



Schriftliche Abiturprüfung Schuljahr 2022/2023

Informatik
auf grundlegendem Anforderungsniveau
an allgemeinbildenden und beruflichen gymnasialen Oberstufen

Haupttermin
Donnerstag, 20. April 2023, 09:00 Uhr

Unterlagen für die Prüflinge

Allgemeine Arbeitshinweise

- Überprüfen Sie diese Unterlagen auf Vollständigkeit.
- Schreiben Sie auf alle Prüfungsunterlagen Ihren Namen und zusätzlich auf dieses Deckblatt Ihre Kursnummer.
- Kennzeichnen Sie Ihre Entwurfsblätter (Kladde) und Ihre Reinschrift.

Fachspezifische Arbeitshinweise¹

- Die Arbeitszeit beträgt **270 Minuten**.²
- Eine Lese- und Auswahlzeit von **30 Minuten** ist der Arbeitszeit vorgeschaltet. In dieser Zeit darf noch nicht mit der Bearbeitung der Aufgaben begonnen werden.
- Hilfsmittel: Taschenrechner (nicht programmierbar, nicht grafikfähig), Formelsammlung „Das große Tafelwerk“ (Cornelsen-Verlag), Rechtschreibwörterbuch

Aufgabenauswahl

- Sie erhalten **drei** Aufgaben zu unterschiedlichen Schwerpunkten.
- Bearbeiten Sie die **Pflichtaufgabe (Aufgabe I)** entweder in der Java-Version oder in der Python-Version.
- Wählen Sie **eine** weitere Aufgabe (Aufgabe II oder III) aus und bearbeiten Sie diese. Für Aufgabe III können Sie zwischen einer Dynasys- und einer Consideo-Version wählen.
- Vermerken Sie auch auf Ihrer Reinschrift, welche Aufgaben Sie ausgewählt und bearbeitet haben.

Bearbeitet wurden die folgenden Aufgaben (bitte kreuzen Sie an):

I	Objektorientierte Modellierung und Programmierung (Pflicht)	(<input type="checkbox"/> Java <input type="checkbox"/> Python)
II	Datensicherheit in verteilten Systemen	
III	Simulation dynamischer Systeme	(<input type="checkbox"/> Dynasys <input type="checkbox"/> Consideo)

¹ Entsprechend der „Richtlinie über die Gewährung von Erleichterungen für neu zugewanderte Schülerinnen, Schüler und Prüflinge bei Sprachschwierigkeiten in der deutschen Sprache“ (MBISchul Nr. 08, 7. Oktober 2016, S. 60) werden für die betroffenen Prüflinge die folgenden Erleichterungen gewährt:

- Die Bearbeitungszeit wird um 30 Minuten auf **300 Minuten** erhöht.

- Ein nicht-elektronisches Wörterbuch Deutsch – Herkunftssprache / Herkunftssprache – Deutsch wird bereitgestellt.

² Gemäß Entscheidung vom 5. Januar 2023.

Bewertung

Jeder Aufgabe sind 50 Bewertungseinheiten (BE) zugeordnet. In allen Teilaufgaben werden nur ganze BE vergeben. Insgesamt sind 100 BE erreichbar. Bei der Festlegung von Notenpunkten gilt die folgende Tabelle:

Erbrachte Leistung (in BE)	Notenpunkte
≥ 95	15
≥ 90	14
≥ 85	13
≥ 80	12
≥ 75	11
≥ 70	10
≥ 65	9
≥ 60	8

Erbrachte Leistung (in BE)	Notenpunkte
≥ 55	7
≥ 50	6
≥ 45	5
≥ 40	4
≥ 33	3
≥ 27	2
≥ 20	1
< 20	0

Für die Erteilung der **Note gut** (11 Punkte) ist mindestens erforderlich, dass annähernd vier Fünftel der erwarteten Gesamtleistung sowie Leistungen in allen drei Anforderungsbereichen erbracht werden. Dabei muss die Prüfungsleistung in ihrer Gliederung, in der Gedankenführung, in der Anwendung fachmethodischer Verfahren sowie in der fachsprachlichen Artikulation den Anforderungen voll entsprechen.

Für die Erteilung der **Note ausreichend** (5 Punkte) ist mindestens erforderlich, dass annähernd die Hälfte der erwarteten Gesamtleistung und über den Anforderungsbereich I hinaus Leistungen in einem weiteren Anforderungsbereich erbracht werden.

Die zwei voneinander unabhängigen Aufgaben der Prüfungsaufgabe werden jeweils mit 50 Bewertungseinheiten bewertet. Die erbrachte Gesamtleistung ergibt sich aus der Summe der Bewertungseinheiten in den beiden Aufgaben.

Bei erheblichen Mängeln in der sprachlichen Richtigkeit und der äußerlichen Form sind bei der Bewertung der schriftlichen Prüfungsleistung je nach Schwere und Häufigkeit der Verstöße bis zu zwei Notenpunkte abzuziehen. Dazu gehören auch Mängel in der Gliederung, Fehler in der Fachsprache, Ungenauigkeiten in Zeichnungen sowie falsche Bezüge zwischen Zeichnungen und Text.

Aufgabe I: Grundriss (Java-Version)

50 BE

Schwerpunkt: Objektorientierte Modellierung und Programmierung von Grafiksystemen

Für Architektenzeichnungen soll eine Software zur Darstellung von Grundrissen entwickelt werden. Eine erste Version kann Wände, Türen und Fenster darstellen. Ein Beispiel zeigt die folgende Abbildung 1:

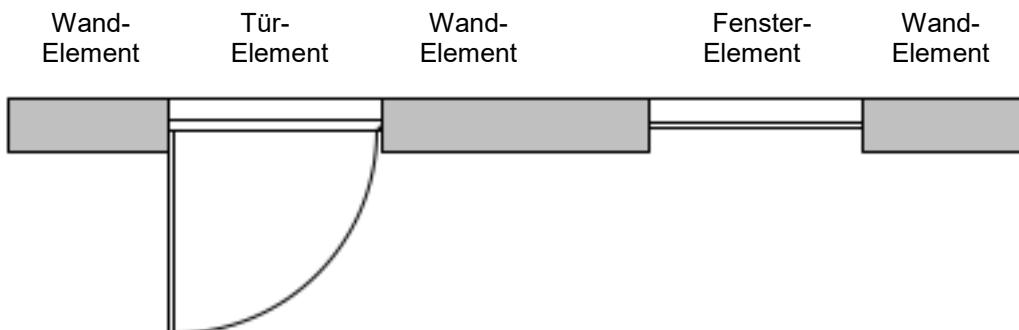


Abbildung 1

Die folgende Abbildung 2 zeigt ein passendes Klassendiagramm zu dieser Software-Version:

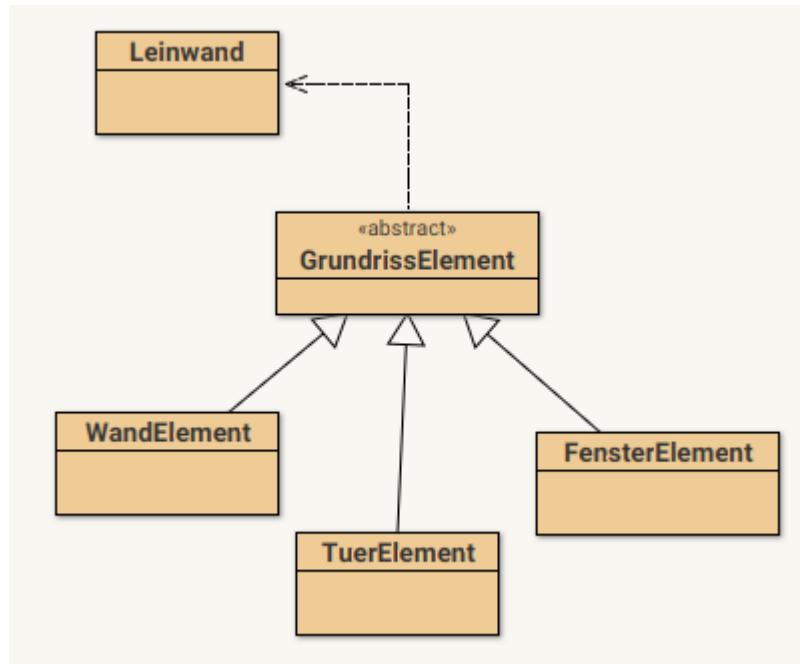


Abbildung 2

I.a **Erläutern** Sie das angegebene Klassendiagramm (Abbildung 2) zu dieser ersten Version der Software.

(6 BE)

I.b **Beschreiben** Sie, wodurch

- Vererbungsbeziehungen und
- Nutzerbeziehungen (auch: Verwendungsbeziehungen)

jeweils im Programmtext einer Klasse erkennbar sind, und **erläutern** Sie diese Beziehungstypen jeweils an einem einfachen Beispiel im Kontext der Aufgabe.

(7 BE)

- I.c **Begründen** Sie, warum es sinnvoll ist, eine Klasse `GrundrissElement` (vgl. Abbildung 2) in der Modellierung einzuführen.

(6 BE)

Beim Einbauen weiterer Wandelemente, die in der ersten Modellierung als einfache Rechtecke modelliert worden sind, treten einige Ränder der Rechtecke als Trennstriche zu anderen Wandelementen auf (siehe Abbildung 3).

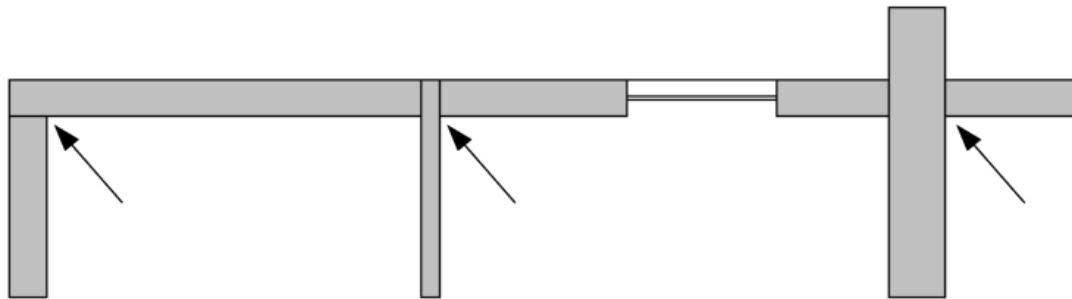


Abbildung 3

Die Entwickler*innen möchten Wandabschnitte aber mit einer durchgehenden hellgrauen Füllung dargestellt haben.

Als zweiten Entwurf verwenden sie für Wandelemente nicht ein einfaches Rechteck, sondern ein n-Eck, also eine geschlossene Figur mit gradlinigen Kanten, die mehr als vier Eckpunkte haben kann. Die folgende Abbildung 4 zeigt dazu ein Beispiel:

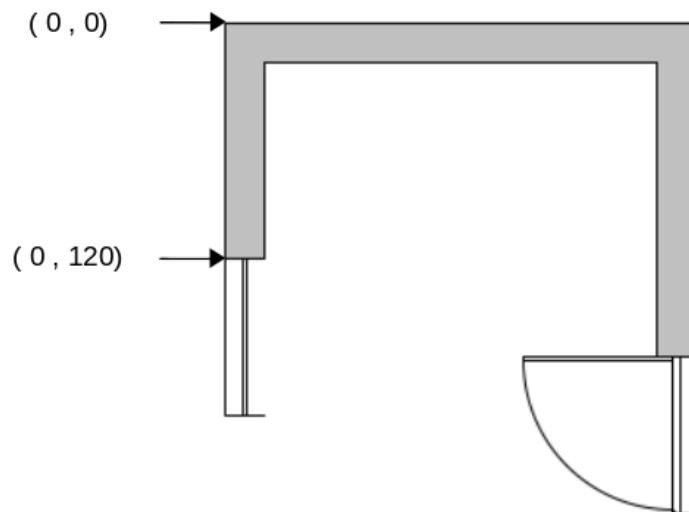


Abbildung 4

Bezogen auf die linke obere Ecke mit den (relativen) Koordinaten (0,0) können die (ebenfalls relativen) Koordinaten der Eckpunkte leicht angegeben werden, wenn man die einzelnen Längen kennt. Die Entwickler*innen entscheiden sich, die Daten intern in einer `ArrayList` zu speichern, die Elemente einer (*internen*) Klasse `Ecke` verwaltet (siehe dazu die Anlage 2).

- I.d **Geben** Sie die (relativen) Koordinaten in Bildschirmpixeln in einer passenden Reihenfolge zu dem dargestellten Wandelement in Form einer Liste `an`. Verwenden Sie für diese Teilaufgabe auch das Bild in der Anlage 1, das mit Bemessungslinien versehen ist.

(6 BE)

Durch das Hinzufügen von Ecken ändern sich in einigen Fällen die Breite und/oder die Tiefe des Wandelements.

- I.e • **Berechnen** Sie die passenden Attributwerte von `breite` und `tiefe` zu dem dargestellten `WandElement` (siehe dazu auch die Anlage 1).

- **Entwickeln** Sie eine Methode `public void aktualisiereAbmessungen()` zur Bestimmung der aktuellen Werte der Attribute `breite` und `tiefe` für die Klasse `WandElement`.

Hinweise: Gehen Sie dabei davon aus, dass die linke obere Ecke mit den (relativen) Koordinaten (0,0) weiterhin die Ausgangscke der Figur bleibt, aber nicht notwendig die oberste oder am weitesten links liegende Ecke sein muss.

Die Klasse `GrundrissElement` stellt die beiden Methoden

`aendereBreite(neueBreite)` und `aendereTiefe(neueTiefe)` bereit.

(16 BE)

Die Entwickler*innen fürchten, dass die Bestimmung der Eckpunkte für die Anwender*innen möglicherweise zu kompliziert ist. Daher versuchen sie einen anderen Entwurf, bei dem Wandelemente allein rechteckig sind, zusätzlich aber die Klassen `EckElement`, `TElement` und `PlusElement` realisiert werden (siehe dazu auch die Anlage 3 derselben Grundrisselemente in zerfallender Darstellung).

