

WINTERSEMESTER 2022/23

Studiengang/Abschluss: Wirtschaftsinformatik / B.Sc.
Prüfungsfach: Grundlagen der Programmierung
Prüfungsnummer: 5701
Prüfer: Prof. Dr. Andreas Biesdorf
Anzahl der Arbeitsblätter: 9 Seiten (ohne Deckblätter)
Bearbeitungszeit: 90 Minuten

Mit Bleistift oder in roter Farbe geschriebene Ausführungen werden nicht gewertet!

Zugelassene Hilfsmittel:

Eine einseitig handbeschriebene DIN A4-Seite Python-Befehle oder -Syntax (z.B. die Syntax einer for- oder while-Schleife), jedoch keine Theorie oder Strukturen konkreter Implementierungen kompletter Aufgaben.

Matrikelnummer:

--	--	--	--	--	--

Datum der Klausur:

.....

Platznummer:

Punkte: _____ von 90 möglichen

Note: _____

Handzeichen des Prüfers: _____

Hinweise zu Klausuren im FB Wirtschaft

- Mit Aufnahme der Prüfung bringen Sie zum Ausdruck, dass Sie sich für prüfungstauglich halten. Ein Rücktritt von der Klausur während der Bearbeitungszeit ist damit nur noch in ganz besonderen Ausnahmefällen möglich.
 - Sollten Sie zu Beginn der Bearbeitungszeit prüfungstauglich sein, dann aber während der Bearbeitungszeit prüfungsuntauglich werden, müssen Sie die Klausur abbrechen, Ihre Prüfungsuntauglichkeit den Aufsichtführenden anzeigen und schnellstmöglich, in jedem Fall aber fristgerecht für ein ärztliches Attest sorgen.
 - Mit der Abgabe Ihrer Klausur bringen Sie zum Ausdruck, dass Sie sich während der gesamten Bearbeitungszeit für prüfungstauglich gehalten haben. Ein Rücktritt ist dann nicht mehr möglich.
 - Jacken und Taschen sind in ausreichender Entfernung vom Platz zu deponieren.
 - Die Bearbeitungszeit beginnt erst, wenn alle Klausuren ausgeteilt sind.
 - Es sind nur die zugelassenen Hilfsmittel zu verwenden.
 - Das Verlassen des Raumes für Toilettengänge während der Klausur ist nur nach Meldung und Bestätigung der Aufsicht gestattet. Die vorübergehende Abwesenheit wird im Protokoll vermerkt.
 - Es sind keine anderen als die ausgegebenen Bearbeitungsblätter zu benutzen.
 - Auf dem Tisch sind nur Schreibutensilien erlaubt, keine Mäppchen.
 - Jeder Teilnehmer muss sich mit einem Studierendenausweis oder Lichtbildausweis ausweisen.
 - Geräusche und Störungen jeder Art sind zu vermeiden.
 - Die Klausurensätze sind grundsätzlich nicht zu öffnen. Wer den Klausurensatz öffnet, trägt dafür Sorge, dass die Blätter (innerhalb der Bearbeitungszeit) sortiert und neu geheftet werden. Das Risiko des Verlustes von einzelnen Blättern trägt der Studierende!
 - Es ist nicht erlaubt, Aufgabenstellungen auf gesondertes Papier abzuschreiben.
 - Mobiltelefone, Smartphones und Smartwatches sind nicht erlaubt!
 - Alle Studierenden bleiben auf ihren Plätzen und verhalten sich ruhig, bis die Klausuren vollständig eingesammelt sind. Folgen Sie den Anweisungen der Aufsichtsführenden.
 - Bei Täuschung oder versuchter Täuschung wird mit „nicht ausreichend“ bewertet.
-

Die Klausur besteht aus 8 Aufgaben zu den in der Lehrveranstaltung behandelten Themenbereichen. Die Punktzahl ist bei jeder Aufgabe mit angegeben.

Aufgabe	Mögliche Punkte	Erreichte Punkte
A1	9	
A2	6	
A3	6	
A4	6	
A5	8	
A6	20	
A7	10	
A8	25	
Summe	90	
Evtl. Bonus		

Bitte dokumentieren Sie die Antworten zu allen Aufgaben in der entsprechenden **Textbox** auf dem Blatt oder der **entsprechend benannten Datei** im Ordner **C:\temp\Klausur**.

