

Erläutere die folgenden Begriffe:

(a) Programmierung

A full-page view of a blank sheet of graph paper. The grid consists of small squares formed by thin gray lines. There are 20 columns and 15 rows of squares. The entire page is covered by this grid, with no margins or other markings.

(b) Programmiersprache

[illegible]

(c) **Kommentare (Kopfkommentar, normaler Kommentar und Docstring)**

[illegible]



Aufgabe 2 (Ausgaben und Datentypen)

Erläutere die Ausgaben der folgenden Befehle.

Gehe dabei insbesondere auf die Unterschiede in den Argumenten der `print()`-Funktion ein.

1

```
# print("Viel Erfolg!")
```

1

```
print(2)
```

1

```
for zwei in range(2):  
    print("zwei")
```

2

1

```
print("2"+"18")
```

1

```
def myFunction():  
    print("Hallo Welt")
```

2



Aufgabe 3 (Elemente der Programmierung)

- (a) Stelle kurz dar, zu welchem Zweck im Rahmen der Programmierung die Kontrollstruktur Schleife eingesetzt wird und zwischen welchen beiden grundlegenden Arten man dabei unterscheidet.

- (b) In einem Programm benötigst du an unterschiedlichen Stellen eine bestimmte Folge von Befehlen. Es ist möglich, diese Befehle an allen entsprechenden Stellen in das Programm zu schreiben. Wir haben noch eine zweite Möglichkeit kennengelernt, dies zu realisieren. Erläutere diese kurz und stelle mindestens einen Vorteil dieser Realisierung dar.

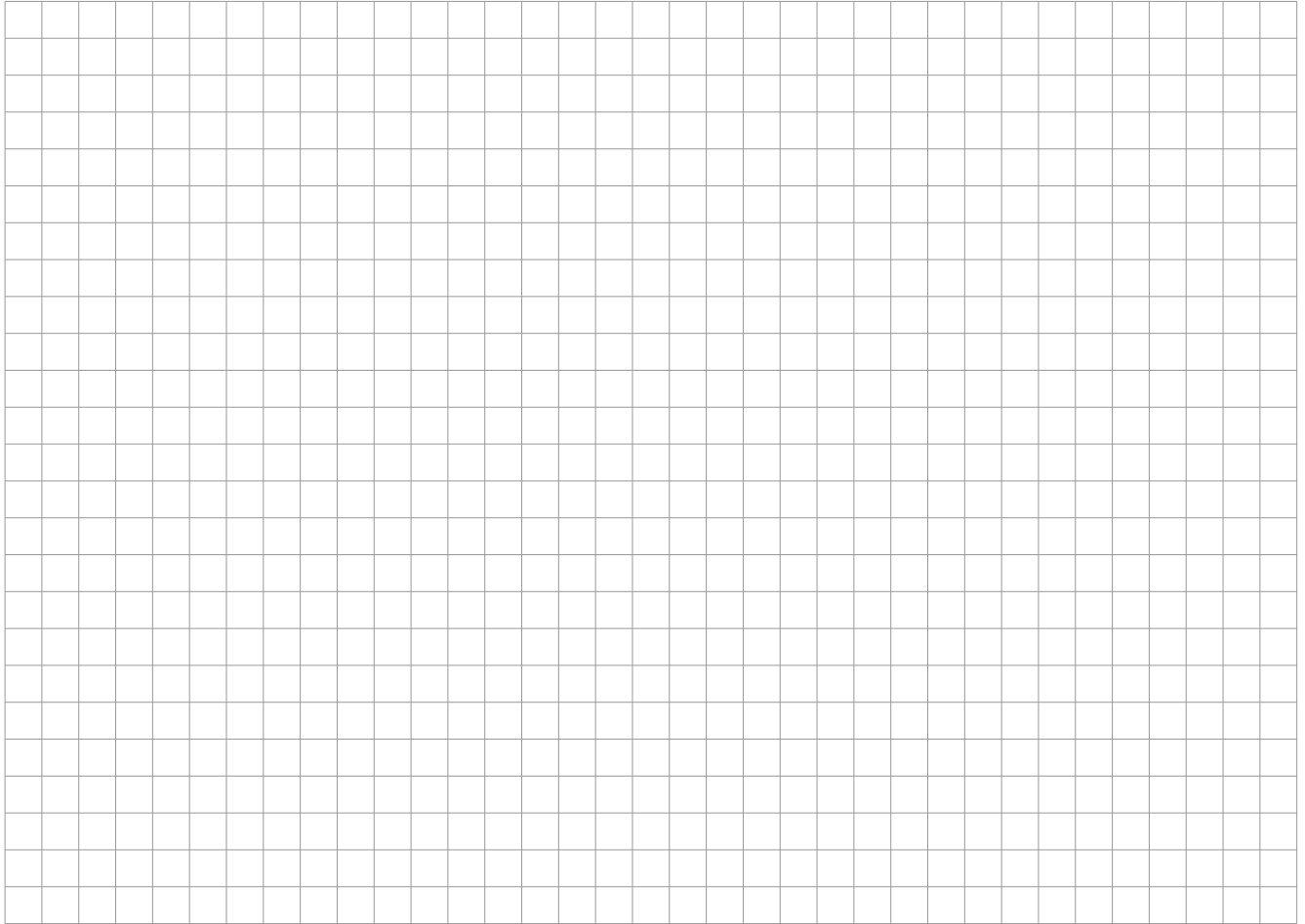
A full-page sheet of graph paper featuring a uniform grid of thin, light gray lines on a white background. The grid consists of small squares covering the entire area.

