

## Leistungsbewertung im Informatik-Unterricht

Die Leistungsbewertung im Informatikunterricht richtet sich nach den präzisen Vorgaben in den aktuellen Richtlinien und Lehrplänen. Die folgenden Angaben ergänzen diese Vorgaben.

### 1. MÜNDLICHE MITARBEIT IM INFORMATIKUNTERRICHT

In allen Jahrgangsstufen nimmt die Bewertung der Einzel- und Partnerarbeit am Computer einen großen Stellenwert ein (ca. 30-40% der Note außerhalb der Kursarbeiten und Klausuren). Im Laufe praktisch jeder Unterrichtsreihe werden zur Lernerfolgskontrolle Projekte durchgeführt, die bewertet werden.

In der gesamten Sekundarstufe 2 werden immer wieder Übungen für das mündliche Abitur durchgeführt, bei denen bewertete Prüfungssimulationen mit den Schülern geübt werden. Diese Übungen werden in der Regel angekündigt. Ihre Bewertung geht in die Note für die sonstige Mitarbeit ein.

### 2. KURSARBEITEN UND KLAUSUREN

Im Differenzierungsbereich der Jahrgangsstufen 8 und 9 werden im ersten Schulhalbjahr zwei 45- bis 60-minütige Kursarbeiten geschrieben. Im zweiten Schulhalbjahr wird eine Kursarbeit (45 bis 60 Min.) geschrieben und die zweite Kursarbeit durch ein mehrwöchiges Projekt ersetzt.

In den Oberstufen-Kursen werden die Klausuren gemäß den aktuellen Richtlinien und den Beschlüssen der Lehrerkonferenz geschrieben. Im Schuljahr 2011/2012 bedeutet dies: In der Einführungsphase pro Kurs eine 90-minütige Klausur. Im ersten Jahr der Qualifizierungsphase je zwei 90-minütige (Grundkurs), bzw. 135-minütige (Leistungskurs) Klausuren pro Kurs. Im ersten Halbjahr des zweiten Jahres der Qualifizierungsphase je zwei 135-minütige (Grundkurs) bzw. 180-minütige (Leistungskurs) Klausuren pro Kurs. Im zweiten Halbjahr des zweiten Jahres der Qualifizierungsphase eine Klausur unter Abitur-Bedingungen.

Bei der Bewertung der Kursarbeiten und Klausuren sind die Punkteschemata des Zentralabiturs nicht verbindlich. Die Typen der im Zentralabitur vorkommenden Aufgaben werden jedoch schon ab dem Beginn der Qualifizierungsphase geübt und in den Klausuren verwendet.

Details zu den Themen der Kursarbeiten und Projekte finden sich im schulinternen Curriculum.

### 3. BEISPIELE FÜR KURSARBEITEN UND KLAUSUREN IM FACH INFORMATIK

Zur Dokumentation des Anforderungsniveaus der Kursarbeiten sind als Anhang zu dieser Dokumentation zwei Arbeiten und die dazugehörigen Bewertungsschemata hinzugefügt:

- Eine Arbeit der Jahrgangsstufe 8 zu HTML und den Anfängen der Codierung mit Dualzahlen,
- eine Arbeit zum Thema "Datenkompression" aus der Jahrgangsstufe 12.

Bei der Arbeit der Jahrgangsstufe 8 gehören die Aufgaben 1 und 5 zum Anforderungsniveau 1, die Aufgaben 2 und 4 mehrheitlich zum Anforderungsniveau 2 und die Aufgabe 3 mehrheitlich zum Anforderungsniveau 3.

Bei der Klausur der Jahrgangsstufe 12 handelt es sich bei Aufgabe 1 um eine modifizierte Aufgabe des Zentralabiturs, die allerdings, da sie für die Klausur vereinfacht wurde, etwas gleichgewichtig die Anforderungsniveaus 1 und 2 abdeckt. Aufgabe 2 befindet sich etwa gleichgewichtig auf den Anforderungsniveaus 2 und 3.

### 4. FACHARBEITEN

Im Fach Informatik kann in der Jahrgangsstufe 12 eine Facharbeit geschrieben werden, die dann eine der Klausuren ersetzt. Im Anhang findet sich eine Thema und eine Beurteilung.