Bitte aktualisieren Sie in jeder Datei immer zunächst den Header mit Ihrer Matrikelnummer und Ihrem Namen!

Aufgabe 1

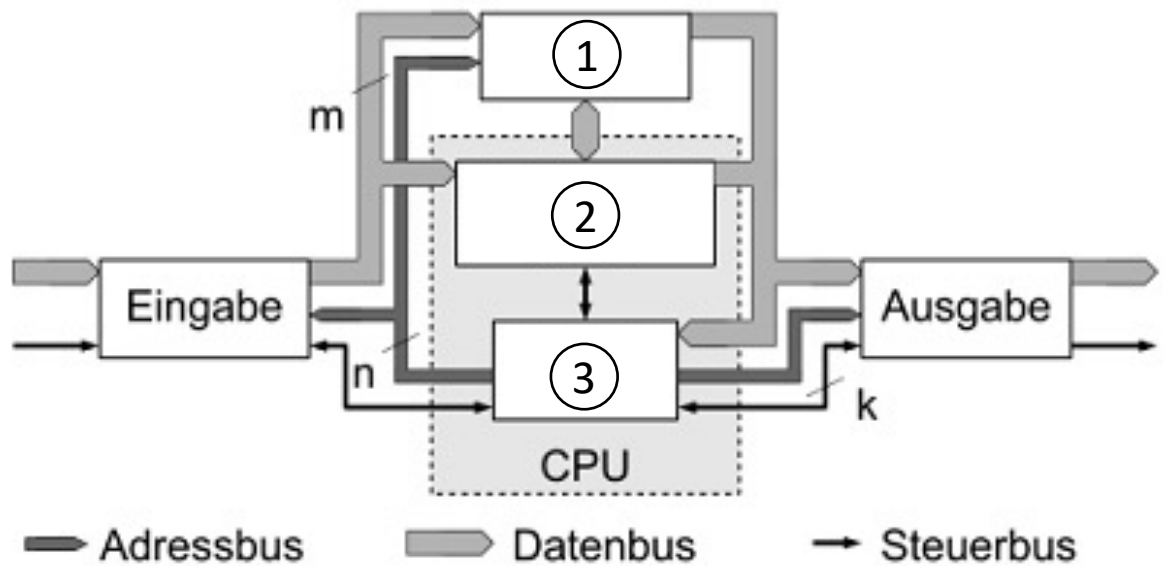
- 1) Was versteht man unter „Computational Thinking“? Bitte erläutern Sie das Konzept anhand eines Alltagsproblems. (2 Punkte)

- 2) Was ist der grundlegende Unterschied zwischen dem Problem- und dem Lösungsraum? Grenzen Sie die beiden Begriffe stichpunktartig voneinander ab. (1 Punkt)

Kreuzen Sie an: Was ist eine Aufgabe im Rahmen des Problemraums und was ist eine Aufgabe im Rahmen des Lösungsraums? (3 Punkte)

Aufgabe	Problemraum	Lösungsraum
Entwicklung unter Verwendung einer bestimmten Programmiersprache (z.B. Python)		
Erstellung des Domänenmodells		
Entscheidung für eine bestimmte Cloud-Infrastruktur (z.B. AWS)		
Analyse der Komplexität einer zu lösenden Aufgabe		
Implementierung eines Prototyps		
Dokumentation der nicht-funktionalen Anforderungen		

- 3) Benennen und erläutern Sie stichwortartig die drei im folgenden Bild fehlenden Komponenten der Von-Neumann Rechnerarchitektur (3 Punkte)



Nummer	Name der Komponente und stichwortartige Erläuterung
1	
2	
3	

Aufgabe 2

Schreiben Sie folgendes Programm mit einem iterativen Ansatz: Geben Sie die Fakultät einer nicht negativen ganzen Zahl aus. (6 Punkte)

Zur Erinnerung: Die Fakultät einer nicht negativen ganzen Zahl ist das Produkt aller positiven ganzen Zahlen kleiner oder gleich der Zahl:

$$n! = 1 \cdot 2 \cdot 3 \cdots n = \prod_{k=1}^n k$$

Hierbei gilt jedoch

$$0! = 1$$

Einige Beispiele:

```
1! = 1
2! = 1 × 2 = 2
3! = 1 × 2 × 3 = 6
4! = 1 × 2 × 3 × 4 = 24
```

Aufgabe 3

Schreiben Sie einen einfachen Rechentrainer für Additions- und Subtraktionsaufgaben. (6 Punkte)