[illegible]

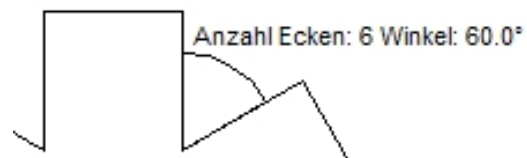
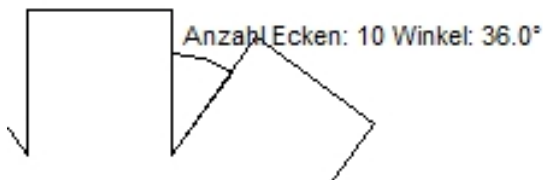
A full-page view of a blank sheet of graph paper. The grid consists of thin, light gray horizontal and vertical lines forming small squares across the entire page. There are no margins, text, or other markings on the paper.

(b) Um entsprechende Zeichnungen erstellen zu können, müssen wir u.a. ein Objekt der Klasse Turtle erzeugen. Treffe auch die dazu nötigen Vorbereitungen. Lege ebenfalls Variablen für die Zahnanzahl und die zugehörige Abmessung an. Die zugehörigen Variablen und Werte darfst du frei wählen.

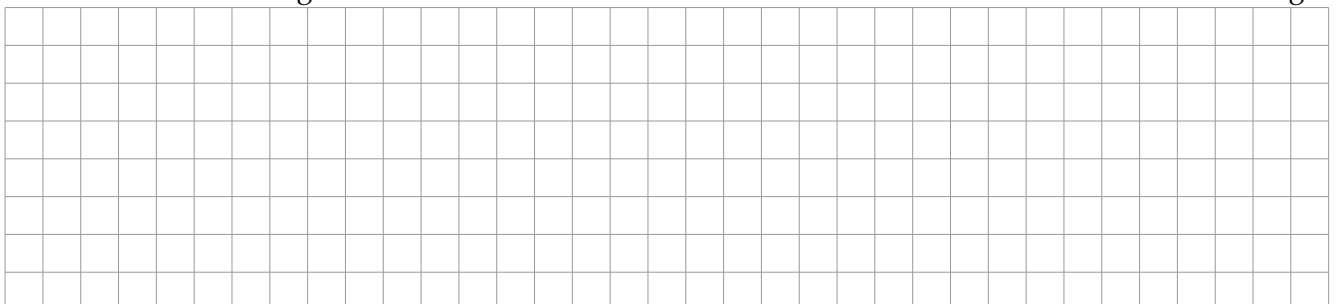
- (c) Für das Zeichnen eines einzelnen Zahnes soll eine entsprechende Funktion geschrieben werden. Die Turtle soll am Ende wieder in die ursprüngliche Richtung gedreht sein. Schreibe die entsprechende Funktion inkl. Funktionskopf etc.