## Beispiel für eine Kursarbeit der Jahrgangsstufe 8

**Bearbeitungszeit:** 90 Minuten  
**Hilfsmittel:** Eine handgeschriebene DIN-A-4-Seite

**Hinweise:**

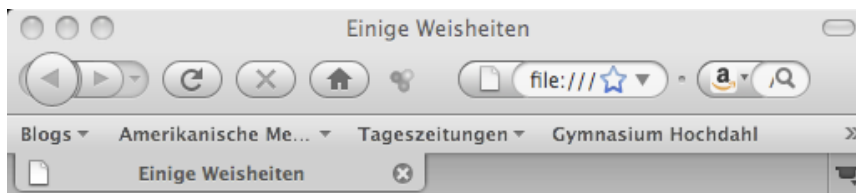
- a) Nur XHTML-kompatibler Code ist erlaubt. (Das betrifft nur diejenigen, die vorher schon HTML konnten. Alle anderen haben sowieso nur XHTML gelernt...)
- b) Für das "Vergessen" von sinnvollen Einrückungen im HTML-Quelltext gibt es Punktabzüge! Nicht eingrücktes HTML gilt als "falsch"!

### AUFGABE 1

- a) Erläutere mit Text und Beispielen: Was ist der Unterschied zwischen einem "Tag" und einem "Element"?
- b) Worin unterscheidet sich das Style-Attribut von praktisch allen anderen Attributen in HTML? (Mindestens die 3 wichtigsten Unterschiede nennen!)
- c) Wie kann ich in einem angezeigten HTML-Dokument (nicht im Quelltext) Leerraum, also unbeschriebenen Platz erzeugen? Was passiert demgegenüber mit Leerraum, der im Quelltext vorhanden ist?

### AUFGABE 2

Notiere den HTML-Code, der notwendig ist, um folgende Webseite zu erzeugen!



## Einige Programmierweisheiten

| Weisheit  |              |
|---|--------------|
| Neue Systeme erzeugen neue Probleme.                      | stimmt       |
| Wenn <b>alles andere</b> fehlschlägt: Schau ins Handbuch! | wie zynisch! |

der Rahmen ist grün gepunktet

Fertig

### AUFGABE 3

Das folgende HTML-Dokument enthält mindestens 6 Fehler. Berichtige die Fehler, indem du eine korrekte Version des Dokuments ins Heft schreibst und die Fehler erläuterst!

```
<htm><head>Spruch<body>Ein schöner Tag, die Welt steht still: Ein schöner Tag...<htm><body>
```



**BEWERTUNGSSCHEMA ZUR "HTML"-KURSARBEIT:**

|    | DIFF IF 8  |  | Max.      |
|----|--|--|-----------|
|    | Kursarbeit Nr. 1   |  |           |
| 1a | Element = Textformatierung; Attribut verändert Standard-Aussehen   |  | <b>3</b>  |
|    | Beispiel   |  | <b>1</b>  |
| b  | Kann überall vorkommen; kann mehrere Werte enthalten; kann ganz unterschiedliche Dinge bewirken                              |  | <b>3</b>  |
| c  | Leerraum erzeugen z.B. mit <br>, <p> oder Span-Attributen  |  | <b>2</b>  |
|    | Leerraum im Quelltext wird "zusammengefasst" => ist nicht sichtbar und kann für Formatierung des Quelltextes benutzt werden. |  | <b>1</b>  |
|    |  |  | <b>10</b> |
| 3  | <htm> -> <html>  |  | <b>1</b>  |
|    | <title>Spruch</title>  |  | <b>1</b>  |
|    | </head>  |  | <b>1</b>  |
|    | ">" bei body   |  | <b>1</b>  |
|    | <p> oder <div> um Text einzuschließen  |  | <b>1</b>  |
|    | Anführungszeichen beim style-Attribut  |  | <b>1</b>  |
|    | "/" beim öffnenden i muss weg  |  | <b>1</b>  |
|    | Bei schließendem Tag: <body> -> </body>; analog bei <html> und <i>   |  | <b>1</b>  |
|    | Reihenfolge: </i></body></html>  |  | <b>1</b>  |
|    | Sonderpunkt für text-color -> color (Fehler im Aufgabentext)   |  | <b>1</b>  |
|    | Einrückung   |  | <b>1</b>  |
|    |  |  | <b>10</b> |
| 2  | Gerüst (mit HTML, Head und Body)   |  | <b>1</b>  |
|    | Titel oben im Browserfenster als Element im Head   |  | <b>1</b>  |
|    | css-Datei mit Angaben für den Rahmen in der Überschrift (style und farbe)  |  | <b>2</b>  |
|    | Überschrift in h-Element oder korrekt formatiert   |  | <b>1</b>  |
|    | Table-Element mit Border-Attribut  |  | <b>1</b>  |
|    | TR-Elemente korrekt angewendet   |  | <b>1</b>  |
|    | TD-Elemente korrekt angewendet   |  | <b>1</b>  |
|    | "Weisheit" zentriert und fett  |  | <b>1</b>  |
|    | Tooltip auf dem leeren Feld  |  | <b>1</b>  |
|    | Letzte Zeile: Fett und Unterstrichen richtig angewendet  |  | <b>1</b>  |
|    | korrekte Verschachtelung   |  | <b>1</b>  |
|    |  |  | <b>12</b> |
| 4  | Titel im Browserfenster oben   |  | <b>1</b>  |
|    | Überschrift zentriert und winzig   |  | <b>2</b>  |
|    | p mit Verfasser und folgender Zeile rechts   |  | <b>1</b>  |
|    | Verfasser fett   |  | <b>1</b>  |
|    | Zeilenumbruch korrekt ("unbekannt, frei nach Murphy" in einer Zeile)   |  | <b>1</b>  |
|    | Liste mit Punkten (unordered), links-orientiert  |  | <b>1</b>  |
|    | Textfarbe in Liste: schwarz  |  | <b>1</b>  |
|    | Zeilenumbrüche in Liste korrekt (incl.: Text mindestens so weit notiert, dass er einen Zeilenumbruch enthält)                |  | <b>1</b>  |

