

# Niveaubestimmende Aufgabe zum Fachlehrplan

## Informationstechnik

### Fachgymnasium

#### „Objektorientierter Würfel“

Schuljahrgang 12

Arbeitsstand: 28.04.2017

Niveaubestimmende Aufgaben sind Bestandteil des Lehrplankonzeptes für das Gymnasium und das Fachgymnasium. Die nachfolgende Aufgabe soll Grundlage unterrichtlicher Erprobung sein. Rückmeldungen, Hinweise, Anregungen und Vorschläge zur Weiterentwicklung der Aufgabe senden Sie bitte über die Eingabemaske (Bildungsserver) oder direkt an [andrea.neubauer@lisa.mb.sachsen-anhalt.de](mailto:andrea.neubauer@lisa.mb.sachsen-anhalt.de)

An der Erarbeitung der niveaubestimmenden Aufgabe haben mitgewirkt:

Dr. Preuschoff, Frank	Halle
Schneider, Jörg	Magdeburg
Schreiber, Sikor	Halle
Schulze, Holger	Halle (Leitung der Fachgruppe)

Herausgeber im Auftrag des Ministeriums für Bildung des Landes Sachsen-Anhalt:  
Landesinstitut für Schulqualität und Lehrerbildung Sachsen-Anhalt  
Riebeckplatz 09  
06110 Halle



Die vorliegende Publikation, mit Ausnahme der Quellen Dritter, ist unter der „Creative Commons“-Lizenz veröffentlicht.

 CC BY-SA 3.0 DE <http://creativecommons.org/licenses/by-sa/3.0/de/>

Sie dürfen das Material weiterverbreiten, bearbeiten, verändern und erweitern. Wenn Sie das Material oder Teile davon veröffentlichen, müssen Sie den Urheber nennen und kennzeichnen, welche Veränderungen Sie vorgenommen haben. Sie müssen das Material und Veränderungen unter den gleichen Lizenzbedingungen weitergeben.

Die Rechte für Fotos, Abbildungen und Zitate für Quellen Dritter bleiben bei den jeweiligen Rechteinhabern, diese Angaben können Sie den Quellen entnehmen. Der Herausgeber hat sich intensiv bemüht, alle Inhaber von Rechten zu benennen. Falls Sie uns weitere Urheber und Rechteinhaber benennen können, würden wir uns über Ihren Hinweis freuen.

# Objektorientierter Würfel

## Qualifikationsphase

Entwickeln Sie im Rahmen der objektorientierten Programmierung mit einer Programmiersprache eine eigene Klasse für einen universellen Würfel gemäß folgender Definition:

Jeder instanziierte Würfel soll mit gleichartigen Flächen vorstellbar sein. Dementsprechend soll er von der Zahl Eins beginnend bis zu einer jeweils maximalen Augenzahl zufallsbedingt würfeln können. Ausgangspunkt der Überlegungen soll der Standardwürfel (ein Hexaeder) sein.

Die Anzeige der gewürfelten Augenzahl erfolgt mit einer vom Programmierer ausgewählten Farbe. Eine Ausgabeform der Augen kann mit arabischen Zahlen oder der entsprechenden Anzahl eines festgelegten Symbols realisiert werden. Außerdem soll ein frei wählbarer Hilfetext anzeigbar sein.

Gehen Sie bei der Bearbeitung systematisch nach den Methoden des Softwareengineering vor.

1. Verschaffen Sie sich einen Überblick über die fünf platonischen Körper.
2. Ermitteln Sie alle notwendigen Eigenschaften und Methoden der zu erstellenden Klasse als Sollzustand und stellen Sie diese mit einem UML-Diagramm geeignet dar.
3. Implementieren Sie Ihren Entwurf mithilfe einer Programmiersprache und testen Sie diesen auf Korrektheit. Dokumentieren Sie die Testergebnisse.

