

3. Klassenarbeit



Hinweis

- Falls in den Aufgaben nicht anders angegeben, formuliere stets ganze Sätze!
- Für Ausdruck, Rechtschreibung und Form gibt es zusätzlich 6 Punkte!
- Es sind **keine** Hilfsmittel erlaubt!
- Die Bearbeitungszeit beträgt **75 Minuten**.

Viel Erfolg!



Ergebnis

	durchgängig angemessen	im Allgemeinen angemessen	nicht angemessen	erreichte BE/ erreichbare BE
Ausdruck , inkl. Fachbe- griffe und fachliche Wen- dungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	___ / ___
Sprachliche Normen (Rechtschreibung, Gram- matik, Zeichensetzung)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	___ / ___
Äußere Form	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	___ / ___

Inhalt:	/54	Sprache/Form:	/6	Σ	/60	%
---------	-----	---------------	----	----------	-----	---

1	2	3	4	5	6

Note

Datum, Paraphe

Unterschrift

Teil 1: Objektorientierter Ansatz

Löse diesen Teil der Klassenarbeit schriftlich auf Papier.

Aufgabe 1

Nenne und erkläre zwei Vorteile der objektorientierten Programmierung (OOP) gegenüber dem imperativen Ansatz. _____/4 P.

Aufgabe 2

Erstelle ein Klassendiagramm für die in Material M1 gegebene Klasse. _____/8 P.

Aufgabe 3

Erkläre ausführlich die Funktionsweise der in Material M1 dargestellten Methode update(). _____/4 P.

Aufgabe 4

Erläutere die folgenden Begriffe mithilfe eines Beispiels aus Material M1 und/oder M2. _____/12 P.

a) Methode

c) Parameter

b) Konstruktor

d) Attribut

Aufgabe 5

Erkläre ausführlich den Unterschied zwischen einem Parameter und einem Argument. Wähle ein Beispiel aus dem Material. _____/5 P.

Teil 2: Spieleprogrammierung mit Processing

Löse diesen Teil der Klassenarbeit soweit nicht anders angegeben am Computer. Kopiere die im Tauschlaufwerk bereitgestellten Dateien in den Einsammeln-Ordner. Drucke am Ende den Quelltext aus und unterschreibe ihn.

Aufgabe 6

Erweitere das bestehende Programm so, dass magnetische Gegenstände von dem Magneten angezogen werden. Dafür soll ein Gegenstand, der sich auf der Höhe des Magneten befindet, angezogen werden und am Magneten haften bleiben. Gegenstände, die nicht magnetisch sind, sollen nicht beeinflusst werden und weiter geradlinig nach unten fallen. _____/6 P.

Aufgabe 7

Erweitere das bestehende Programm so, dass zeitgleich zum Hammer ein Diamant herabfällt. Erstelle dafür ein neues Objekt mit passender Grafik. _____/5 P.

Aufgabe 8

Zeichne eine Objektkarte (Objektdiagramm) des Diamant-Objektes aus Aufgabe 7 **auf Papier**. _____/6 P.

Aufgabe 9

Der Hammer und der Diamant überlappen sich. Erweitere den Konstruktor der Klasse Gegenstand so, dass beim Erstellen eines Objektes ein Argument für die x-Position mit übergeben werden muss. Ändere das Programm so ab, dass der Diamant rechts neben dem Hammer herab fällt. _____/4 P.

Material M1

```
1 class Gegenstand {
2
3     int xPos, yPos;
4     String bezeichnung;
5     boolean magnetisch;
6     PImage bild;
7
8     public Gegenstand(String img, String bezeichnung, boolean magnetisch)
9     {
10         xPos = 400;
11         yPos = -50;
12
13         bild = loadImage(img);
14         this.bezeichnung = bezeichnung;
15         this.magnetisch = magnetisch;
16     }
17
18     public void update()
19     {
20         if (yPos < 700) {
21             yPos = yPos + 3;
22         } else {
23             yPos = -100;
24         }
25     }
26
27     public void draw()
28     {
29         image(bild, xPos, yPos);
30     }
31 }
```

Material M2

```
1 PImage magnet;
2 Gegenstand g1;
3
4 void setup() {
5     size(800,600);
6     magnet = loadImage("magnet.png");
7     g1 = new Gegenstand("hammer.png","Hammer",true);
8 }
9
10 void draw() {
11     background(150,200,255);
12     image(magnet,0,200,200,191);
13
14     g1.update();
15     g1.draw();
16 }
```