



I Grundsätze und Formen der Leistungsbewertung

1.1 Allgemeine Grundsätze der Leistungsbewertung

Die Leistungsbewertung bezieht sich auf die im Unterricht behandelten Inhalte und die dort erworbenen Kompetenzen. Grundlage sind die schulinternen Lehrpläne für die Sekundarstufe I und II sowie der geltende Lehrplan für das Fach Informatik in der Sekundarstufe II.

Zu Beginn jedes Schuljahres ist die Lehrkraft verpflichtet, die Schülerinnen und Schüler über die Anforderungen, die Arten der Leistungsüberprüfungen, die Bewertungskriterien sowie die Bildung der Note zu informieren.

1.2 Formen der Leistungsbewertung

Die Zeugnisnote setzt sich in der Regel aus den „**Schriftlichen Arbeiten**“ und den „**Sonstigen Leistungen im Unterricht**“ zusammen.

Die „**Schriftlichen Arbeiten**“ bestehen aus den Leistungen in den Kursarbeiten bzw. Klausuren.

Die „**Sonstigen Leistungen im Unterricht**“ umfassen u. a. folgende Formen:

- Beteiligung an Unterrichtsgesprächen
- Referate und Präsentationen
- Arbeitsphasen (Einzel-, Partner- und Gruppenarbeit)
- Projektarbeit
- Hausaufgaben
- Lernkontrollen, Test
- Heftführung (digital/Papierform)
- Portfolio
- Dokumentation von Projekten
- Protokolle
- Moderationen
- Rollenspiele



II Leistungsbewertung in der Sekundarstufe I

II.1 Formen der Leistungsbewertung in der Sekundarstufe I

Die Zeugnisnote im Differenzierungskurs Informatik setzt sich in etwa gleichen Teilen aus den „Schriftlichen Arbeiten“ und den „Sonstigen Leistungen im Unterricht“ zusammen.

Die Teilnote der „Schriftlichen Arbeiten“ wird aus den beiden Kursarbeiten ermittelt, wobei eine Kursarbeit durch ein Projekt ersetzt werden kann. Siehe dazu auch II.2.g.

Die Teilnote der „Sonstigen Mitarbeit im Unterricht“ umfasst die in I.2 aufgeführten Formen, deren Bewertungskriterien in II.3 näher erläutert werden.

II.2 Form und Bewertung von Kursarbeiten

2.a Inhalte der Kursarbeiten

Die Inhalte der Kursarbeiten beziehen sich auf den Unterrichtsstoff der vorangegangenen Unterrichtseinheiten unter Einbeziehung von Basiswissen vorheriger Inhalte im Differenzierungsbereich Informatik. Die Unterrichtsinhalte sind im schulinternen Lehrplan für das Fach Informatik in der Sekundarstufe I festgelegt.

2.b Anzahl und Dauer der Kursarbeiten

Die Termine für die Kursarbeiten im Differenzierungsbereich werden durch die Mittelstufenleitung jeweils zu Beginn eines Halbjahres bekannt gegeben.

Jgst.	Anzahl	Dauer	PC-Einsatz	Bemerkung
8	2 / HJ	45-90min	Bis zu 50 % der Aufgaben dürfen Aufgaben sein, die mit Hilfe des PCs bearbeitet werden sollen.	Eine Kursarbeit pro SJ soll durch ein Projekt ersetzt werden (z. B. HTML-Projekt)
9	2 / HJ	45-90min	Bis zu 50 % der Aufgaben dürfen Aufgaben sein, die mit Hilfe des PCs bearbeitet werden sollen.	Eine Kursarbeit pro SJ soll durch ein Projekt ersetzt werden (z. B. Lego-Projekt)



2.c Vorbereitung der Kursarbeiten

Der Termin der Kursarbeit wird rechtzeitig bekannt gegeben. Die Schülerinnen und Schüler erhalten in der Sekundarstufe I spätestens eine Woche vor der Kursarbeiten eine inhaltliche Übersicht.

2.d Aufgabenstellung und Aufgabenauswahl

Formulierungen und Aufgabenarten sollten den Schülerinnen und Schülern aus dem Unterricht weitestgehend bekannt sein. Die Schwierigkeit der Aufgaben sollte sich nach folgender Aufteilung richten:

Anforderungsbereich I (Reproduktion)	ca. 50 %
Anforderungsbereich II (Anwendung)	ca. 40 %
Anforderungsbereich III (Transfer)	ca. 10 %

2.e Korrektur und Bewertung

In den Kursarbeiten der Schülerinnen und Schüler werden sowohl die Fehler als auch richtige Ergebnisse, Rechenschritte und Teilaspekte der Aufgaben deutlich gekennzeichnet. Die Fehlerkennzeichnung orientiert sich dabei an der üblichen Kennzeichnung im Sinne der Richtlinien und Lehrpläne. Bei Aufgaben, die am PC bearbeitet werden, reicht ein angemessener Hinweis auf den Fehler aus.

Die Teilaufgaben der Kursarbeiten und Klausuren werden mit Punkten bewertet und die zu erreichende Punktzahl wird auf dem Aufgabenblatt angegeben.

Die Vergabe der Noten richtet sich nach der erreichten Gesamtpunktzahl, wobei sich diese an folgendem Notenschlüssel orientiert.