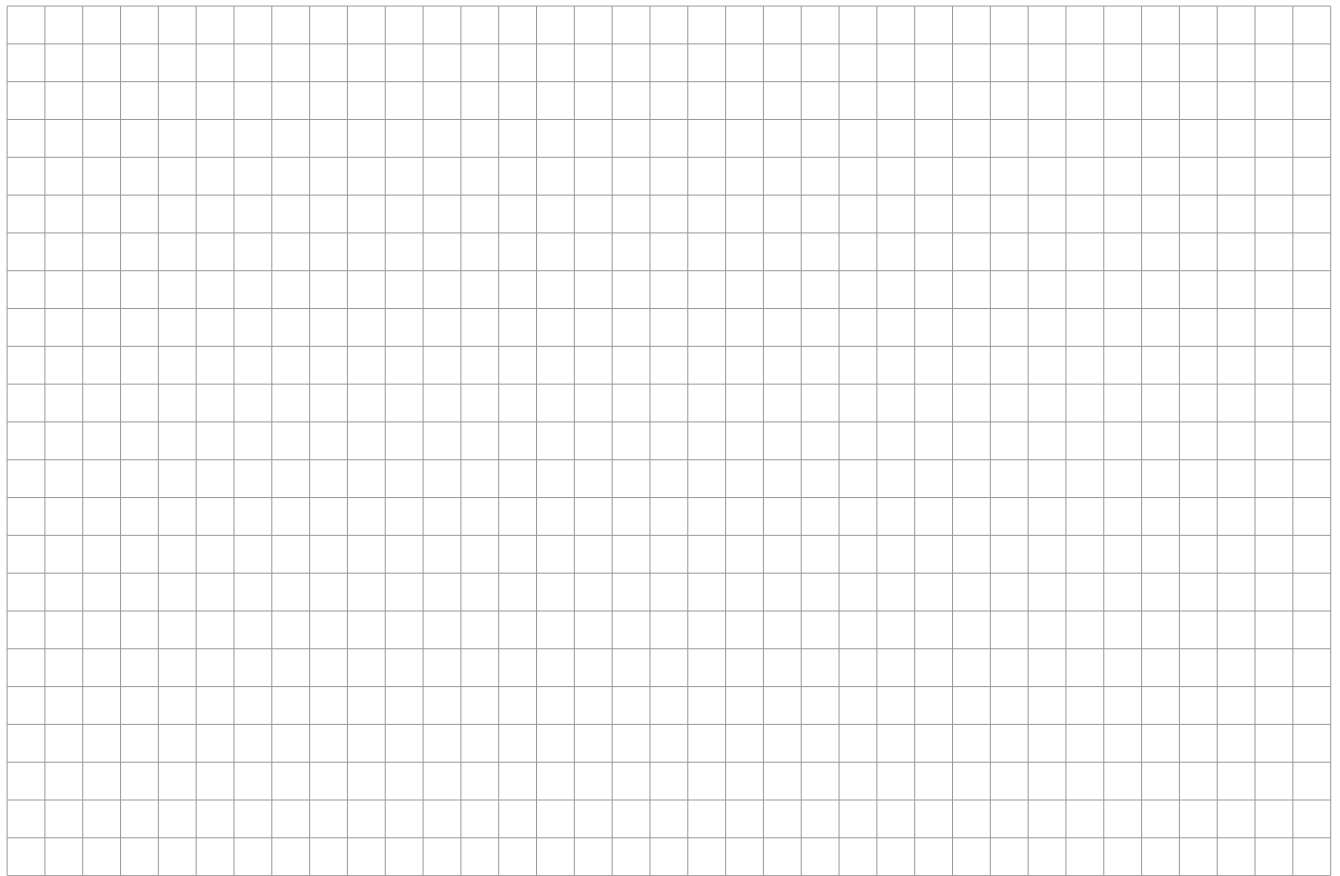


- (d) Abschließend muss nun der Programmausschnitt geschrieben werden, der mittels der programmierten Funktion und einer weiteren geeigneten Kontrollstruktur das gesamte Zahnrad zeichnet. Zur Bestimmung des Drehwinkels zwischen zwei Zähnen helfen dir die beiden Abbildungen. Achte darauf, dass sich das Turtle-Objekt in die richtige Richtung dreht.