|    |   |           |
|----|---|-----------|
|    | "Fehler", "Aufmerksamkeitsspanne", "irren" kursiv   | 1         |
|    | "Fehler" mit blauem Hintergrund   | 1         |
|    | "Aufmerksamkeitsspanne" roter Hintergrund   | 1         |
|    | "Irren" mit grünem Hintergrund  | 1         |
|    | Link unterstrichen dargestellt  | 1         |
|    | Gesamter letzter Punkt 150% Schriftgröße  | 1         |
|    | Sonstige Formatierung korrekt (z.B. keine "falschen Fehler", Hintergründe und Schrift einheitlich...) | 2         |
|    |   | <b>17</b> |
| 5a | Großbuchstaben  | 2         |
|    | Kleinbuchstaben   | 2         |
|    | Satzzeichen   | 1         |
|    | Leerzeichen und Zeilenumbruch   | 1         |
| b  | i) 10.011   | 1         |
|    | ii) 11.110  | 1,5       |
|    | iii) 10.010.110   | 1,5       |
|    |   | <b>10</b> |
|    |   | <b>59</b> |

Note: Minimale Punktzahl:

|    |      |
|----|------|
| 1+ | 51,5 |
| 1  | 49,5 |
| 1- | 47   |
| 2+ | 44,5 |
| 2  | 42,5 |
| 2- | 40   |
| 3+ | 37,5 |
| 3  | 35,5 |
| 3- | 33   |
| 4+ | 30,5 |
| 4  | 28,5 |
| 4- | 26   |
| 5+ | 23,5 |
| 5  | 21,5 |
| 5- | 19   |
| 6  | 0    |

## Beispiel für eine Klausur der Qualifizierungsphase (1. Jahr)

### AUFGABE 1

Bei Warteschlangen kommt es häufig vor, dass einige Elemente aus berechtigten Gründen nicht am Ende der Schlange eingefügt werden sollen, sondern an einem weiter vorne gelegenen Platz eingereiht werden. Dies sind zum Beispiel bestimmte Druckaufträge in einer Druckerwarteschlange oder Patienten mit verschiedenen Dringlichkeiten in einer Arztpraxis.

Zur Realisierung einer solchen Schlange werden die einzufügenden Objekte mit einer ganzzahligen Priorität versehen. Je höher die Priorität, desto weiter nach vorne gelangen die Objekte innerhalb der Schlange. Besitzen mehrere Objekte dieselbe Priorität, so soll das neue Objekt hinter die bereits enthaltenen Objekte gleicher Priorität eingereiht werden. Aus diesem Grund werden derartige Schlangen auch als Prioritätenwarteschlange (Priority Queue) bezeichnet.

#### A)

In einer Arztpraxis mit anfangs leerem Wartezimmer spielt sich folgendes Szenario ab:

- Herr Arendt erhält die Priorität 4 und betritt das Wartezimmer.
- Frau Wolff erhält die Priorität 2 und betritt das Wartezimmer.
- Frau Fritz erhält die Priorität 5 und betritt das Wartezimmer.
- Der erste Patient wird aufgerufen und verlässt das Wartezimmer.
- Herr Kluge erhält die Priorität 4 und betritt das Wartezimmer.

Stellen Sie die Belegung der Warteschlange nach jedem Einfügen eines Patienten geeignet dar.