(1) Sehr gut	87,5 % - 100 %
(2) Gut	75 % - 87,5 %
(3) Befriedigend	62,5 % - 75 %
(4) Ausreichend	50 % - 62,5 %
(5) Mangelhaft	20 % - 50 %
(6) Ungenügend	weniger als 20 %

Kleinere Anpassungen sind möglich, z.B. das gleichmäßige Vergrößern der Prozentspannen. Grundsätzlich muss aber eine Arbeit, in der 50 % der Gesamtpunkte erreicht werden, mindestens mit der Note „ausreichend“ bewertet werden. Die Note „ungenügend“ darf nur vergeben werden, wenn weniger als 20 % der Gesamtpunkte erreicht wurde.



2.f Nachbereitung

Die Kursarbeit wird in der Regel ausführlich besprochen und ggf. eine Musterlösung ausgegeben. Die Schülerinnen und Schüler erhalten ausreichend Gelegenheit zum Klären ihrer Fragen.

2.g Ersatz von Kursarbeiten durch Projekte

In den Differenzierungskursen der Sekundarstufe I soll in jedem Schuljahr eine Kursarbeit durch ein Projekt ersetzt werden. Folgende Übersicht dient der Orientierung und ist nicht als verpflichtend anzusehen:

Differenzierungskurs 8 – HTML-Projekt

Differenzierungskurs 9 – LegoMindstorms-Projekt

Welche Kursarbeiten im jeweiligen Schuljahr durch Projekte ersetzt werden, entscheidet die Fachkonferenz auf Vorschlag der Fachlehrkräfte der betreffenden Kurse.

2.h Hilfsmittel

Hilfsmittel sind grundsätzlich keine erlaubt. Folgende Hilfsmittel kann die Lehrerin bzw. der Lehrer aber für die jeweilige Kursarbeit gestatten:

- Einsatz einer bestimmten Software (z. B. MS Excel) unter Verwendung des Computers im Klausurmodus,
- Taschenrechner,
- Befehlsübersichten zu verwendeten Sprachen (HTML, SQL, PHP, UML usw.),
- Wörterbuch der deutschen Sprache bzw. Fremdwörterbuch.

Über den Einsatz anderer Hilfsmittel entscheidet die Fachkonferenz.



II.3 Formen und Bewertung der „Sonstigen Leistungen im Unterricht“

In der folgenden Übersicht werden die Bewertungskriterien der wichtigsten „Sonstigen Leistungen im Unterricht“ vorgestellt.

Form der „Sonstigen Leistung im Unterricht“	Bewertungskriterien
Beteiligung an Unterrichtsgesprächen	<ul style="list-style-type: none">✓ Qualität der Unterrichtsbeiträge✓ Quantität der Unterrichtsbeiträge✓ Verwendung der Fachsprache
Referate und Präsentationen	<ul style="list-style-type: none">✓ Verständlichkeit des Vortrags✓ Qualität des Vortrags✓ sinnvoller Einsatz von Medien✓ Arbeitsaufteilung bei Gruppenvorträgen✓ Qualität und Umfang des Handouts✓ Verwendung der Fachsprache
Arbeitsphasen (Einzel-, Partner-, Gruppenarbeit), Projektarbeit	<ul style="list-style-type: none">✓ Zuverlässigkeit und Zielstrebigkeit✓ Engagement und Selbständigkeit✓ Kooperation untereinander✓ Qualität des Resultats✓ Verwendung der Fachsprache
Heftführung (digital/Papierform), Portfolio und Projektdokumentationen, Hausaufgaben, Protokolle	<ul style="list-style-type: none">✓ Vollständigkeit✓ Qualität der Darstellung✓ Qualität der Inhalte✓ Pünktlichkeit (rechtzeitige Abgabe usw.)✓ Verwendung der Fachsprache

Eine Schwerpunktsetzung zur gezielten Förderung, z.B. bei Referaten, ist möglich und erwünscht. Diese sollte den Schülerinnen und Schülern rechtzeitig im Vorfeld mitgeteilt werden.



III Leistungsbewertung in der Sekundarstufe II

III.1 Fachspezifische Formen der Leistungsbewertung

Die Zeugnisnote im Grundkurs Informatik setzt sich in etwa gleichen Teilen aus den „Schriftlichen Arbeiten“ und den „Sonstigen Leistungen im Unterricht“ zusammen, sofern Informatik als schriftliches Fach belegt wird. Wird es als mündliches Fach belegt, entspricht die Endnote der Teilnote der „Sonstigen Leistungen im Unterricht“.

Die Teilnote der „Schriftlichen Leistung“ wird in der Einführungsphase aus einer Klausur pro Halbjahr, in der Qualifikationsphase (mit Ausnahme des zweiten Halbjahres der Jahrgangsstufe 12) aus zwei Klausuren ermittelt. Im zweiten Halbjahr der Jahrgangsstufe 11 kann eine Klausur gemäß der APOGOst durch eine Facharbeit ersetzt werden.

Die Teilnote der „Sonstigen Leistungen im Unterricht“ umfasst die in I.2 aufgeführten Formen, deren Bewertungskriterien in III.3 näher erläutert werden.