### Material

- Nachschlagewerk
- Internet
- eigene Aufzeichnungen

## Einordnung in den Fachlehrplan

### Kompetenzschwerpunkt:

- Informationstechnische Systeme objektorientiert modellieren

### zu entwickelnde Schlüsselkompetenzen:

- vielfältige Informationen beschaffen und bewerten sowie ständig mit Informationssystemen umgehen
- das eigene Lernen selbstorganisieren und reflektieren
- mathematische Strukturen in Problemen erkennen und mathematische Darstellungen anwenden
- Sprache sach- und normgerecht einsetzen
- individuell und in Kooperation mit anderen lernen und handeln

### zu entwickelnde fachspezifische Kompetenzen:

- Modelle für Problemstellungen adäquat objektorientiert entwerfen
- Programme für naturwissenschaftlich-technische Problemstellungen in einer objektorientierten Programmiersprache entwickeln
- Modelle für Problemstellungen (Erzeugen von Klassen mit den zugehörigen Eigenschaften und Methoden) adäquat objektorientiert entwerfen
- Objekt- und Klassenmethoden (Instanzen, Zugriff auf die Klasselemente, entsprechende Testszenarien) für programmiertechnische Probleme evaluieren

### Bezug zu grundlegenden Wissensbeständen:

- Klassen- und Objektdiagramme in UML
- Begriffe: Klasse, Eigenschaft, Methode, Instanz, Objekt und Vererbung, Polymorphie
- objektorientierte Darstellung in einer Programmiersprache

## **Anregungen und Hinweise zum unterrichtlichen Einsatz**

Die Aufgabe sollte zum Ende des Kurshalbjahres 13/1 gestellt werden. Die Schülerinnen und Schüler kennen die objektorientierte Modellierung mit UML-Diagrammen bereits aus anderen Aufgabenstellungen. Ebenso sind sie mit den grundlegenden Begriffen, wie Klasse, Eigenschaft, Methode, Instanz, Objekt, Vererbung und Polymorphie, vertraut.

Sie planen und organisieren ihre Arbeit selbstständig in Gruppen und dokumentieren den Lernprozess sowie die Ergebnisse vollständig am Computer.

Die Arbeitszeit für die Lösung der niveaubestimmenden Aufgabe beträgt ca. vier Unterrichtsstunden.

## Variationsmöglichkeiten

Die niveaubestimmende Aufgabe kann durch gezielte Vorgabe von Teillösungen in Zeitumfang und Komplexität reduziert werden, beispielsweise der teilweisen Vorgabe von Eigenschaften oder ausgewählter Methoden. Im Rahmen einer Klausur oder als Bestandteil einer Vorprüfungsklausur ist die Aufgabe in gekürztem Umfang geeignet.

Der Programmieraufwand kann komplexer gestaltet werden, wenn mit einer erstellten Instanz eine konkrete Anwendung realisiert wird. Denkbar ist die Realisierung eines einfachen Würfelspiels. Durch diese anwendungsbezogene Erweiterung werden die Aufmerksamkeit und die Motivation der Schülerinnen und Schüler wesentlich erhöht. Ebenso sind in diesem Kontext mehrere Instanzen möglich.

Außerdem könnte die Farbauswahl bei der Anzeige der gewürfelten Augenzahl durch den Benutzer frei wählbar gestaltet werden.

Denkbar ist auch eine anschließende Vererbung, um im Rahmen einer Klassenhierarchie in den abgeleiteten Klassen Besonderheiten, z. B. die Ausgabe der gewürfelten Augen als römische Zahlen, zu realisieren.

Eine weitere mögliche Verortung besteht in den mit einem Standardwürfel gewürfelten Augen, die in einer Häufigkeitstabelle zusammengefasst sind. Anschließend erfolgen die Berechnung der relativen Häufigkeit und deren Vergleich. Allerdings muss eine hinreichend große Wurfanzahl umgesetzt werden.

In den letztgenannten Fällen erhöht sich die Bearbeitungszeit entsprechend.