- Das Programm soll dem Nutzer fünf zufällige Additions- oder Subtraktionsaufgaben mit Zahlen im Zahlenbereich von 1 bis 10 stellen und das Ergebnis jeweils prüfen.
- Bei fehlerhaften Eingaben soll ein entsprechender Hinweis erfolgen und eine neue Aufgabe gestellt werden (keine Wiederholung derselben Aufgabe)
- Die Dokumentation der richtigen Ergebnisse soll durch einen einfachen Fortschrittsbalken zu Beginn der Zeile dokumentiert werden, bei dem korrekt gerechnete Ergebnisse mit einem „X“ markiert werden und noch zu rechnende Aufgaben mit „_“.
- Die Wahl des Aufgabentyps (Addition vs. Subtraktion) soll auch zufällig sein.

Hilfestellung:

Binden Sie in Ihrem Programm zu Beginn die Bibliothek `random` ein und nutzen die Funktion `randint(untere_grenze, obere_grenze)`, um eine Zufallszahl im angegebenen Intervall zu erzeugen. Die Grenzen sind jeweils Teil des Intervalls.

```
# Einbinden der Bibliothek
import random

# Erzeugt eine Zufallszahl im Intervall 1 <= zufallszahl <= 10
zufallszahl = random.randint(1, 10)
```

Im Folgenden sehen Sie zwei Beispielausgaben, die durch das Programm erzeugt werden könnten.
Achtung: die **gelb markierten** Zahlen sind jeweils Nutzereingaben!

Beispielausgabe 1 ohne Fehler in der Eingabe (Nutzereingaben in **gelb**):

Rechne die folgenden fünf Aufgaben

[____] 5 + 2 = **7**

[X____] 2 + 1 = **3**

[XX____] 6 - 4 = **2**

[XXX____] 7 - 2 = **5**

[XXXX_] 8 - 4 = **4**

[XXXXX] Gut gerechnet!

Beispielausgabe 2 mit Fehlern in der Eingabe (Nutzereingaben in **gelb**):

Rechne die folgenden fünf Aufgaben

[____] 4 - 2 = **2**

[X____] 10 - 2 = **8**

[XX____] 3 - 2 = **3**

Falsch, korrekte Antwort wäre: 1

[XX____] 1 + 6 = **7**

[XXX____] 7 - 4 = **3**

[XXXX_] 3 + 7 = **10**

[XXXXX] Gut gerechnet!

Aufgabe 4

Geben Sie auf der Kommandozeile eine vereinfachte Raute mit Sternen aus. Hierfür wird der Nutzer um Angabe der maximalen Breite der Raute gebeten. Die minimale Breite beträgt 5. Zur Vereinfachung dürfen Sie davon ausgehen, dass nur ungerade Zahlen vorgegeben werden. (**6 Punkte**)

Im Folgenden drei Beispielausgaben:

Raute mit Breite 5:	Raute mit Breite 7:	Raute mit Breite 9:
<pre> X XXX XXXXX XXX X </pre>	<pre> X XXX XXXXX XXXXXX XXXXX XXX X </pre>	<pre> X XXX XXXXX XXXXXX XXXXXXXX XXXXXXXXXXX XXXXXX XXXXX XXX X </pre>

Aufgabe 5

Die „Caesar Chiffre“ ist eine einfache Verschlüsselungsmethode für Texte. Hierbei wird jedes Zeichen im Alphabet (ohne Sonderzeichen) mit dem um n Zeichen verschobenen Zeichen ersetzt. Dies erfolgt zyklisch, d.h. Verschiebungen über das Ende des Alphabets hinaus beginnen wieder von vorne.

Beispiel für Verschiebung um 3 Zeichen:

x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	Y	z	...
			↓																									↓	
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	B	c	...

- ,a' wird durch ,d' ersetzt, da ,d' im Alphabet drei Zeichen nach ,a' kommt
- ,z' wird durch ,c' ersetzt, da ,c' im (fortgesetzten) Alphabet drei Zeichen nach ,a' kommt

Zur Vereinfachung können Sie Umlaute, Sonderzeichen sowie Groß- und Kleinschreibung ignorieren und davon ausgehen, dass lediglich die 26 Buchstaben des Alphabets als Kleinbuchstaben vorkommen.