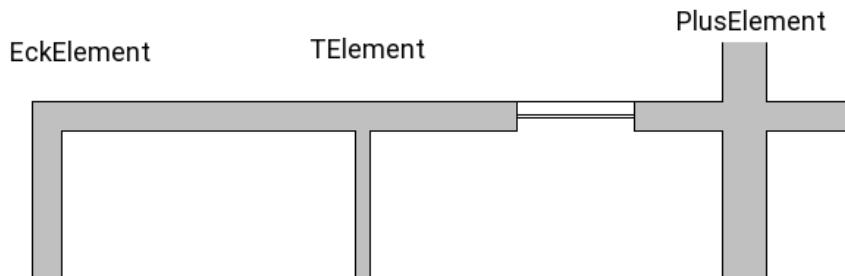
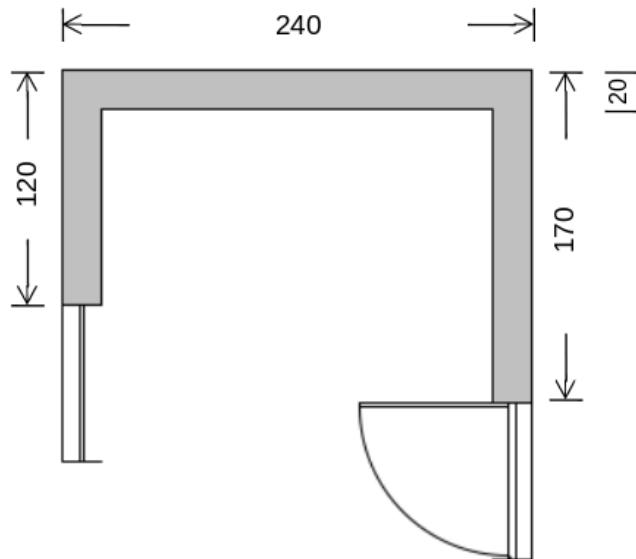


Abbildung 5

- I.f **Stellen** Sie eine mögliche Erweiterung des Klassendiagramms um die Klassen (`EckElement`, `TElement` und `PlusElement`) **dar**. (Alternativ können Sie diese Erweiterung auch **skizzieren**.)

(9 BE)

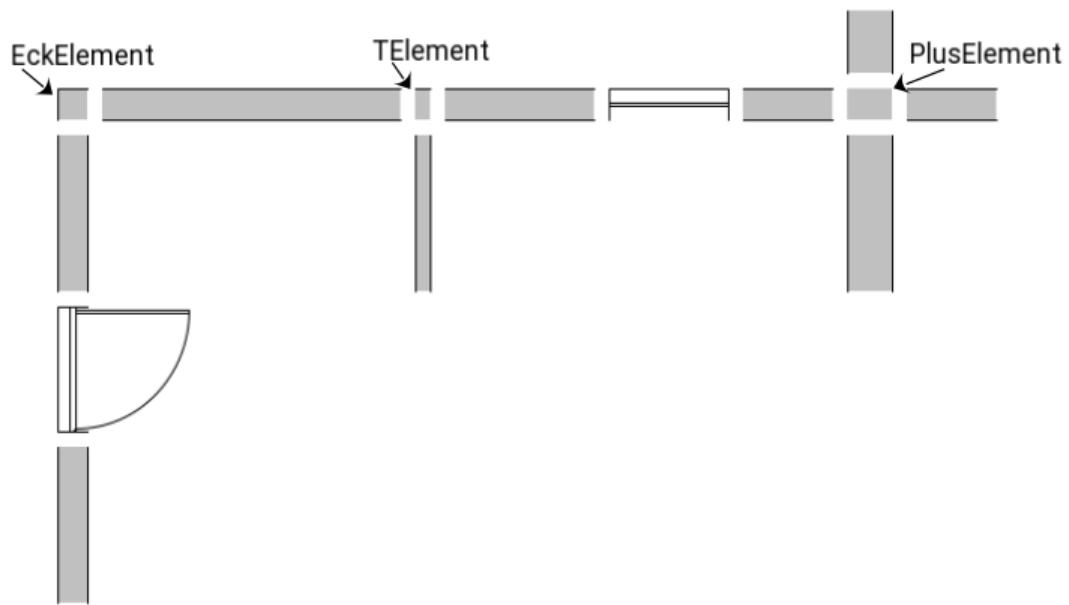
Anlagen zur Aufgabe I Grundriss (Java-Version)**Anlage 1** Bild mit Bemessungslinien

Alle Größen sind in Pixel angegeben. Die Wanddicke der WandElemente ist durchgehend 20 Pixel.

Anlage 2 Interne Klasse Ecke

```
1  /*
2   *  * Interne Klasse Ecke
3   */
4  private class Ecke
5  {
6      private int x,y;
7
8      public Ecke(int x, int y) {
9          this.x=x;
10         this.y=y;
11     }
12
13     public Ecke(int[] koordinatenPaar) {
14         this.x=koordinatenPaar[0];
15         this.y=koordinatenPaar[1];
16     }
17
18     public int gibX() {return x;}
19     public int gibY() {return y;}
20
21 }
```

Anlage 3 Elemente in getrennter Darstellung



Aufgabe I: Grundriss (Python-Version)**50 BE****Schwerpunkt: Objektorientierte Modellierung und Programmierung von Grafiksystemen**

Für Architektenzeichnungen soll eine Software zur Darstellung von Grundrissen entwickelt werden. Eine erste Version kann Wände, Türen und Fenster darstellen. Ein Beispiel zeigt die folgende Abbildung 1:

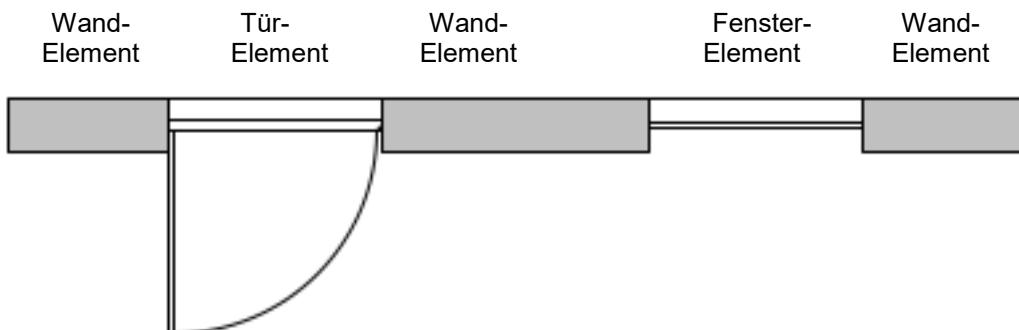


Abbildung 1

Die folgende Abbildung 2 zeigt ein passendes Klassendiagramm zu dieser Software-Version:

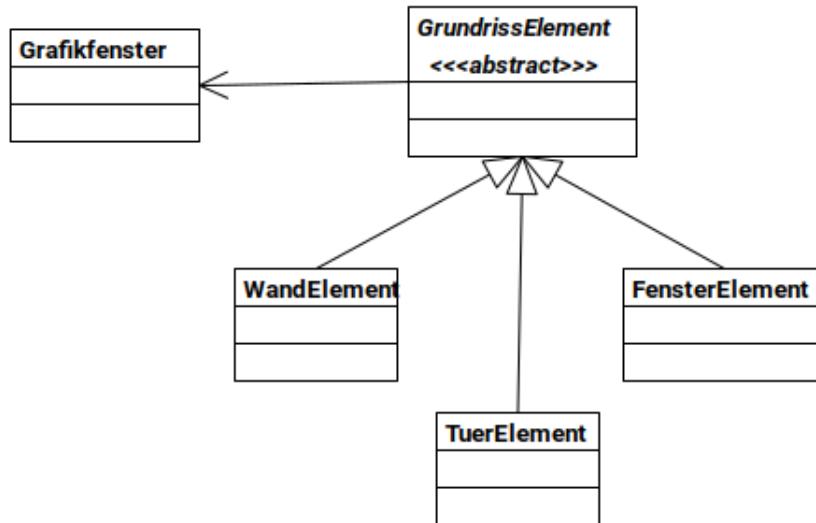


Abbildung 2

- I.a **Erläutern** Sie das angegebene Klassendiagramm (Abbildung 2) zu dieser ersten Version der Software.