## Erwarteter Stand der Kompetenzentwicklung

Aufgabe	erwartete Schülerleistung	AFB prozent. Anteil																					
1	<p>Die Schülerinnen und Schüler:</p> <ul style="list-style-type: none"> <li>– führen Recherchen und analytische Suchbewegungen zu fachlichen Hintergrundinformationen der gestellten Aufgabe durch,</li> <li>– ermitteln alle notwendigen Eigenschaften und Methoden der zu erstellenden Klasse als Sollzustand,</li> <li>– dokumentieren übersichtlich und nachvollziehbar alle Eigenschaften (Variablentypen) und Methoden (Übergabeparameter mit Typangabe und Typ des Rückgabewertes der Methode).</li> </ul> <p>Eigenschaften tabellarisch dargestellt: private</p> <table border="0"> <tr> <td>Anzahl der Würfelflächen,</td> <td>Typ: int,</td> <td>Name: anzahl_flaechen</td> </tr> <tr> <td>Farbe angezeigte Augen,</td> <td>Typ: int,</td> <td>Name: farbe_auge</td> </tr> <tr> <td>Anzahl gewürfelte Augen,</td> <td>Typ: int,</td> <td>Name: anzahl_auge</td> </tr> <tr> <td>Aussehen der Augen,</td> <td>Typ: int,</td> <td>Name: aussehen_auge</td> </tr> </table> <p>Methoden tabellarisch dargestellt: public</p> <table border="0"> <tr> <td>Ausgabe Hilfetext,</td> <td>Rückgabotyp: string,</td> <td>Name: gib_hilfe</td> </tr> <tr> <td>Aktuelle Augenzahl,</td> <td>Rückgabotyp: int,</td> <td>Name: gib_auge</td> </tr> <tr> <td>Anzeigeform der Augen</td> <td>Rückgabotyp: void,</td> <td>Name: anzeige_auge</td> </tr> </table> <p>Sollspezifikation: Konsolenprogramm</p> <p>Erstellung einer Klasse gemäß Aufgabenstellung einschließlich Instanziierung zum Zwecke der Durchführung von Testverfahren.</p>	Anzahl der Würfelflächen,	Typ: int,	Name: anzahl_flaechen	Farbe angezeigte Augen,	Typ: int,	Name: farbe_auge	Anzahl gewürfelte Augen,	Typ: int,	Name: anzahl_auge	Aussehen der Augen,	Typ: int,	Name: aussehen_auge	Ausgabe Hilfetext,	Rückgabotyp: string,	Name: gib_hilfe	Aktuelle Augenzahl,	Rückgabotyp: int,	Name: gib_auge	Anzeigeform der Augen	Rückgabotyp: void,	Name: anzeige_auge	I und II 25 %
Anzahl der Würfelflächen,	Typ: int,	Name: anzahl_flaechen																					
Farbe angezeigte Augen,	Typ: int,	Name: farbe_auge																					
Anzahl gewürfelte Augen,	Typ: int,	Name: anzahl_auge																					
Aussehen der Augen,	Typ: int,	Name: aussehen_auge																					
Ausgabe Hilfetext,	Rückgabotyp: string,	Name: gib_hilfe																					
Aktuelle Augenzahl,	Rückgabotyp: int,	Name: gib_auge																					
Anzeigeform der Augen	Rückgabotyp: void,	Name: anzeige_auge																					