III.2 Form und Bewertung von Kursarbeiten

2.a Inhalte der Klausuren

Die Inhalte der Klausuren richten sich nach den Unterrichtsinhalten, die im Kernlehrplan für das Fach Informatik in der Sekundarstufe II und insbesondere im schulinternen Lehrplan für das Fach Informatik in der Sekundarstufe II festgelegt sind.

2.b Anzahl und Dauer der Kursarbeiten

Die Termine für die Klausuren in der Sekundarstufe II werden durch die Oberstufenleitung jeweils zu Beginn eines Halbjahres bekannt gegeben.

Jgst.	Anzahl	Dauer	PC-Einsatz	Bemerkung
10.1 und 10.2	1 / HJ	90min	Bis zu 50 % der Aufgaben dürfen Aufgaben sein, die mit Hilfe des PCs bearbeitet werden sollen.	
11.1 und 11.2	2 / HJ	90min	Keine PC-Nutzung.	
12.1	2	135min	Keine PC-Nutzung.	
12.2	1	180min	Keine PC-Nutzung.	Abitur-Vorklausur



2.c Vorbereitung der Kursarbeiten

Der Termin der Klausur in der Sekundarstufe II wird rechtzeitig bekannt gegeben. In der Sekundarstufe II ist der Verweis auf gültige Klausurpläne ausreichend. Die Klausuren in der Sekundarstufe II, insbesondere in der Qualifikationsphase, sollen sich an den Aufgaben im Abitur orientieren.

2.d Aufgabenstellung und Aufgabenauswahl

Formulierungen und Aufgabenarten sollten den Schülerinnen und Schülern aus dem Unterricht weitestgehend bekannt sein. Die Schwierigkeit der Aufgaben sollte sich an folgender Aufteilung orientieren:

Anforderungsbereich I (Reproduktion)	ca. 35 bis 40 %
Anforderungsbereich II (Anwendung)	ca. 45 bis 50 %
Anforderungsbereich III (Transfer)	ca. 10 bis 15 %

2.e Korrektur und Bewertung

In den Klausuren der Schülerinnen und Schüler werden sowohl die Fehler als auch richtige Ergebnisse, Rechenschritte und Teilaspekte der Aufgaben deutlich gekennzeichnet. Die Fehlerkennzeichnung orientiert sich dabei an der üblichen Kennzeichnung im Sinne der Richtlinien und Lehrpläne. Bei Aufgaben, die am PC bearbeitet werden, reicht ein angemessener Hinweis auf den Fehler aus.

Die Teilaufgaben der Klausuren werden mit Punkten bewertet und die zu erreichende Punktzahl wird auf dem Aufgabenblatt angegeben.

Die Vergabe der Noten richtet sich nach der erreichten Gesamtpunktzahl, wobei sich diese an folgendem Notenschlüssel orientiert.

Sehr gut (plus)	95 % - 100 %	Ausreichend (plus)	50 % - 55 %
Sehr gut	90 % - 95 %	Ausreichend	45 % - 50 %
Sehr gut (minus)	85 % - 90%	Ausreichend (minus)	40 % - 45 %
Gut (plus)	80 % - 85%	Mangelhaft (plus)	33,3 % - 40 %
Gut	75 % - 80 %	Mangelhaft	26,7 % - 33,3 %
Gut (minus)	70 % - 75 %	Mangelhaft (minus)	20 % - 26,7 %
Befriedigend (plus)	65 % - 70 %	Ungenügend	0 % - 20%
Befriedigend	60 % - 65 %		
Befriedigend (minus)	55 % - 60 %		



2.f Nachbereitung

Die Klausur wird ausführlich besprochen und ggf. eine Musterlösung ausgegeben. Die Schülerinnen und Schüler erhalten ausreichend Gelegenheit zum Klären ihrer Fragen.

2.g Hilfsmittel

Hilfsmittel sind grundsätzlich keine erlaubt. Folgende Hilfsmittel kann die Lehrerin bzw. der Lehrer aber für die jeweilige Kursarbeit gestatten:

- Einsatz einer bestimmten Software (z.B. BlueJ) unter Verwendung des Computers im Klausurmodus,
- Taschenrechner,
- Befehlsübersichten zu verwendeten Sprachen (Java, SQL, UML usw.),
- Wörterbuch der deutschen Sprache bzw. Fremdwörterbuch.

Über den Einsatz anderer Hilfsmittel entscheidet die Fachkonferenz.



III.3 Formen und Bewertung der „Sonstigen Leistungen im Unterricht“

In der folgenden Übersicht werden die Bewertungskriterien der wichtigsten „Sonstigen Leistungen im Unterricht“ vorgestellt.