(6 BE)

- I.b **Beschreiben** Sie, wodurch

- Vererbungsbeziehungen und
- Nutzerbeziehungen (auch: Verwendungsbeziehungen)

jeweils im Programmtext einer Klasse erkennbar sind, und **erläutern** Sie diese Beziehungstypen jeweils an einem einfachen Beispiel im Kontext der Aufgabe.

(7 BE)

- I.c **Begründen** Sie, warum es sinnvoll ist, eine Klasse `GrundrissElement` (vgl. Abbildung 2) in der Modellierung einzuführen.

(6 BE)

Beim Einbauen weiterer Wandelemente, die in der ersten Modellierung als einfache Rechtecke modelliert worden sind, treten einige Ränder der Rechtecke als Trennstriche zu anderen Wandelementen auf (siehe Abbildung 3).

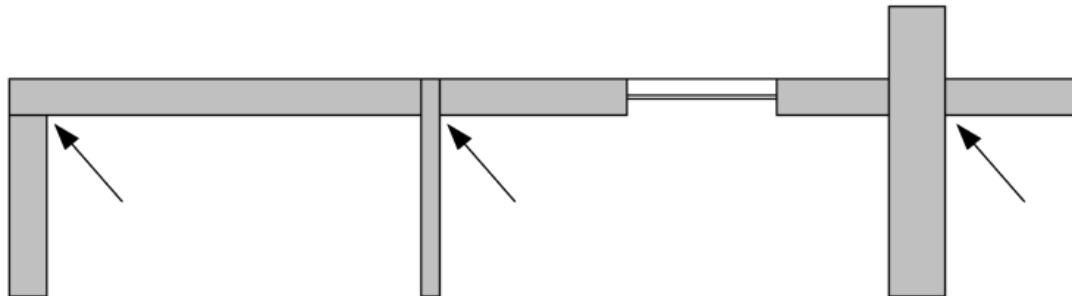


Abbildung 3

Die Entwickler*innen möchten Wandabschnitte aber mit einer durchgehenden hellgrauen Füllung dargestellt haben.

Als zweiten Entwurf verwenden sie für Wandelemente nicht ein einfaches Rechteck, sondern ein n-Eck, also eine geschlossene Figur mit gradlinigen Kanten, die mehr als vier Eckpunkte haben kann. Die folgende Abbildung 4 zeigt ein Beispiel:

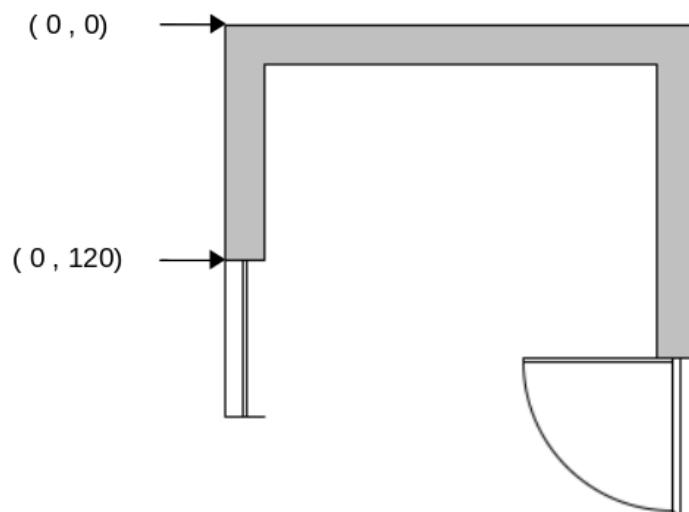


Abbildung 4

Bezogen auf die linke obere Ecke mit den (relativen) Koordinaten (0,0) können die (ebenfalls relativen) Koordinaten der Eckpunkte leicht angegeben werden, wenn man die einzelnen Längen kennt. Die Entwickler*innen entscheiden sich, die Daten in einer Liste zu speichern.

- I.d **Geben** Sie die (relativen) Koordinaten in Bildschirmpixeln in einer passenden Reihenfolge zu dem dargestellten Wandelement in Form einer Liste `an`. Verwenden Sie für diese Teilaufgabe auch das Bild in der Anlage 1, das mit Bemessungslinien versehen ist.

(6 BE)

Durch das Hinzufügen von Ecken ändern sich in einigen Fällen die Breite und/oder die Tiefe des Wandelements.

- I.e • **Berechnen** Sie die passenden Attributwerte von `breite` und `tiefe` zu dem dargestellten `WandElement` (siehe dazu auch die Anlage 1).

- **Entwickeln** Sie eine Methode `AktualisiereAbmessungen(self)` zur Bestimmung der aktuellen Werte der Attribute `breite` und `tiefe` für die Klasse `WandElement`.

Hinweise: Gehen Sie dabei davon aus, dass die linke obere Ecke mit den (relativen) Koordinaten (0,0) weiterhin die Ausgangscke der Figur bleibt, aber nicht notwendig die oberste oder am weitesten links liegende Ecke sein muss.

Die Klasse `GrundrissElement` stellt die beiden Methoden

`AendereBreite(self, neueBreite)` und `AendereTiefe(self, neueTiefe)` *bereit.*

(16 BE)

Die Entwickler*innen fürchten, dass die Bestimmung der Eckpunkte für die Anwender*innen möglicherweise zu kompliziert ist. Daher versuchen sie einen anderen Entwurf, bei dem Wandelemente allein rechteckig sind, zusätzlich aber die Klassen `EckElement`, `TElement` und `PlusElement` realisiert werden (siehe dazu auch die Anlage 3 derselben Grundrisselemente in zerfallender Darstellung).

