，即一个字符串。

Abbildung 1: UML-Klassendiagramm zur Klasse Student.

- Die Methode `Student` ist der Konstruktor der Klasse und erlaubt das Erzeugen neuer Objekte vom Typ `Student`. Die übergebenen Parameter `vorname`, `name` und `matrikelnr` initialisieren die Instanzvariablen der Klasse mit Anfangswerten.
- Die Methode `getData` gibt die Daten eines Studierenden als `String` in folgender Form wieder:

Dijkstra, Edsger (21345)

- Die Methode `getMatrikelnr` gibt die Matrikelnummer eines Studierenden als `int`-Wert wieder.

- `getMatrikelnr`方法将学生的入学编号作为一个int值返回。

b) 编写一个主程序StudentListManager: 该程序应首先为用户提供输入一个或多个新学生的可能性。随后 所有雇员的表格输出, 其形式如下。(10分)

- b) Schreiben Sie ein **Hauptprogramm** StudentListManager: Das Programm soll dem Benutzer zunächst die Möglichkeit bieten, einen oder mehrere neue Studierende zu erfassen. Anschließend soll eine tabellarische Ausgabe aller Mitarbeiter in der folgenden Form erstellt werden: (10 Punkte)

```
---
Studierende
---
Lovelace, Ada (12345)
Boole, George (12541)
Dijkstra, Edsger (21345)
---
```

对于StudentListManager类的实现, 你可以使用以下程序框架。

Für die Implementierung der Klasse StudentListManager können Sie folgendes Programmgerüst verwenden:

```
1 public class StudentListManager {
2     public static void main(String[] args) {
3         /* Ihre Implementierung hier einfüegen */
4     }
5 }
```

Programmgerüst der Klasse StudentListManager.

Die main-Methode der Klasse StudentListManager soll dabei wie folgt implementiert werden:

1. Anlegen eines **Arrays** studList für Objekte des Typs Student.
2. Anlegen einer **int-Variable** anzahl, die den Index des ersten freien Platzes im Array studList angibt (Anzahl Studierende).
3. Anlegen einer geeigneten **Schleife**, in der je Schleifendurchlauf ...
  - Von der Tastatur der Vorname, der Nachname und die Matrikelnummer eines Studierenden **eingelezen** wird.
  - Mit den eingelesenen Werten ein **neues Objekt** neu vom Typ Student instanziiert wird.
  - Das Objekt neu am Ende des Arrays studList **eingefügt** wird.
  - Der **Index** anzahl aktualisiert wird.
  - Der Benutzer gefragt wird, ob er einen **weiteren Studierenden hinzufügen** möchte. (In dem Fall erfolgt ein erneuter Schleifendurchlauf.)
4. **Ausgabe** sämtlicher Studierenden im Array studList.

StudentListManager类的主方法要实现如下:

1.为学生类型的对象创建一个数组studList

2.创建一个int变量anzahl, 它指定了数组studList中第一个空闲位置的索引。阵列 (学生人数)。

3.创建一个合适的循环, 在这个循环中, 每个循环通道.....。

4.输出数组studList中所有的学生。

- 学生的名、姓和预科生编号从键盘上读入。  
是在读。

- 一个新的学生类型的对象被实例化, 带有读入的值。

- 新的对象被插入到studList数组的末端。

- 索引号被更新。

- 用户被问及是否要添加另一个学生。(在这种情况下, 循环会被重复)。