Form der „Sonstigen Leistung im Unterricht“	Bewertungskriterien
Beteiligung an Unterrichtsgesprächen	<ul style="list-style-type: none">✓ Qualität der Unterrichtsbeiträge✓ Quantität der Unterrichtsbeiträge✓ Verwendung der Fachsprache
Referate und Präsentationen	<ul style="list-style-type: none">✓ Verständlichkeit des Vortrags✓ Qualität des Vortrags✓ sinnvoller Einsatz von Medien✓ Arbeitsaufteilung bei Gruppenvorträgen✓ Qualität und Umfang des Handouts✓ Verwendung der Fachsprache
Arbeitsphasen (Einzel-, Partner-, Gruppenarbeit), Projektarbeit	<ul style="list-style-type: none">✓ Zuverlässigkeit und Zielstrebigkeit✓ Engagement und Selbständigkeit✓ Kooperation untereinander✓ Qualität des Resultats✓ Verwendung der Fachsprache
Heftführung (digital/Papierform), Portfolio und Projektdokumentationen, Hausaufgaben, Protokolle	<ul style="list-style-type: none">✓ Vollständigkeit✓ Qualität der Darstellung✓ Qualität der Inhalte✓ Pünktlichkeit (rechtzeitige Abgabe usw.)✓ Verwendung der Fachsprache

Eine Schwerpunktsetzung zur gezielten Förderung, z.B. bei Referaten, ist möglich und erwünscht. Diese sollte den Schülerinnen und Schülern rechtzeitig im Vorfeld mitgeteilt werden.



IV Anlagen

IV.1 Beispiel-Aufgaben für Kursarbeiten im Differenzierungsbereich

1.a Beispiel für eine Aufgabe im Differenzierungskurs 8

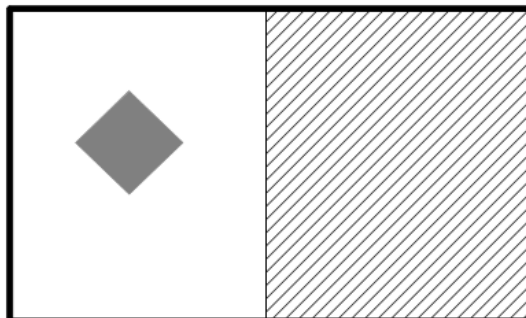
Die Datei bild.xcf besteht aus den drei abgebildeten Ebenen, wobei die Ebene 2 ganz oben angeordnet ist, darunter befindet sich Ebene 1 und ganz unten der Hintergrund.

Aus Gründen der Einfachheit werden nur drei „Farben“ unterschieden: **weiß**, **grau** und (vollständig) **transparent** (gestreift dargestellt).

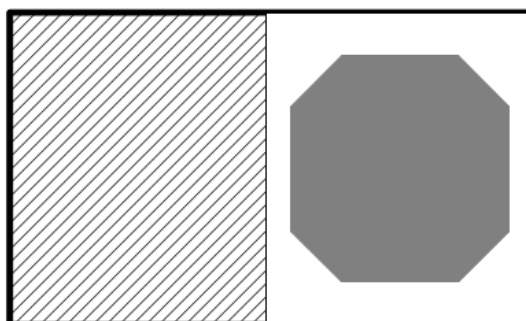
Hintergrund



Ebene 1



Ebene 2





- a) Skizziere das Bild so, wie es am Monitor dargestellt würde, wenn man alle drei Ebenen in dieser Anordnung anzeigen ließe. **[4 Punkte]**



- b) Skizziere das Bild so, wie es am Monitor dargestellt würde, wenn nur die Ebene 2 und der Hintergrund eingeschaltet wären und die Ebene 1 nicht. **[4 Punkte]**

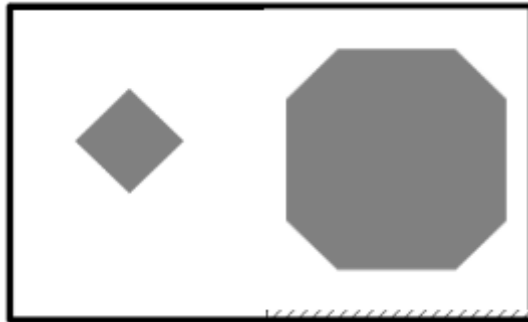


- c) Begründe, warum man das Bild im Dateiformat xcf speichern sollte. **[2 Punkte]**

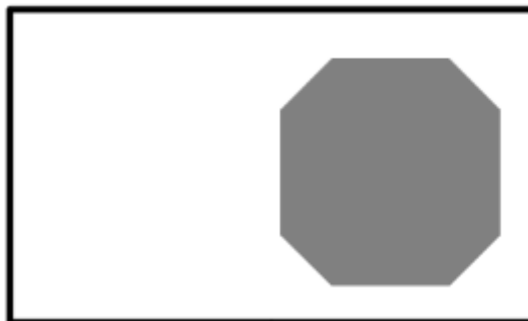


1.b Lösung zur Beispielaufgabe im Differenzierungskurs 8

- a) Skizziere das Bild so, wie es am Monitor dargestellt würde, wenn man alle drei Ebenen in dieser Anordnung anzeigen ließe. **[4 Punkte]**



- b) Skizziere das Bild so, wie es am Monitor dargestellt würde, wenn nur die Ebene 2 und der Hintergrund eingeschaltet wären und die Ebene 1 nicht. **[4 Punkte]**



- c) Begründe, warum man das Bild im Dateiformat xcf speichern sollte. **[2 Punkte]**

Das Dateiformat xcf ist das programmeigene Dateiformat von Gimp. Es speichert z.B. Ebenen verlustfrei, so dass man diese weiterhin bearbeiten kann.