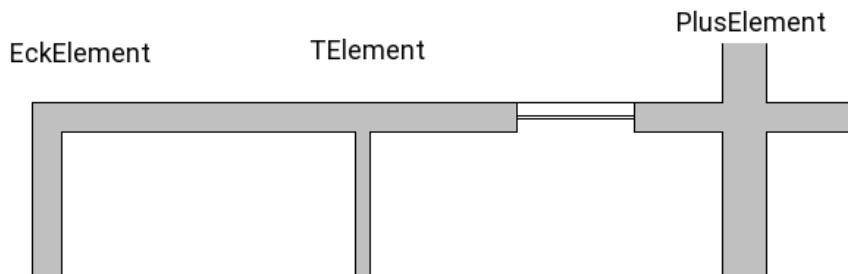
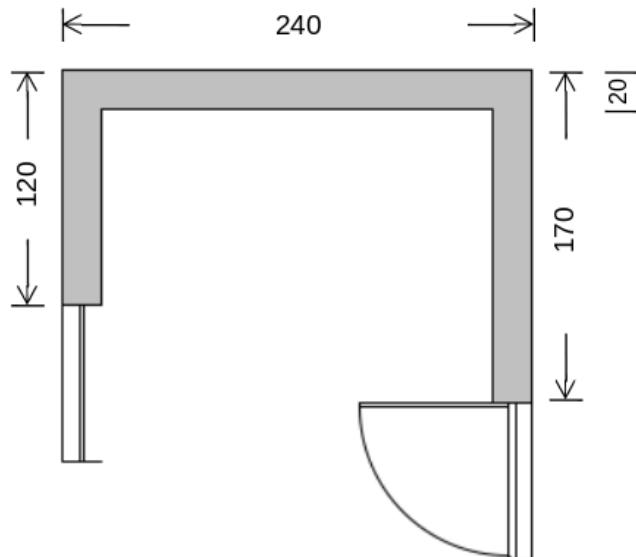


Abbildung 5

- I.f **Stellen** Sie eine mögliche Erweiterung des Klassendiagramms um die Klassen (`EckElement`, `TElement` und `PlusElement`) **dar**. (Alternativ können Sie diese Erweiterung auch **skizzieren**.)

(9 BE)

Anlagen zur Aufgabe I Grundriss (Python-Version)**Anlage 1** Bild mit Bemessungslinien

Alle Größen sind in Pixel angegeben. Die Wanddicke der WandElemente ist durchgehend 20 Pixel.

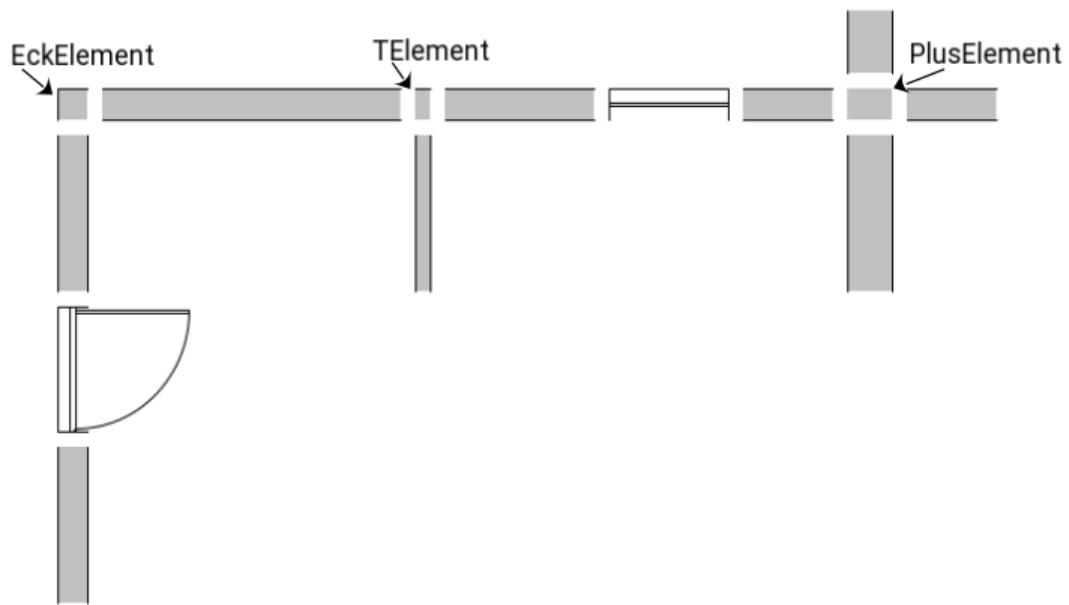
Anlage 2 Klasse Ecke

```

1  class Ecke:
2      '''Eckpunkte fuer WandElement'''
3
4      def __init__(self, parameter1, parameter2=None):
5          '''Konstruktor ermoeglicht zwei Arten von Aufrufen:
6          Ecke( 2 , 3 ) oder Ecke( (2,3) )'''
7          if parameter2==None:
8              self.__x,self.__y=parameter1
9          else:
10             self.__x,self.__y=parameter1,parameter2
11
12     def GibX(self):
13         return self.__x
14     def GibY(self):
15         return self.__y

```

Anlage 3 Elemente in getrennter Darstellung



Aufgabe II: Datensicherheit in der Cloud**50 BE****Schwerpunkt: Datensicherheit in verteilten Systemen**

Immer wieder werden interne Daten der Firma *Skilpaddel* öffentlich bekannt. Dies gilt sowohl für Daten, die nie über das Internet verschickt worden sind, als auch für Daten, die in einer Cloud¹ gespeichert werden, was auf Sicherheitslücken innerhalb der Firma hindeutet.

Die Sicherheitschefin der Firma möchte deshalb als eine erste Maßnahme die Dokumente, die auf den Rechnern der Firma gespeichert sind, verschlüsseln.

Sie stellt der Firmenleitung verschiedene symmetrische und asymmetrische Verfahren vor. Die Firmenleitung entscheidet sich schließlich dafür, ein symmetrisches Verfahren einzusetzen.

- II.a
- **Nennen** Sie die wesentlichen Unterschiede zwischen symmetrischen und asymmetrischen Verfahren.
 - **Stellen** Sie **dar**, warum die Entscheidung der Firmenleitung für ein symmetrisches Verfahren hier passend ist.

(14 BE)

Die Firmenleitung hat sich weiter informiert und gelesen, dass symmetrische Verfahren, bei denen der Schlüssel mindestens so lang ist wie der zu verschlüsselnde Text, nicht knackbar sind. Sie entscheidet, dass ein solches Verfahren innerhalb der Firma für die Verschlüsselung der internen Dokumente eingeführt werden soll.

Damit sich die Angestellten der Firma den Schlüssel zum Ver- und Entschlüsseln ihrer eigenen Dokumente leichter merken können, gibt sie die Funktion `schluesselErzeugung` zur Schlüsselgenerierung vor, die Sie in Anlage 3 zu dieser Aufgabe finden.

- II.b
- **Stellen** Sie die Funktionsweise der Methode `schluesselErzeugung` (siehe Anlage 3 zu dieser Aufgabe) im Detail **dar**.
 - **Begründen** Sie, dass dieses Verfahren zur Schlüsselgenerierung nach dem Prinzip von Kerckhoffs nicht sicher ist.
 - **Untersuchen** Sie, ob bei einem nach Kerckhoffs' Prinzip geeigneten Schlüssel das hier vorgestellte Verfahren für die Firma eingesetzt werden sollte.

(15 BE)

Da für die Anbindung verschiedener Geschäftsstellen regelmäßig Informationen auch in Cloud-Diensten (siehe Anlage 1) bereitgestellt werden müssen, geht die Sicherheitschefin nun der Frage nach, ob es hier eventuell zusätzliche Sicherheitslücken gibt.