Implementieren Sie nun folgende zwei Funktionen: (8 Punkte)

- 1) *Kodieren* eines einfachen Wortes mit Verschiebung um einen wählbaren Wert
- 2) *Decodieren* eines kodierten Wortes mit Verschiebung um einen wählbaren Wert

Rufen Sie die beiden Funktionen testweise hintereinander auf, um das Funktionieren zu demonstrieren.

Beispielausgabe 1 (Nutzereingaben in **gelb**):

Um wie viele Stellen sollen die Zeichen im Wort verschoben werden? **1**

Welches Wort soll verschlüsselt werden? **testwort**

Ergebnis: uftuxpsu

Test der Entschlüsselung mit dem Wort uftuxpsu

Ergebnis: testwort

Beispielausgabe 2 (Nutzereingaben in **gelb**):

Um wie viele Stellen sollen die Zeichen im Wort verschoben werden? **3**

Welches Wort soll verschlüsselt werden? **wirtschaftsinformatik**

Ergebnis: zluwvfkdiwvlqirupdwln

Test der Entschlüsselung mit dem Wort zluwvfkdiwvlqirupdwln

Ergebnis: wirtschaftsinformatik

Beispielausgabe 3 (Nutzereingaben in **gelb**):

Um wie viele Stellen sollen die Zeichen im Wort verschoben werden? **5**

Welches Wort soll verschlüsselt werden? **kraftfahrzeughaftpflichtversicherung**

Ergebnis: pwfkykfmwejjlmfkyukqnhmyajwxnhmjwzsl

Test der Entschlüsselung mit dem Wort pwfkykfmwejjlmfkyukqnhmyajwxnhmjwzsl

Ergebnis: kraftfahrzeughaftpflichtversicherung

Aufgabe 6

Die Fläche eines Kreises ergibt sich aus dem Radius durch folgende Gleichung:

$$A = r^2 \cdot \pi$$

Der Nutzer möchte nun zu einer gegebenen Fläche A den zugehörigen Radius r bestimmen. Hierbei soll in der Lösung jedoch auf die Verwendung des Wurzel-Operators in Python verzichtet werden.

Schreiben Sie daher bitte folgende zwei Funktionen:

- 1) Bestimmung des Radius r zu einer Fläche A mit Hilfe der linearen Suche (8 Punkte)
- 2) Bestimmung des Radius r zu einer Fläche A mit Hilfe der binären Suche (10 Punkte)

Erstellen Sie einen einfachen Benchmark: Wie viele Iterationen benötigen Ihre zwei Implementierungen? (2 Punkte)

Aufgabe 7

Palindrome sind Zeichenketten, die von vorne und von hinten gelesen das identische Wort ergeben. Beispiele sind Wörter wie *Lagerregal*, *Elle*, *Ebbe*, *Ehe*, *Neffen*, *Retsina-Kanister*, *Reliefpfeiler* oder auch *Rentner*.

Schreiben Sie bitte eine *rekursive* Funktion, die prüft, ob es sich (ohne Berücksichtigung der Groß- und Kleinschreibung) bei einer Zeichenkette um ein Palindrom handelt.

(10 Punkte)

Aufgabe 8

Ein Fahrradverleih hat mehrere Fahrräder. Fahrräder haben eine eindeutige Nummer, eine Größe, eine Marke und einen Anschaffungspreis. Es gibt unterschiedliche Typen von Fahrrädern. So bietet der Fahrradverleih aktuell Rennräder, Mountainbikes und Elektroräder an. Weiterhin hat der Fahrradverleih Kunden und Mitarbeiter. Beide sind Personen und haben einen Namen, eine Identifikationsnummer sowie eine Telefonnummer. Mitarbeiter haben zusätzlich ein Gehalt und einen Dienstgrad. Der Fahrradverleih ermöglicht es Kunden, Fahrräder zu leihen. Eine Leihe hat einen Start- und einen End-Zeitpunkt sowie einen Preis und bezieht sich auf ein Fahrrad.

- 1) Modellieren Sie die o.g. Zusammenhänge als UML-Modell (10 Punkte)

- 2) Implementieren Sie die grobe *Struktur* der Klassen Ihres UML-Modells (10 Punkte)
- 3) Implementieren Sie folgende zwei Funktionen und führen diese beispielhaft aus: (5 Punkte)
 - a. Einstellen eines Fahrrads im Fahrradverleih
 - b. Ausleihen eines Fahrrads durch einen Kunden

Viel Erfolg bei der Bearbeitung der Aufgaben!