1.c Beispiel für eine Aufgabe im Differenzierungskurs 9

Robot Karol hat den Auftrag bekommen, eine Lagerhalle leer zu räumen. D.h., er soll alle Ziegel, die sich in den einzelnen Kammern der Halle befinden einsammeln. Die Lagerhalle besteht aus vier Kammern, die allesamt nach Westen hin geöffnet sind (Siehe Abbildung!). Karol kann eine unbeschränkte Anzahl Ziegel zeitgleich transportieren.

- a) Skizzieren Sie die Welt, wie sie sich nach Bearbeitung der angegebenen Anweisungsfolge ergeben würde. [3 Punkte]

- b) Robot Karol muss sich während des Aufräumens mehrmals umdrehen. Aus diesem Grund soll eine eigene Anweisung `Umdrehen()` erstellt werden, die Karol bei Aufruf dazu veranlasst, sich um 180° zu drehen. Notieren Sie den vollständigen Code für die Definition dieser eigenen Anweisung. [2 Punkte]

- c) Die ersten drei Kammern haben denselben Grundriss. Daher reicht eine eigene Anweisung, damit Karol bei Aufruf dieses Befehls die entsprechende Kammer aufräumt. Der Programmcode würde sich dadurch folgendermaßen ändern:

```
Programm
  Schritt ();
  KammerLeeren ();
  Schritt (2);
  KammerLeeren ();
  Schritt (2);
  KammerLeeren ();
  Schritt (3);
// ...
*Programm
```

The screenshot shows a programming environment with a code editor on the left and a grid world on the right. The code editor contains the following code:

```
Programm
Schritt ();
LinksDrehen ();
Schritt (2);
Aufheben ();

// Umdrehen
LinksDrehen ();
LinksDrehen ();

Schritt (2);
LinksDrehen ();
Schritt (2);
LinksDrehen ();
Schritt ();
Aufheben ();
Schritt ();
Aufheben ();

// Umdrehen
LinksDrehen ();
LinksDrehen ();

Schritt (2);
LinksDrehen ();
Schritt (2);
LinksDrehen ();
Schritt ();
Aufheben (4);
Schritt ();
Aufheben (2);

//Umdrehen
LinksDrehen ();
LinksDrehen ();

Schritt (2);
LinksDrehen ();
Schritt (3);
// ...
*Programm
```

The grid world is a 10x10 grid. A black triangle at (1,1) indicates the robot's position and direction (West). Red numbers in the grid represent bricks: (2,4)=1, (3,3)=1, (3,4)=1, (4,3)=4, (4,4)=2, (5,3)=2, (6,4)=3. The grid is mostly black, representing walls or obstacles.

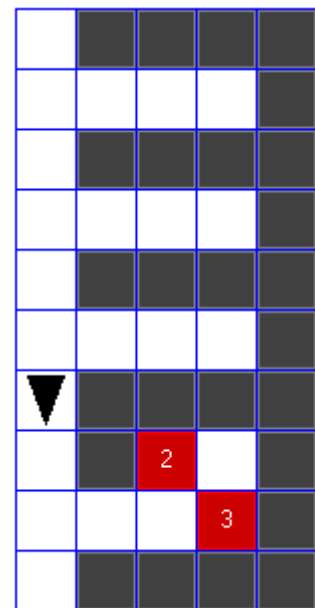
Notieren Sie den Quelltext für die eigene Anweisung `KammerLeeren()`. Sie dürfen dazu auch die Anweisung `Umdrehen()` verwenden. [5 Punkte]



- d) Schreiben Sie den Quelltext für die eigene Anweisung `LetzteKammerLeeren()`, die Karol dazu veranlasst, auch die letzte (unterste) Kammer zu leeren. Die Anweisung soll Karol dazu bringen, sämtliche Ziegel dieser Kammer aufzunehmen, wobei sich aber auf jedem der fünf Felder dieser Kammer beliebig viele Ziegel befinden können. Wenn Karol fertig ist, soll er auf dem Feld vor dieser Kammer stehen und nach Süden (unten) schauen. **[10 Punkte]**

1.d Lösung zur Beispielaufgabe im Differenzierungskurs 9

- a) Es ergibt sich diese Welt (siehe Abbildung!).



- b) Anweisung `Umdrehen()`
`LinksDrehen();`
`LinksDrehen();`
*Anweisung
- c) Anweisung `KammerLeeren()`
`LinksDrehen();`
`Schritt();`
`wiederhole solange IstZiegel()`
`Aufheben();`
*wiederhole
`Schritt();`
`wiederhole solange IstZiegel()`
`Aufheben();`
*wiederhole
`Umdrehen();`
`Schritt(2);`
`LinksDrehen();`
*Anweisung



d) Anweisung ZiegelAufheben()
wiederhole solange IstZiegel()
Aufheben();
*wiederhole
*Anweisung

Anweisung LetzteKammerLeeren()
LinksDrehen();
ZiegelAufheben();
Schritt();
ZiegelAufheben();
Schritt();
ZiegelAufheben();
Schritt();
LinksDrehen();
ZiegelAufheben();
Schritt();
LinksDrehen();
ZiegelAufheben();
Schritt();
LinksDrehen();
Schritt();
RechtsDrehen();
Schritt(2);
LinksDrehen();
*Anweisung

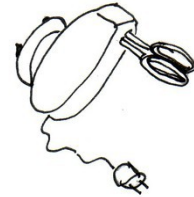


IV.2 Beispiel-Aufgaben für Klausuren im Grundkurs der Oberstufe

2.a Beispiel für eine Aufgabe in der Einführungsphase

Ein Handrührgerät soll über vier Geschwindigkeitsstufen verfügen:

- (0) Aus
- (1) Langsam (zum Kneten von Teig)
- (2) Normal (zum Verrühren)
- (3) Schnell (zum Schlagen von Sahne)



Das folgende Klassendiagramm zeigt, wie die Modellierung aussehen soll.