- II.c
- **Stellen** Sie mögliche Sicherheitsrisiken **dar**, die durch die Nutzung eines Cloud-Dienstes entstehen können.

(8 BE)

Der Cloud-Dienstleister der Firma *Skilpaddel* wurde danach ausgewählt, dass er eine Ende-zu-Ende-Verschlüsselung anbietet (siehe Anlage 2). Annabel, eine Mitarbeiterin der Firma, möchte Max, einem als vertrauenswürdig eingestuften Kollegen, Daten über die Cloud zur Verfügung stellen.

- II.d
- **Erklären** Sie anhand des gegebenen Sachkontexts, wie eine „echte“ Ende-zu-Ende-Verschlüsselung aussehen könnte, bei der Annabel ihre Daten in der Cloud speichern und Max sie dann herunterladen und lesen kann.
 - **Untersuchen** Sie, welche Sicherheitsrisiken sich bei einer passwortbasierten „Ende-zu-Ende-Verschlüsselung“ (siehe Klasse 2 in der Anlage 2) ergeben.

(13 BE)

¹ Eine Cloud sind ein oder mehrere Server, auf die über das Internet zugegriffen wird. Die Cloud-Server befinden sich in Rechenzentren, die weltweit verteilt sein können. (Für eine ausführliche Erläuterung siehe die Anlagen 1 und 2.)

Anlagen zur Aufgabe II Datensicherheit in der Cloud

Anlage 1 Cloud-Computing

„Cloud Computing ist ein Modell, das es erlaubt, bei Bedarf jederzeit und überall bequem über ein Netz auf einen geteilten Pool von konfigurierbaren Rechnerressourcen (z. B. Netze, Server, Speichersysteme, Anwendungen und Dienste) zuzugreifen, die schnell und mit minimalem Managementaufwand oder geringer Serviceprovider-Interaktion zur Verfügung gestellt werden können.“

(Definition der US-amerikanischen Standardisierungsstelle NIST (*National Institute of Standards and Technology*), die auch von der ENISA (*European Network and Information Security Agency*) genutzt wird)

Ein Cloud-Dienstleister bietet diese Services an.

Anlage 2 Datenverschlüsselung in der Cloud

„Daten in Cloud-Diensten zu verschlüsseln, ist weitaus schwieriger. Die Datenverarbeitung findet beim Diensteanbieter statt. Das heißt, eine echte Ende-zu-Ende-Verschlüsselung ist beinahe unmöglich, falls der Diensteanbieter dies nicht explizit unterstützt.

[...]

Klasse 1: "Echte" Ende-zu-Ende-Verschlüsselung

Die Daten werden auf dem Endgerät des Anwenders verschlüsselt. Der zur Verschlüsselung genutzte Schlüssel wird lokal auf dem Endgerät erzeugt und gespeichert. Ein Upload des Schlüssels in die Cloud erfolgt nicht. Der Anwender muss zur Vermeidung von Datenverlusten selbst für ein Schlüssel-Backup sorgen. Die eigentlichen Daten werden vor dem Upload in die Cloud durch das Endgerät verschlüsselt, entsprechend erfolgt die Entschlüsselung nur auf dem Gerät.

Klasse 2: Passwortbasierte Ende-zu-Ende-Verschlüsselung

Der Dienst erzeugt den zur Verschlüsselung genutzten Schlüssel direkt oder indirekt aus dem Passwort des Anwenders. Das Passwort wird nicht gespeichert, sondern beim Login in die Applikation wird nach Industriestandards durch eine sichere Hashwert-Berechnung der Verschlüsselungsschlüssel wiederhergestellt. Die Ver- und Entschlüsselung der Daten erfolgt in der Applikation auf dem Server des Diensteanbieters. Da das Passwort und die Daten erst auf dem Server verschlüsselt werden, ist eine funktionierende Transportverschlüsselung (HTTPS) erforderlich.

Sollten Sie der Verschlüsselung der Klasse 2 vertrauen, ist die Frage nach dem Fall des Passwortverlustes zusätzlich interessant. Ist es dem Anbieter möglich, Ihre Daten auch bei Passwortverlust wiederherzustellen, so erfolgt eine zusätzliche Verschlüsselung Ihrer Daten mit einem Masterkey oder einer vergleichbaren Lösung. Hierfür gibt es verschiedene Implementierungsmöglichkeiten, die die Stärke der Verschlüsselung massiv abschwächen können.“²

² Hißen, Frank (16.10.2015): Verschlüsselung ist nicht gleich Verschlüsselung, Datenverschlüsselung in der Cloud, [online] <https://www.tecchannel.de/a/verschlüsselung-ist-nicht-gleich-verschlüsselung,3281630,2> [abgerufen am 02.07.2022].

Anlage 3 Implementierung zu Aufgabenteil II.b**Hinweise:**

Die Implementierung der Funktion liegt hier sowohl in einer Java- als auch in einer Python-Version (siehe folgende Seite) vor. Beziehen Sie sich bei der Bearbeitung der Aufgabe auf eine der beiden Versionen.

Zur Vereinfachung werden jeweils lediglich Kleinbuchstaben verarbeitet. „97“ ist der ASCII-Code für den Buchstaben „a“.

Java-Version:

```

1  public void schluesselErzeugung (String vorname,
2                               String nachname,
3                               int textLaenge,
4                               int wert)
5  {
6      String neuSchluessel= vorname + nachname;
7      String gesamtSchluessel = neuSchluessel;
8      int verschiebung = wert;
9
10     while (gesamtSchluessel.length() < textLaenge){
11         gesamtSchluessel = gesamtSchluessel +
12                         verschieben(neuSchluessel, verschiebung);
13         verschiebung += 1;
14     }
15     System.out.println("Chiffre: " + gesamtSchluessel);
16 }
17
18
19 public String verschieben(String schluessel, int wert)
20 {
21     int i=0;
22     int buchstabeAlsZahl;
23     char buchstabeNeu;
24     String schluesselNeu="";
25
26     while (i < schluessel.length()){
27         buchstabeAlsZahl = (int) schluessel.charAt(i);
28         buchstabeAlsZahl = buchstabeAlsZahl - 97;
29         buchstabeAlsZahl = (buchstabeAlsZahl + wert)% 26 + 97;
30         buchstabeNeu = (char) buchstabeAlsZahl;
31         schluesselNeu = schluesselNeu + buchstabeNeu;
32         i += 1;
33     }
34     return schluesselNeu;
35 }
```

Python-Version:

```
1 def schluesselErzeugung(vorname, nachname, textLaenge, wert):
2     neuSchluessel = vorname + nachname
3     gesamtSchluessel = neuSchluessel
4     verschiebung = wert
5     while len(gesamtSchluessel) < textLaenge:
6         gesamtSchluessel = gesamtSchluessel + verschieben(neuSchluessel,
7         verschiebung)
7         print("Chiffre: " + gesamtSchluessel)
8         verschiebung += 1
9
10
11 def verschieben(schluessel, wert):
12     i = 0
13     schluesselNeu = ""
14
15     while i < len(schluessel):
16         buchstabeAlsZahl = ord(schluessel[i])
17         buchstabeAlsZahl = buchstabeAlsZahl - 97
18         buchstabeAlsZahl = (buchstabeAlsZahl + wert) % 26 + 97
19         buchstabeNeu = chr(buchstabeAlsZahl)
20         schluesselNeu = schluesselNeu + buchstabeNeu;
21         i += 1;
22
23     return schluesselNeu
```

Aufgabe III: Eichhörnchen (Dynasys-Version)**50 BE****Schwerpunkt: Simulation dynamischer Systeme**

In Hamburg ist das Eurasische Eichhörnchen (*Sciurus vulgaris*) weit verbreitet. Es hat eine rötliche Fellfärbung, wobei selbst die Geschwister innerhalb eines Wurfs unterschiedlich dunkles Fell bis hin zu schwarz haben können. Eichhörnchen sind Einzelgänger.