Appendix: English translations of the texts of the questions

Please note: In case of doubt, the German exam serves as reference!

Aufgabe 1 – Translations

- 1) What is meant with the term „Computational Thinking“? Please explain the concept using a real-life problem
- 2) What is the fundamental difference between problem- and solution space? Outline the difference between the terms in bullet points.
- 3) Please set checkmarks: Which task is a task in the context of the problem space and which tasks are tasks in the context of the solution space?

Task	Problem space	Solution space
Development using a particular programming language (e.g., Python)		
Creation of the domain model		
Decision for a particular cloud infrastructure (e.g., AWS)		
Analysis of the complexity of a task to be solved		
Implementation of a prototype		
Documentation of the non-functional requirements		

- 4) Please name and briefly explain the three missing components of the Von-Neumann computer architecture in the following image.

Aufgabe 2 – Translations

Please write the following program with an iterative approach: Compute the factorial of a non-negative integer.

As a reminder: The factorial of a non-negative integer is the product of all positive integers less than or equal to that number:

$$n! = 1 \cdot 2 \cdot 3 \cdots n = \prod_{k=1}^n k$$

with

$$0! = 1$$

Aufgabe 3 – Translations

Please write a simple arithmetic trainer for addition and subtraction problems:

- The program should present the user with five random addition or subtraction problems with numbers in the range of 1 to 10 and check the result each time.
- If the input is incorrect, a corresponding warning should be given, and a new problem should be presented (not repeating the same problem)
- The documentation of correct results should be recorded by a simple progress bar at the beginning of the line, where correctly calculated results are marked with an "X" and problems yet to be calculated are marked with "_".
- The choice of problem type (addition vs subtraction) should also be random.

Aufgabe 4 – Translations

Please output a simplified diamond with stars on the command line. The user will be asked for the maximum width of the diamond, with a minimum width of 5. For simplicity, it can be assumed that only odd numbers are provided.

Aufgabe 5 – Translations

The "Caesar Cipher" is a simple encryption method for texts. In this method, each character in the alphabet (without special characters) is replaced with the character shifted by n characters. This is done cyclically, i.e., shifts beyond the end of the alphabet start again from the beginning.

Example with a shift of 3:

x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	Y	z	...
			↓																									↓	
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	B	c	...

- ,a' is replaced by ,d', since ,d' follows 'a' after three characters
- ,z' is replaced by ,c', since ,c' follows ,z' after three characters when continued cyclically

For simplification, you can ignore umlauts, special characters, and upper- and lower-case letters and assume that only the 26 letters of the alphabet appear as lowercase letters.

Please implement the following two functions:

- 3) *Encode* a simple word by shifting by an offset of choice
- 4) *Decode* a coded word by shifting by an offset of choice

Call both functions sequentially to demonstrate their functioning.

Aufgabe 6 – Translations

The area of a circle can be computed using the radius with the following equation:

$$A = r^2 \cdot \pi$$

A user wants to determine the radius r for an area A . Please do not use the root-operator in Python.

Please implement the following two functions:

- 1) Determine the radius r for the area A of a circle using a linear search algorithm
- 2) Determine the radius r for the area A of a circle using a binary search algorithm

Please create a simple benchmark: how many iterations do the two implementations need in comparison?

Aufgabe 7 – Translations

Palindromes are character strings that give the same word when read from front to back and from back to front. Examples are words like "Lagerregal", "Elle", "Ebbe", "Ehe", "Neffen", "Retsina-Kanister", "Reliepfweiler" or "Rentner".

Please write a *recursive* function that checks if a character string is a palindrome (without considering capitalization).

Aufgabe 8 – Translations

A bicycle rental has multiple bicycles. Bicycles have a unique number, a size, a brand, and a purchase price. There are different types of bicycles currently offered by the bicycle rental, including road bikes, mountain bikes, and electric bikes. Additionally, the bicycle rental has customers and employees. Both are people and have a name, an identification number, and a telephone number. Employees also have a salary and a rank. The bicycle rental allows customers to rent bicycles. A rental has a start and end time, a price, and refers to a bicycle.

- 1) Please model the relationships in form of a UML model
- 2) Please implement the structure of the classes of the UML model
- 3) Implement the following two functions and run them:
 - a. Add a bike to the bicycle rental
 - b. Rent a bike by a customer