- a) Erläutern Sie anhand des Klassendiagramms, die Begriffe Klasse, Objekt, Attribut und Methode. **[8 Punkte]**
- b) Geben Sie zu zwei **verschiedenen** Objekten der Klasse Handruehrgeraet jeweils ein Objektdiagramm an. **[8 Punkte]**
- c) Gegeben ist der Quelltext des Konstruktors der Klasse Handruehrgeraet. Erläutern Sie, was ein Konstruktor ist, und erklären Sie den Quellcode in Zeile 6. **[4 Punkte]**

```
4 public Handruehrgeraet()  
5 {  
6     geschwindigkeitsstufe = 0;  
7 }
```

- d) Implementieren Sie die Methode `geschwindigkeitErhoehenUm1()` so, dass beim Aufruf der Methode die Geschwindigkeit um eins erhöht wird, sofern sich das Handrührgerät nicht bereits in der höchsten Stufe (schnell) befindet. **[5 Punkte]**



2.b Lösung zur Beispielaufgabe der Einführungsphase

- a) Erläutern Sie anhand des Klassendiagramms, die Begriffe Klasse, Objekt, Attribut und Methode. **[8 Punkte]**

Eine Klasse ist eine Art Bauplan für Objekte. Ein Java-Programm besteht aus lauter Klassen, in denen die Eigenschaften und Fähigkeiten der Objekte festgelegt werden.

Objekte sind konkrete Instanzen einer Klasse und haben Attributwerte. Sie „leben“ nur während das Programm abläuft.

Ein Attribut ist eine Eigenschaft eines Objekts. Alle Attribute zusammen legen den Zustand des Objekts fest. Die Objekte der Klasse Handruehrgeraete haben jeweils ein Attribut geschwindigkeitsstufe, welches die Werte 0, 1, 2 und 3 annehmen kann.

Methoden beschreiben die Fähigkeiten der Objekte. Die Objekte der Klasse Handruehrgeraete haben die Methoden geschwindigkeitErhoehenUm1(), geschwindigkeitVerringernUm1() und gibGeschwindigkeit().

- b) Geben Sie zu zwei **verschiedenen** Objekten der Klasse Handruehrgeraet jeweils ein Objektdiagramm an. **[8 Punkte]**

h1: Handruehrgeraet

- geschwindigkeitsstufe: 1

+ geschwindigkeitErhoehenUm1(): void
+ geschwindigkeitVerringernUm1(): void
+ gibGeschwindigkeit(): int

h2: Handruehrgeraet

- geschwindigkeitsstufe: 2

+ geschwindigkeitErhoehenUm1(): void
+ geschwindigkeitVerringernUm1(): void
+ gibGeschwindigkeit(): int



- c) Gegeben ist der Quelltext des Konstruktors der Klasse `Handruehrgeraet`. Erläutern Sie, was ein Konstruktor ist, und erklären Sie den Quellcode in Zeile 6. **[4 Punkte]**

```
4 public Handruehrgeraet()  
5 {  
6     geschwindigkeitsstufe = 0;  
7 }
```

Ein Konstruktor ist eine spezielle Methode, mit deren Hilfe man ein Objekt der Klasse erzeugen kann. Sie legt den Ausgangszustand des Objekts fest. Ihr Bezeichner entspricht dem der Klasse.

In Zeile 6 wird dem Attribut `geschwindigkeitsstufe` der Wert 0 zugewiesen.

- d) Implementieren Sie die Methode `geschwindigkeitErhoehenUml()` so, dass beim Aufruf der Methode die Geschwindigkeit um eins erhöht wird, sofern sich das Handrührgerät nicht bereits in der höchsten Stufe (schnell) befindet. **[5 Punkte]**

```
public void geschwindigkeitErhoehenUml() {  
    if ( geschwindigkeit < 3 )  
        geschwindigkeit += 1;  
}
```



2.c Beispiele für Aufgaben in der Qualifikationsphase

Eine Übersetzungsfirma hat sich auf die Übersetzungen von technischen Dokumentationen spezialisiert. Zur Verwaltung von Mitarbeitern, Aufträgen, Übersetzungen usw. wird ein relationales Datenbanksystem eingesetzt.

- a) In der Datenbank Mitarbeiter werden die Übersetzer nach folgender Anforderungsbeschreibung verwaltet.

Ein Übersetzer hat einen Vor- und einen Nachnamen, sowie eine eindeutige Personalnummer. Zu jedem Übersetzer werden die Kontodaten (Kontonr und Bankleitzahl) gespeichert, sowie dessen Email-Adresse.

Jeder Übersetzer hat genau eine Muttersprache und eine oder mehrere Ausgangssprachen (Fremdsprachen). (Er übersetzt Projekte ausschließlich von einer seiner Ausgangssprachen in seine Muttersprache.)