In den USA ist das Grauhörnchen (*Sciurus carolinensis*) heimisch. Die Tiere sind etwas größer, haben ein graues Fell und leben gesellig in Gruppen. In Großbritannien und in Norditalien wurden Grauhörnchen ausgesetzt. Sie verbreiten sich nun stark und verdrängen die einheimischen roten Eichhörnchen, so dass in England und Wales inzwischen fast ausschließlich Grauhörnchen leben und nur in Schottland noch rote Eichhörnchen zu finden sind. Aktuell geht man von 140.000 roten und 2.500.000 grauen Eichhörnchen in Großbritannien aus.

Die Grauhörnchen haben den Eurasischen Eichhörnchen gegenüber den Vorteil, dass sie Eicheln gut verdauen können und ihre Futterverstecke im Winter sicherer wiederfinden. Rote Eichhörnchen kommen dagegen besser mit dem Futterangebot in Nadelwäldern zurecht.



Abb. 1: Rotes Eichhörnchen mit einer Walnuss,
Wikimedia-Nutzer Sven-Christoph Petersen
CC-BY 3.0



Abb. 2: Grauhörnchen im Park Bunhill Fields (London),
Wikimedia-Nutzer HH58, CC-BY-SA 4.0

- III.a
- **Beschreiben** Sie die Wachstumsform exponentielles Wachstum an einem selbst gewählten Beispiel und gehen Sie dabei auf die Voraussetzungen für ihr Auftreten ein.
 - **Zeichnen** Sie ein passendes Modelldiagramm.
 - **Skizzieren** Sie den Verlauf der Werte des Bestandsfaktors in einem Zeitdiagramm.

(10 BE)

Das folgende Modelldiagramm beschreibt zusammen mit den Simulationsgleichungen ein biologisches System.

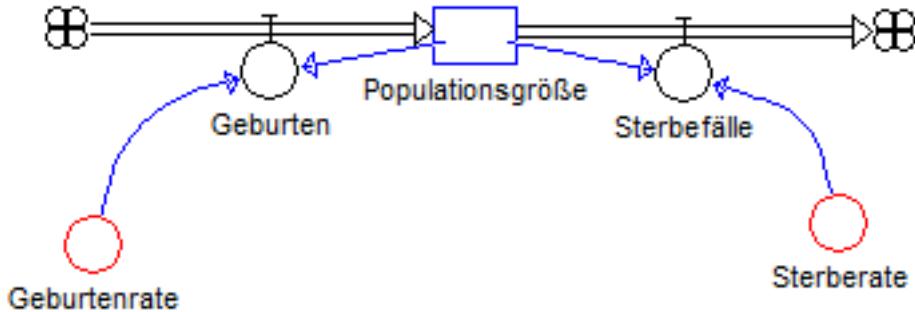


Abb. 3: Modelldiagramm

Zustandsgleichungen:

$$\text{Populationsgröße.neu} = \text{Populationsgröße.alt} + dt * (\text{Geburten} - \text{Sterbefälle})$$

$$\text{Startwert Populationsgröße} = 100$$

Zustandsänderungen:

$$\text{Geburten} = \text{Geburtenrate} * \text{Populationsgröße}$$

$$\text{Sterbefälle} = \text{Sterberate} * \text{Populationsgröße}$$

Parameter:

$$\text{Geburtenrate} = 0,2$$

$$\text{Sterberate} = 0,1$$

- III.b **Erklären** Sie, weshalb es nach diesem Modell zu einem starken Wachstum der Population kommen muss.

(6 BE)

- III.c **Stellen Sie dar**, warum für biologische Systeme exponentielles Wachstum für große Zeiträume keine korrekte Modellierung darstellen kann, und **beschreiben** Sie ein alternatives Wachstumsmodell, das die Populationsentwicklung besser wiedergibt.

(6 BE)

Die Grauhörnchen verdrängen die einzigerischen Eurasischen Eichhörnchen alleine durch ihre Anwesenheit, da diese in ihrem Revier keine Konkurrenz mögen. In Großbritannien kommt hinzu, dass 60% der Grauhörnchen dort mit dem Parapoxvirus infiziert sind, an dem sie nicht erkranken, das sie aber weitergeben und das für die roten Eichhörnchen tödlich ist.

- III.d
- **Entwickeln** Sie eine Veränderung für das obige Modell, so dass die Größe der beiden Populationen der Eurasischen Eichhörnchen und der Grauhörnchen auf der britischen Insel simuliert werden kann.
 - **Skizzieren** Sie ein Modelldiagramm und ein Zeitdiagramm, die zu diesen Annahmen passen.
 - **Geben** Sie passende Simulationsgleichungen **an**.

(13 BE)

Ist eine neue Art erst einmal etabliert, ist es oft schwierig oder gar unmöglich, sie wieder auszurotten. Die Grauhörnchen verdrängen zum einen die einheimische Art, zum anderen richten sie höhere Schäden an Bäumen an. In Großbritannien ist die Art inzwischen so dominant und so weit verbreitet, dass es aussichtslos scheint, sie wieder auszurotten. In Norditalien sind Versuche, die Tiere einzufangen und zu töten, durch einen mehrjährigen Gerichtsprozess gegen dieses Vorgehen so verzögert worden, dass es dort nun ebenfalls kaum noch möglich ist.

- III.e **Untersuchen** Sie, ob eine Ausrottung der Grauhörnchen in Norditalien weiter versucht werden sollte.

(6 BE)

Um eine Ausbreitung in Deutschland zu verhindern, dürfen Grauhörnchen hier von Privatpersonen weder gezüchtet noch angeboten, abgegeben oder zur Weitergabe vorrätig gehalten werden.

Trotzdem stellt sich die Frage, ob die einheimischen Eichhörnchen irgendwann von Grauhörnchen verdrängt werden, wenn sie sich von Italien aus ausbreiten. Eine Antwort kann auf der Grundlage von Simulationen gegeben werden, wie sie u. a. an der Universität Turin (Norditalien) erarbeitet werden.

Mit einem Modell wie oben kann die Population aber nur innerhalb einer geographisch fest umrissenen Region, etwa einer Insel, gut simuliert werden.

III.f Entwickeln Sie eine Vorgehensweise, um die Ausbreitung der norditalienischen Grauhörnchen so mit Dynasys zu simulieren, dass man zu einer Vorhersage kommen kann, ob und wenn ja, wann sie in Hamburg ankommen werden.