#### B)

Basierend auf der bekannten Klasse `Queue` soll die Klasse `PriorityQueue` durch eine Ist-Beziehung oder durch eine Hat-Beziehung realisiert werden. (*Okay, beides ist nicht unbedingt sinnvoll. Aber es soll nun einmal eines von beiden sein!*) Beschreiben Sie die Vor- und Nachteile der beiden Möglichkeiten! Welche Möglichkeit ist besser?

#### C)

Objekte, die in eine Prioritätenwarteschlange eingefügt werden sollen, sollen vom Typ `PriorityObject` sein, d. h., sie sollen stets einer Klasse angehören, die Unterklasse der Klasse `PriorityObject` ist. Für den Fall des Wartezimmers könnte dies etwa eine Klasse `Patient` sein.

Entwerfen Sie ein UML-Klassendiagramm für die Klassen `PriorityQueue`, `PriorityObject` und `Patient`. Geben Sie dabei alle *notwendigen* Methoden und Beziehungen an. (*Überlegen Sie sich also: Wie sollen die Klassen miteinander kommunizieren, wie sollen Objekte der `PriorityQueue` hinzugefügt und aus ihr entnommen werden?*) Erläutern Sie kurz die Wirkung aller in `PriorityQueue` und `PriorityObject` enthaltenen Methoden mit Ausnahme der Konstruktoren.

#### D)

In der folgenden Aufgabe 'haben' Objekte der Klasse `PriorityQueue` eine Schlange `s` vom Typ `Queue`, in der sie die Objekte der Klasse `PriorityObject` verwalten.

In der Klasse `PriorityQueue` soll nun eine Methode `add(PriorityObject pObj)` zur Aufnahme eines Objektes der Klasse `PriorityObject` in die Prioritätenwarteschlange zur Verfügung stehen. Die Methode soll das einzufügende Objekt gemäß seiner Priorität an der entsprechenden Position in die Warteschlange einfügen.

Erläutern Sie die Vorgehensweise einer solchen Methode und implementieren Sie diese.

## AUFGABE 2

Im Unterricht haben wir Lauflängen-Codierung nur mit Daten durchgeführt, die ausschließlich aus einer Folge von '0' und '1' bestanden. In der folgenden Aufgabe soll es darum gehen, *Texte* mit Hilfe der Lauflängen-Codierung zu komprimieren.

A)

Warum ist es problematisch, einen Text mit Hilfe der Lauflängen-Codierung zu komprimieren? Begründen Sie!

B)

Modifizieren Sie den im Unterricht behandelten Lauflängen-Codierer so, dass eine Lauflängen-Codierung von Texten möglich wird!

- i. Beschreiben Sie in Textform, wie Sie bei der Lauflängen-Codierung von Texten vorgehen!
- ii. Geben Sie mit Hilfe eines Klassendiagramms alle notwendigen Methoden des neuen Lauflängen-Codierers mit Ein- und Ausgabeparametern an! (Sie dürfen dabei den im Unterricht behandelten Lauflängen-Codierer voraussetzen, müssen aber nicht.)
- iii. Beschreiben Sie in Textform möglichst präzise, wie die Methoden den neuen Lauflängen-Codierers arbeiten! (Nur Methoden beschreiben, die sich vom "alten" Lauflängen-Codierer unterscheiden!)

**BEWERTUNGSSCHEMA ZUR KLAUSUR:**

 IF 12-Klausur Queue und Datenkompression  
 Version 2009-06-05

|                  |                              |               | Nötig für 1  | Nötig für 4  |
|------------------|------------------------------|---------------|--------------|--------------|
| <b>Aufgabe 1</b> |                              |               |              |              |
| <b>a</b>         | 5 richtige Zustände          | 5             | 90%          | 75%          |
| <b>b</b>         | 4 pro/contra Argumente       | 4             | 50%          | 25%          |
| <b>c</b>         | Diagramm PriorityQueue       | 3             | 90%          | 50%          |
|                  | Diagramm PriorityObject      | 2             | 90%          | 50%          |
|                  | Diagramm Patient             | 1             | 90%          | 50%          |
| <b>d</b>         | Beziehungen                  | 2             | 75%          | 50%          |
|                  | Beschreibung                 | 2             | 75%          | 50%          |
|                  | Implementierung              | 3             | 75%          | 50%          |
|                  |                              | <b>22</b>     | <b>17,15</b> | <b>11,25</b> |
| <b>Aufgabe 2</b> |                              |               |              |              |
| <b>a</b>         | 1 gute Begründung            | 2             | 75%          | 25%          |
| <b>b.i</b>       | Umwandlung in 0/1            | 2             | 75%          | 25%          |
| <b>b.ii</b>      | Diagramm mit Ist-Bez.+1 Meth | 4             | 75%          | 25%          |
| <b>b.iii</b>     | Pseudocode                   | 4             | 50%          | 0%           |
|                  |                              | <b>12</b>     | <b>8</b>     | <b>2</b>     |
|                  |                              | 34            | 25,15        | 13,25        |
|                  |                              | <b>100,0%</b> | <b>74,0%</b> | <b>39,0%</b> |