Außerdem ist jeder Übersetzer Experte in mindestens einem Fachgebiet wie beispielsweise Technik, Medizin oder Recht.

Erstellen Sie ein Entity-Relationship-Diagramm (ER-Diagramm) nach oben aufgeführter Anforderungsbeschreibung inklusive aller Attribute und Kardinalitäten. **[17 Punkte]**

- b) Dokumente werden mit Hilfe von Translation Tools (Übersetzungsprogramme) übersetzt. Dabei wird zu jedem Kunden eine Übersetzungsdatenbank für jedes Sprachenpaar aus Ausgangs- und Zielsprache angelegt, in dem die bereits erledigten Übersetzungen zur schnelleren Bearbeitung späterer Projekte gespeichert werden. Für die Firma ABC wurde beispielsweise folgende Datenbank **Uebersetzung** angelegt, die nur aus der abgebildeten Tabelle **Uebersetzung** besteht. Die Trefferwahrscheinlichkeit gibt dabei an, wie oft diese Übersetzung anteilig verwendet wurde.

Uebersetzung

ID	Ausgangssatz	Zielsatz	Trefferwahrscheinlichkeit
1	Please open the door.	Bitte öffnen Sie die Tür.	90
2	Press the button.	Drücken Sie den Knopf.	75
3	Press the button.	Taste drücken.	15
4	Start the engine.	Motor starten.	70
5	See above.	Siehe oben.	100
6	Press the button.	Drücke Knopf.	10



Notieren Sie die Ergebnistabellen für folgende Abfragen. [20 Punkte]

- (1) SELECT * FROM Uebersetzung;
- (2) SELECT ID, Ausgangssatz FROM Uebersetzung;
- (3) SELECT * FROM Uebersetzung WHERE Trefferwahrscheinlichkeit > 70;
- (4) SELECT Zielsatz, Trefferwahrscheinlichkeit FROM Uebersetzung WHERE Ausgangssatz = 'Press the button.';
- (5) SELECT ID, Zielsatz, Trefferwahrscheinlichkeit FROM Uebersetzung WHERE Ausgangssatz = 'Press the button.' AND Trefferwahrscheinlichkeit > 50;

- c) Gegeben ist das stark vereinfachte Relationenschema zur Datenbank **Auftraege**, in der die Aufträge und Kunden verwaltet werden.

Auftrag (Auftragsnr, Ausgangsdokument, Zeilenanzahl)

Kunde (Kundennr, Firmenname, Ansprechpartner)

Zutat (ID, Auftragsnr, Kundennr, PreisProZeile)

Notieren Sie die Abfragen in der Sprache SQL, die für folgende Ergebnisse die entsprechende Ergebnistabelle liefern. [18 Punkte]

- (1) Die komplette Tabelle Auftrag soll ausgegeben werden.
- (2) Die Auftragsnummern der Aufträge sollen aufgeführt werden, bei denen die Zeilenanzahl nicht größer als 20 ist.
- (3) Der Ansprechpartner der Firma, deren Kundennummer 21 ist, soll angezeigt werden.
- (4) Die Auftragsnummern der Aufträge, die von der Firma „Unity“ stammen, sollen zusammen mit der Zeilenanzahl aufgelistet werden.
- (5) Die Auftragsnummern der Aufträge, deren Preis pro Zeile 7 (in Cent) nicht übersteigt, sollen dargestellt werden zusammen mit den zugehörigen Firmennamen und Ansprechpartnern.



2.d Lösung zur Beispielaufgabe der Qualifikationsphase

Eine Übersetzungsfirma hat sich auf die Übersetzungen von technischen Dokumentationen spezialisiert. Zur Verwaltung von Mitarbeitern, Aufträgen, Übersetzungen usw. wird ein relationales Datenbanksystem eingesetzt.

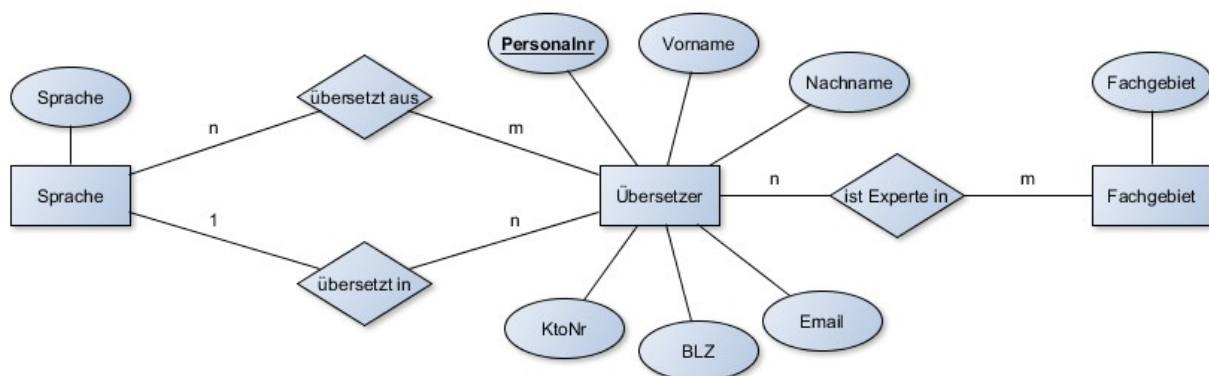
- a) In der Datenbank Mitarbeiter werden die Übersetzer nach folgender Anforderungsbeschreibung verwaltet.

