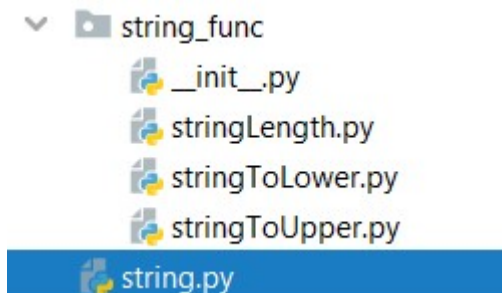


18. Applikationen erstellen in Python

18.1. Modularisierung und Projektstruktur

- unter Modularisierung versteht man die Aufteilung des Quellcodes in Module
- ein Modul umfasst üblicherweise Klassendefinitionen, Datentypen und Funktionen, die einem bestimmten Zweck dienen
- Module lassen sich in einem Programm einbinden
- Module wiederum lassen sich in Pakete zusammenfassen
- ein Paket kann beliebig viele weitere Pakete enthalten
- um ein Paket zu erzeugen, ist eine Verzeichnisstruktur zu erstellen:

- Beispiel:



- in diesem Verzeichnis befindet sich das Programm `string.py`
- im Unterverzeichnis `string_func` befinden sich die Module
- diese Module werden in das Programm eingebunden:

- `string.py`

```
# Einbinden der Module
import string_func

str1 = "Hallo, Welt!"

print(string_func.stringLength(str1))
print(string_func.stringToLower(str1))
print(string_func.stringToUpper(str1))
```

- `__init__.py`

```
# Zusammenfassen der Module im Unterverzeichnis string_func
from .stringLength import stringLength
from .stringToLower import stringToLower
from .stringToUpper import stringToUpper
```

- `stringLength.py`

```
def stringLength(inStr):
    return len(inStr)
```

- `stringToLower.py`

```
def stringToLower(inStr):
    return inStr.lower()
```

- stringToUpper.py

```
def stringToUpper(inStr):  
    return inStr.upper()
```

Aufgaben

1. RemoteControl -
die Applikation ist zu Modularisieren, so dass die Klassendefinition, die GUI-Definitionen, sowie die Ereignisbehandlungen in separaten Python-Dateien gespeichert sind.
2. Bruch -
die Applikation ist zu Modularisieren, so dass die Klassendefinition, die GUI-Definitionen, sowie die Ereignisbehandlungen in separaten Python-Dateien gespeichert sind.