| <b>Noten-Berechnung</b> |                         | Min.  |
|-------------------------|-------------------------|-------|
|                         | Kennzahl "sehr gut -":  | 25,15 |
|                         | Kennzahl "ausreichend": | 13,25 |
| <b>15</b>               | sehr gut +              | 28,13 |
| <b>14</b>               | sehr gut                | 26,64 |
| <b>13</b>               | sehr gut -              | 25,15 |
| <b>12</b>               | gut +                   | 23,66 |
| <b>11</b>               | gut                     | 22,18 |
| <b>10</b>               | gut -                   | 20,69 |
| <b>9</b>                | befriedigend +          | 19,2  |
| <b>8</b>                | befriedigend            | 17,71 |
| <b>7</b>                | befriedigend -          | 16,23 |
| <b>6</b>                | ausreichend +           | 14,74 |
| <b>5</b>                | ausreichend             | 13,25 |
| <b>4</b>                | ausreichend -           | 11,76 |
| <b>3</b>                | mangelhaft +            | 10,28 |
| <b>2</b>                | mangelhaft              | 8,79  |
| <b>1</b>                | mangelhaft -            | 7,3   |
| <b>0</b>                | ungenügend              | 0     |

## Bewertung der Facharbeit von N.N.

(Das Thema der Facharbeit war: *Probleme bei der Kommunikation zwischen Erde und Mars-Robotern.*)

### INHALT:

#### Textteil (I. und II.):

Der erste, theoretische Teil der Arbeit ist nicht gut gelungen. Teil II.1. (Probleme) ist zu oberflächlich. Technische Details fehlen. Auch die Darstellung der Lösungsversuche ist oberflächlich und gibt als einzi-ge Quelle einen Spiegel-Artikel an.

Der praktische Versuch im zweiten Teil der Arbeit wird nicht vorbereitet. Es bleibt unklar, wieso das im praktischen Teil der Arbeit vorgestellte Verfahren gegen die im Textteil beschriebenen Probleme helfen soll.

(4-)

#### Praktischer Teil mit Beschreibung (III.):

Die drei verschiedenen Fehlerklassen (Änderung, Hinzufügen, Entfall von Zeichen), die bei der Kommunikation auftreten können, werden nicht dargestellt.

Die vorgeschlagene "Sicherstellung der Bereitschaft" bläht den Umfang der Kommunikation sehr stark auf, ohne dass damit sichergestellt wird, dass die Kommunikation tatsächlich besser gelingt.

Die textliche Beschreibung des eigentlichen Algorithmus ist gut gelungen und verständlich.

Die Analyse der Zuverlässigkeit ist leider falsch. Es können Fehler unentdeckt bleiben, am offensichtlichsten dann, wenn sowohl die Prüfsumme, als auch die eigentliche Botschaft einen Fehler enthalten.

Die Implementierung des Prüfsummen-Algorithmus ist offensichtlich aus einer Quelle übernommen worden. Dies ist an sich nicht ehrenrührig, im Gegenteil. Warum soll man für ein bereits gelöstes Problem erneut eine Lösung suchen? Allerdings muss das Übernehmen fremden Codes ausdrücklich angegeben werden. Es geht aus dem Quelltext nicht hervor, welche Teile des Programms eigene Leistung sind und welche übernommen wurden.

Es ist mir leider nicht gelungen, das Programm zum Laufen zu bringen. Eine Kommunikation zwischen "Erde" und "Mars" war nicht möglich.

(4-)

### FORM:

Die Arbeit entspricht weitgehend den formalen Vorgaben. Sie enthält recht viele Rechtschreib- und Grammatik-Fehler, ist aber immer noch sehr gut lesbar.

Die Quellen-Verwendung ist nicht gut gelungen. Häufig werden Fakten oder Behauptungen präsentiert, ohne Belegstellen dafür anzugeben (siehe Korrekturen in der Arbeit). Das Literaturverzeichnis ist zwar recht umfangreich. Die Literatur wird aber in der Arbeit nur selten wirklich verwendet.

(3-)

### GESAMTNOTE:

**ausreichend (4)**