Ein Übersetzer hat einen Vor- und einen Nachnamen, sowie eine eindeutige Personalnummer. Zu jedem Übersetzer werden die Kontodaten (Ktonr und Bankleitzahl) gespeichert, sowie dessen Email-Adresse.

Jeder Übersetzer hat genau eine Muttersprache und eine oder mehrere Ausgangssprachen (Fremdsprachen). (Er übersetzt Projekte ausschließlich von einer seiner Ausgangssprachen in seine Muttersprache.)

Außerdem ist jeder Übersetzer Experte in mindestens einem Fachgebiet wie beispielsweise Technik, Medizin oder Recht.

Erstellen Sie ein Entity-Relationship-Diagramm (ER-Diagramm) nach oben aufgeführter Anforderungsbeschreibung inklusive aller Attribute und Kardinalitäten. **[17 Punkte]**





b) Notieren Sie die Ergebnistabellen für folgende Abfragen. [20 Punkte]

Übersetzung

ID	Ausgangssatz	Zielsatz	Trefferwahrscheinlichkeit
1	Please open the door.	Bitte öffnen Sie die Tür.	90
2	Press the button.	Drücken Sie den Knopf.	75
3	Press the button.	Taste drücken.	15
4	Start the engine.	Motor starten.	70
5	See above.	Siehe oben.	100
6	Press the button.	Drücke Knopf.	10

(1) `SELECT * FROM Uebersetzung;`
Siehe *Uebersetzung;*
Ausgangstabelle!

(2) `SELECT ID, Ausgangssatz FROM Uebersetzung;`

ID	Ausgangssatz
1	Please open the door.
2	Press the button.
3	Press the button.
4	Start the engine.
5	See above.
6	Press the button.

(3) `SELECT * FROM Uebersetzung WHERE Trefferwahrscheinlichkeit > 70;`

ID	Ausgangssatz	Zielsatz	Trefferwahrscheinlichkeit
1	Please open the door.	Bitte öffnen Sie die Tür.	90
2	Press the button.	Drücken Sie den Knopf.	75
5	See above.	Siehe oben.	100



(4) `SELECT Zielsatz, Trefferwahrscheinlichkeit FROM Uebersetzung WHERE Ausgangssatz = 'Press the button.';`

Zielsatz	Trefferwahrscheinlichkeit
Drücken Sie den Knopf.	75
Taste drücken.	15
Drücke Knopf.	10

(5) `SELECT ID, Zielsatz, Trefferwahrscheinlichkeit FROM Uebersetzung WHERE Ausgangssatz = 'Press the button.' AND Trefferwahrscheinlichkeit > 50;`

<u>ID</u>	Zielsatz	Trefferwahrscheinlichkeit
2	Drücken Sie den Knopf.	75

- c) Gegeben ist das stark vereinfachte Relationenschema zur Datenbank **Auftraege**, in der die Aufträge und Kunden verwaltet werden.

Auftrag (Auftragsnr, Ausgangsdokument, Zeilenanzahl)

Kunde (Kundennr, Firmenname, Ansprechpartner)

zutat (ID, \hat{A} Auftragsnr, \hat{A} Kundennr, PreisProZeile)

Notieren Sie die Abfragen in der Sprache SQL, die für folgende Ergebnisse die entsprechende Ergebnistabelle liefern. [18 Punkte]

- (1) Die komplette Tabelle *Auftrag* soll ausgegeben werden.

`SELECT * FROM Auftrag;`

- (2) Die Auftragsnummern der Aufträge sollen aufgeführt werden, bei denen die Zeilenanzahl nicht größer als 20 ist.

`SELECT Auftragsnr FROM Auftrag WHERE NOT (Zeilenanzahl > 20);`
(*)

- (3) Der Ansprechpartner der Firma, deren Kundennummer 21 ist, soll angezeigt werden.

`SELECT Ansprechpartner FROM Kunde WHERE Kundennr = 21; (*)`

- (4) Die Auftragsnummern der Aufträge, die von der Firma „Unity“ stammen, sollen zusammen mit der Zeilenanzahl aufgelistet werden.

`SELECT a.Auftragsnr, a.Zeilenanzahl
FROM Auftrag a, Kunde k, Zutat z
WHERE a.Auftragsnr = z.Auftragsnr AND z.Kundennr = k.Kundennr
AND k.Firmenname = 'Unity'; (*)`



(5) Die Auftragsnummern der Aufträge, deren Preis pro Zeile 7 (in Cent) nicht übersteigt, sollen dargestellt werden zusammen mit den zugehörigen Firmennamen und Ansprechpartnern.

```
SELECT a.Auftragsnr, k.Firmenname, k.Ansprechpartner  
FROM Auftrag a, Kunden k, Zutat z  
WHERE a.Auftragsnr = z.Auftragsnr AND z.Kundennr = k.Kundennr  
AND NOT (z.PreisProZeile > 7); (*)
```

*) Es werden auch andere Lösungen akzeptiert, sie müssen nur das gleiche Resultat liefern.