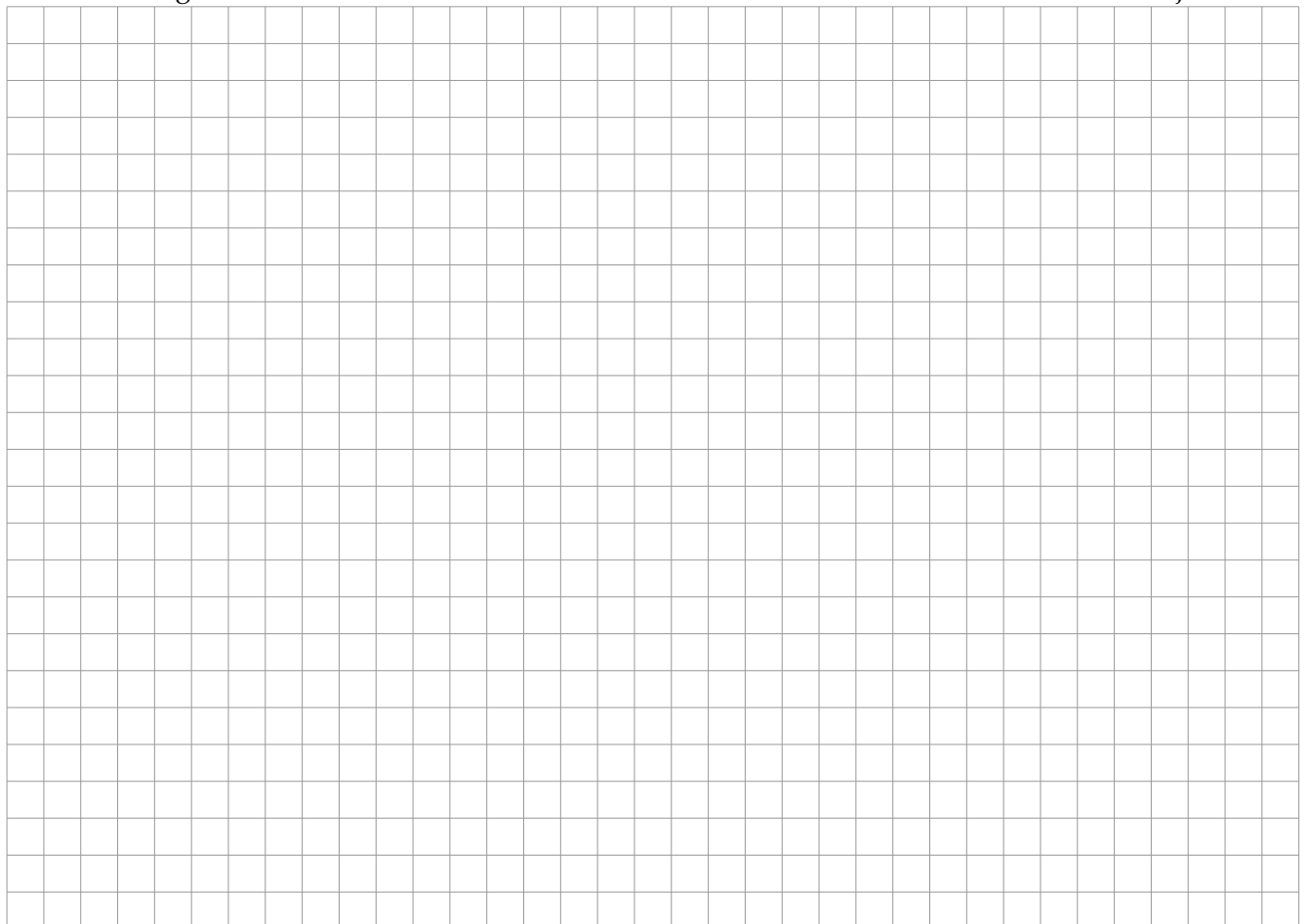


Hinweis: Fertige zur Unterstützung eine Skizze an, in der du bekannte Winkel markierst. Überlege genau, in welche Richtung die Turtle nach dem Zeichnen des ersten Zahnes gedreht ist und in welche Richtung eine Drehung erfolgen muss, damit der nächste Zahn gezeichnet werden kann. Ergänze zur weiteren Unterstützung Hilfslinien. Auf der nächsten Seite findest du weiteren Platz zur Bearbeitung.