2	<p>Die Schülerinnen und Schüler dokumentieren die Klassenstruktur in üblicher UML-Form.</p> <p>Klassendiagramm:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p style="text-align: center;"><b>Wuerfel</b></p> <p>-anzahl_flaechen: int          -farbe_auge:n: int          -anzahl_auge:n: int          -aussehen_auge:n: int</p> <p>+gib_hilfe(): string          +gib_auge:n(): int          +anzei:ge_auge:n(): void</p> </div>	I und II 15 %
3	<p>Die Schülerinnen und Schüler implementieren ihren Entwurf in einer vorgegebenen Programmiersprache, z. B.:</p> <pre> #include &lt;iostream&gt; #include &lt;cstdio&gt; #include &lt;windows&gt; using namespace std;  class wuerfel { private:     int anzahl_flaechen;     int farbe_auge:n;     int anzahl_auge:n;     int aussehen_auge:n; public:     string gib_hilfe(void);     int gib_auge:n(void);     void anzei:ge_auge:n(void);     wuerfel(int, int, int);     ~wuerfel(); };  wuerfel::wuerfel(int a, int f, int g) {     anzahl_flaechen = a;     farbe_auge:n = f;     aussehen_auge:n = g; }  wuerfel::~wuerfel() {  }  string wuerfel::gib_hilfe(void) {     string hilfe = "\n\nBeschreibung zu diesem W\x81rfel! \n===== \n\n";     hilfe += "Der W\x81rfel (platonischer K\x94rper) kann folgende Anzahl n \n";     hilfe += "Fl\x84 \bchen (Augen) maximal annehmen: 4, 6, 8, 12 oder 20. </pre>	II und III 45 %

```

    \n\n";
return hilfe;
}

int wuerfel::gib_auge(n void) {
    anzahl_auge(n = rand() % anzahl_flaeche(n) + 1;
return anzahl_auge(n);
}

void wuerfel::anzei ge_auge(n void) {
    int i;
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), farbe_auge(n));
    if (aussehen_auge(n) == 0) {
        for (i = 1; i <= anzahl_auge(n); i++) {
            cout << "0";
        }
    } else {
        cout << anzahl_auge(n);
    }
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 15);
}

//-----
int main() {
    // Möglichkeit zum Testen der Klasse
    unsigned i;
    srand((unsigned)time(NULL));

    wuerfel w1(4, 8, 0);
    wuerfel w2(6, 9, 1);
    wuerfel w3(8, 12, 0);
    wuerfel w4(12, 13, 1);
    wuerfel w5(20, 14, 0);

    cout << w1.gib_hilfe();
    cout << "\n\n";
    for (i = 1; i <= 10; i++) {
        cout << w1.gib_auge(n) << "\t";
        w1.anzei ge_auge(n);
        cout << "\n" << w2.gib_auge(n) << "\t";
        w2.anzei ge_auge(n);
        cout << "\n" << w3.gib_auge(n) << "\t";
        w3.anzei ge_auge(n);
        cout << "\n" << w4.gib_auge(n) << "\t";
        w4.anzei ge_auge(n);
        cout << "\n" << w5.gib_auge(n) << "\t";
        w5.anzei ge_auge(n); cout << "\n\n";
    }
    getch();
    return 0;
}

```

4	<p>Die Schülerinnen und Schüler planen und dokumentieren die Vorgehensweise der Testdurchführung in Stichpunkten:</p> <ul style="list-style-type: none"> <li>• Test mit gültigen Werten: 4, 6, 8, 12 oder 20 Flächen sind sinnvoll,</li> <li>• Test der Funktion des Zufallsgenerators mit den verschiedenen Würfeln,</li> <li>• Anzeige der gewürfelten Augenzahl mit verschiedenen Vordergrundfarben und in Darstellung arabischer Zahlen oder n mal den Buchstaben 'o'.</li> </ul> <p>– Die Testdurchführung ist frei gestaltbar und könnte beispielsweise mit mehreren Instanzen erfolgen und mithilfe einer Schleife effektiver ablaufen.</p> <p>Die Schülerinnen und Schüler:</p> <ul style="list-style-type: none"> <li>– schätzen sich selbst kritisch ein, vervollständigen die einzelnen, Dokumentationsteile und deren erläutern diese fachgerecht,</li> <li>– präsentieren ihre Problemlösungen.</li> </ul>	II und III 15 %
---	---	--------------------