(9 BE)

Aufgabe III: Eichhörnchen (Consideo-Version)**50 BE****Schwerpunkt: Simulation dynamischer Systeme**

In Hamburg ist das Eurasische Eichhörnchen (*Sciurus vulgaris*) weit verbreitet. Es hat eine rötliche Fellfärbung, wobei selbst die Geschwister innerhalb eines Wurfs unterschiedlich dunkles Fell bis hin zu schwarz haben können. Eichhörnchen sind Einzelgänger.

In den USA ist das Grauhörnchen (*Sciurus carolinensis*) heimisch. Die Tiere sind etwas größer, haben ein graues Fell und leben gesellig in Gruppen. In Großbritannien und in Norditalien wurden Grauhörnchen ausgesetzt. Sie verbreiten sich nun stark und verdrängen die einheimischen roten Eichhörnchen, so dass in England und Wales inzwischen fast ausschließlich Grauhörnchen leben und nur in Schottland noch rote Eichhörnchen zu finden sind. Aktuell geht man von 140.000 roten und 2.500.000 grauen Eichhörnchen in Großbritannien aus.

Die Grauhörnchen haben den Eurasischen Eichhörnchen gegenüber den Vorteil, dass sie Eicheln gut verdauen können und ihre Futterverstecke im Winter sicherer wiederfinden. Rote Eichhörnchen kommen dagegen besser mit dem Futterangebot in Nadelwäldern zurecht.



Abb. 1: Rotes Eichhörnchen mit einer Walnuss,
Wikimedia-Nutzer Sven-Christoph Petersen
CC-BY 3.0



Abb. 2: Grauhörnchen im Park Bunhill Fields (London),
Wikimedia-Nutzer HH58, CC-BY-SA 4.0

- III.a
- **Beschreiben** Sie die Wachstumsform exponentielles Wachstum an einem selbst gewählten Beispiel und gehen Sie dabei auf die Voraussetzungen für ihr Auftreten ein.
 - **Zeichnen** Sie ein passendes Modelldiagramm.
 - **Skizzieren** Sie den Verlauf der Werte des Bestandsfaktors in einem Simulationsdiagramm.
- (10 BE)

Das folgende Modelldiagramm beschreibt zusammen mit den Simulationsgleichungen ein biologisches System.

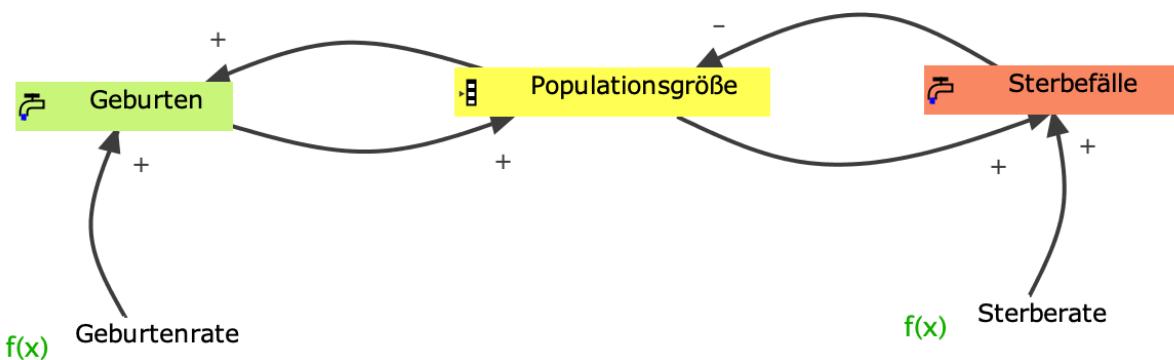


Abb. 3: Modelldiagramm

Zustandsgleichungen:

$$[\text{Populationsgröße.neu}] = [\text{Populationsgröße.alt}] + dt * ([\text{Geburten}] - [\text{Sterbefälle}])$$

$$\text{Startwert } [\text{Populationsgröße}] = 100$$

Zustandsänderungen:

$$[\text{Geburten}] = [\text{Geburtenrate}] * [\text{Populationsgröße}]$$

$$[\text{Sterbefälle}] = [\text{Sterberate}] * [\text{Populationsgröße}]$$

Parameter:

$$[\text{Geburtenrate}] = 0.2$$

$$[\text{Sterberate}] = 0.1$$

- III.b **Erklären** Sie, weshalb es nach diesem Modell zu einem starken Wachstum der Population kommen muss.

(6 BE)

- III.c **Stellen Sie dar**, warum für biologische Systeme exponentielles Wachstum für große Zeiträume keine korrekte Modellierung darstellen kann, und **beschreiben** Sie ein alternatives Wachstumsmodell, das die Populationsentwicklung besser wiedergibt.

(6 BE)

Die Grauhörnchen verdrängen die einzigerischen Eurasischen Eichhörnchen alleine durch ihre Anwesenheit, da diese in ihrem Revier keine Konkurrenz mögen. In Großbritannien kommt hinzu, dass 60% der Grauhörnchen dort mit dem Parapoxvirus infiziert sind, an dem sie nicht erkranken, das sie aber weitergeben und das für die roten Eichhörnchen tödlich ist.

- III.d
- **Entwickeln** Sie eine Veränderung für das obige Modell, so dass die Größe der beiden Populationen der Eurasischen Eichhörnchen und der Grauhörnchen auf der britischen Insel simuliert werden kann.
 - **Skizzieren** Sie ein Modelldiagramm und ein Simulationsdiagramm, die zu diesen Annahmen passen.
 - **Geben** Sie passende Simulationsgleichungen **an**.

(13 BE)

Ist eine neue Art erst einmal etabliert, ist es oft schwierig oder gar unmöglich, sie wieder auszurotten. Die Grauhörnchen verdrängen zum einen die einheimische Art, zum anderen richten sie höhere Schäden an Bäumen an. In Großbritannien ist die Art inzwischen so dominant und so weit verbreitet, dass es aussichtslos scheint, sie wieder auszurotten. In Norditalien sind Versuche, die Tiere einzufangen und zu töten, durch einen mehrjährigen Gerichtsprozess gegen dieses Vorgehen so verzögert worden, dass es dort nun ebenfalls kaum noch möglich ist.

- III.e **Untersuchen** Sie, ob eine Ausrottung der Grauhörnchen in Norditalien weiter versucht werden sollte.

(6 BE)

Um eine Ausbreitung in Deutschland zu verhindern, dürfen Grauhörnchen hier von Privatpersonen weder gezüchtet noch angeboten, abgegeben oder zur Weitergabe vorrätig gehalten werden.

Trotzdem stellt sich die Frage, ob die einheimischen Eichhörnchen irgendwann von Grauhörnchen verdrängt werden, wenn sie sich von Italien aus ausbreiten. Eine Antwort kann auf der Grundlage von Simulationen gegeben werden, wie sie u. a. an der Universität Turin (Norditalien) erarbeitet werden.

Mit einem Modell wie oben kann die Population aber nur innerhalb einer geographisch fest umrissenen Region, etwa einer Insel, gut simuliert werden.

III.f **Entwickeln** Sie eine Vorgehensweise, um die Ausbreitung der norditalienischen Grauhörnchen so mit dem Consideo Modeler zu simulieren, dass man zu einer Vorhersage kommen kann, ob und wenn ja, wann sie in Hamburg ankommen werden.

(9 BE)