

Technischer Unterricht 4

Übung 1:

1. Verwenden Sie `collections.Counter`, um die Anzahl jedes Zeichens in einer Zeichenkette zu zählen. Schreiben Sie eine Funktion `count_characters`, die eine Zeichenkette annimmt und ein `Counter`-Objekt mit der Anzahl der einzelnen Zeichen zurückgibt. Geben Sie anschließend die 3 am häufigsten vorkommenden Zeichen aus.
2. Verwenden Sie `collections.deque`, um eine Warteschlange mit fester Länge zu erstellen. Schreiben Sie eine Funktion `fixed_queue`, die eine Liste von Zahlen und die maximale Länge der Warteschlange annimmt, die Elemente der Liste zur Warteschlange hinzufügt und den endgültigen Zustand der Warteschlange zurückgibt.
3. Verwenden Sie `collections.defaultdict`, um ein Wörterbuch zu erstellen, bei dem jeder Schlüssel mehrere Werte haben kann (zum Beispiel eine Liste). Schreiben Sie eine Funktion `group_by_first_letter`, die eine Liste von Wörtern annimmt und sie nach dem ersten Buchstaben gruppiert, wobei `defaultdict` verwendet wird.
4. Erstellen Sie ein benanntes Tupel `Point` mit den Feldern `x`, `y` und `z`. Schreiben Sie eine Funktion `calculate_distance`, die zwei `Point`-Objekte annimmt und den Abstand zwischen ihnen im dreidimensionalen Raum berechnet.

Übung 2:

1. Schreiben Sie eine Funktion `safe_divide`, die zwei Zahlen annimmt und das Ergebnis ihrer Division zurückgibt. Wenn die Division nicht möglich ist (Division durch Null), soll die Funktion die Meldung "Cannot divide by zero" zurückgeben, ohne die Ausführung des Programms zu unterbrechen.
2. Schreiben Sie eine Funktion `read_file`, die versucht, eine Textdatei zu öffnen und zu lesen. Wenn die Datei nicht existiert, soll die Funktion den Fehler `FileNotFoundError` behandeln und die Meldung "File not found" ausgeben. Im Falle anderer Fehler soll die Meldung "An error occurred" ausgegeben werden.
3. Schreiben Sie eine Funktion `get_integer`, die den Benutzer auffordert, eine Zahl einzugeben. Wenn der Benutzer keine ganze Zahl eingibt, soll die Funktion die Ausnahme `ValueError` behandeln und die Eingabe erneut abfragen, bis ein gültiger Wert eingegeben wird.
4. Schreiben Sie eine Funktion `calculate_square_root`, die eine Zahl annimmt und deren Quadratwurzel zurückgibt. Wenn die Zahl negativ ist, soll eine Ausnahme

`ValueError` ausgelöst werden. Behandeln Sie die Ausnahme und geben Sie die Meldung "Cannot calculate the square root of a negative number" zurück.

Übung 3:

1. Erstellen Sie einen Generator `Rangelterator`, der einen Iterator implementiert, der ähnlich wie die eingebaute Funktion `range()` funktioniert. Der Generator soll einen Startwert, einen Endwert und einen Schrittwert annehmen, über die Werte iterieren und die Iteration beenden, wenn das Ende des Bereichs erreicht ist.
2. Schreiben Sie eine Funktion `file_line_iterator`, die einen Dateinamen annimmt und einen Iterator zurückgibt, der der Reihe nach die Zeilen der Datei zurückgibt. Verwenden Sie diesen Iterator, um alle Zeilen der Datei einzeln zu lesen und auszugeben.
3. Schreiben Sie einen Generator `fibonacci`, der die Fibonacci-Zahlenfolge bis zu einer bestimmten Anzahl von Elementen generiert. Testen Sie den Generator, indem Sie eine Liste der ersten 10 Fibonacci-Zahlen erstellen.
4. Schreiben Sie einen Generator `infinite_counter`, der eine unendliche Sequenz von natürlichen Zahlen, beginnend mit 1, generiert. Verwenden Sie `itertools.islice`, um die ersten 20 Zahlen aus dieser Sequenz zu erhalten.
5. Schreiben Sie einen Generator `random_numbers`, der zwei Parameter `start` und `end` annimmt und eine unendliche Sequenz von Zufallszahlen in diesem Bereich generiert. Verwenden Sie den Generator, um die ersten 5 Zufallszahlen zu erhalten.
6. Erstellen Sie einen Generator `even_numbers`, der nur gerade Zahlen aus einem gegebenen Bereich generiert. Testen Sie den Generator, indem Sie die geraden Zahlen von 1 bis 20 generieren und ausgeben.
7. Schreiben Sie einen Generator `square_numbers`, der eine Liste von Zahlen annimmt und deren Quadrate generiert. Testen Sie den Generator anhand der Liste `[5, 2, 7, 4, 1]`.
8. Schreiben Sie einen Generator `repeat_string`, der eine Zeichenkette und die Anzahl der Wiederholungen annimmt und die Zeichenkette die angegebene Anzahl von Malen generiert. Testen Sie den Generator mit der Zeichenkette "Hello", die 3 Mal wiederholt wird.

Wörterbuch:

Deutsch	Russisch
Modul	Модуль
collections	collections (название модуля в Python)
Counter	Counter (счетчик, тип данных из collections)
Zeichenkette	Строка (текст)
Warteschlange	Очередь
deque	deque (двусторонняя очередь из collections)
Wörterbuch	Словарь
defaultdict	defaultdict (тип данных из collections)
Tupel	Кортеж
benanntes Tupel	Именованный кортеж
Abstand	Расстояние

Fehlerbehandlung	Обработка ошибок
Division durch Null	Деление на ноль
Datei	Файл
Fehler	Ошибка
FileNotFoundError	FileNotFoundError (исключение в Python)
ValueError	ValueError (исключение в Python)
Ausnahme	Исключение
Quadratwurzel	Квадратный корень
Iterator	Итератор
iterierbares Objekt	Итерируемый объект
Generator	Генератор
Rangelterator	Rangelterator (пользовательский итератор)

fibonacci	fibonacci (последовательность Фибоначчи)
unendliche Sequenz	Бесконечная последовательность
itertools.islice	itertools.islice (функция из itertools)
Zufallszahlen	Случайные числа
Filterung	Фильтрация
Quadrate	Квадраты
Wiederholungen	Повторы