- (e) Erweitere den letzten Aufgabenteil so, dass das Zahnrad ausgefüllt wird. Nutze dazu die Methoden `begin_fill()` und `end_fill()`, die jedes Turtle-Objekt bereitstellt. Ferner soll das Objekt der Klasse Turtle am Programmende unsichtbar werden. Rufe dazu die Methode `hideturtle()` des Objektes auf.





Aufgabe 5 (Single-Choice Fragen)

Entscheide, ob die folgenden Aussagen richtig oder falsch sind.

	r	f	Aussage [Objekte etc.]
	☐	☐	Ein Objekt ist der Bauplan einer Klasse.
	☐	☐	Eine grundlegende Art der Programmierung ist die sogenannte variablen-orientierte Programmierung.
	☐	☐	Bei der Objektorientierung versucht man die Realität möglichst detailgetreu in ein Programm bzw. dessen Entwurf umzusetzen.
	☐	☐	Attribute von Objekten kann man nicht verändern.
	☐	☐	Mit Methoden bezeichnen die Informatiker Fähigkeiten von Objekten.
	☐	☐	Bevor man ein Objekt verwenden kann, muss man es erzeugen.
	☐	☐	Sicherheitshalber sollte man alle Klassen laden, denn man kann ja nie wissen, was man später im Programm noch alles benötigt.
	☐	☐	Jedes Objekt der Klasse Mensch hat ein Attribut „Gewicht“.
	r	f	Aussage [Schleifen und Funktionen etc.]
	☐	☐	Mittels Kontrollstrukturen wird es möglich, den Programmablauf zu steuern.
	☐	☐	Das reservierte Wort <code>def</code> benötigt man in einem Schleifenkopf.
	☐	☐	Ein Funktionskopf muss immer eingerückt werden.
	☐	☐	Mittels eines sog. Docstrings kommentiert man selbst geschriebene Funktionen.
	☐	☐	Anweisungen in Schleifen- und Funktionskörpern werden eingerückt.
	r	f	Aussage [Variablen etc.]
	☐	☐	Durch den Einsatz von Variablen kann man bestimmte Werte an unterschiedlichen Stellen im Programm verwenden.
	☐	☐	<code>abcd</code> ist ein sinnvoller Variablenname für ein Turtle-Objekt.
	☐	☐	<code>"Hallo"</code> ist eine Zeichenkette.
	☐	☐	In Variablen sind Leerzeichen erlaubt.
	☐	☐	Variablennamen müssen mit einer Zahl beginnen.
	☐	☐	Das Zeichnen <code>=</code> nennt man in Python den Zuweisungsoperator.
	r	f	Aussage [Turtle etc.]
	☐	☐	Nach Ausführung der Anweisung <code>heinz.left(360)</code> schaut das Turtle-Objekt <code>heinz</code> in eine andere Richtung.
	☐	☐	Die Anweisungen <code>heinz.left(45)</code> und <code>heinz.right(315)</code> haben die gleichen Auswirkungen auf das Turtle-Objekt <code>heinz</code> .
	☐	☐	Das Anfertigen einer Skizze hilft bei der Erstellung von Zeichnungen mit einem Objekt der Klasse Turtle.
	r	f	Aussage [Python etc.]
	☐	☐	Das Tupel <code>range(100)</code> enthält die Zahlen 1 bis 100.
	☐	☐	In Python können Anweisungen beliebig eingerückt werden.
	☐	☐	Das Tupel <code>range(20)</code> enthält 19 Werte.
	☐	☐	Kommentare beginnen in Python mit dem Zeichen <code>#</code> .
	☐	☐	Die einzelnen Zeilen in einem Programm müssen syntaktisch korrekt sein.
	☐	☐	In der PythonShell kann man Programmideen oder Teile von Programmen einfach testen.