Testdokumentation WeatherStation

Testcase 1: getMeteoData()

|  |  |
| --- | --- |
| Testgrund | Ergebnis |
| Generiert die Methode ein Dictionary, in welchem die entsprechenden Informationen drin sind | TypeError: zuweisen der Wert in das leere Dictionary funktioniert nicht |
| Korrektur zu oben: Keys wurden vordefiniert | Erneut TypeError |
| Fehler gefunden: For-Schleife nicht notwendig, da nicht alle Informationen aus dem json-file benötigt werden | Klappt noch nicht, Indexierungsprobleme |
| Test mit einer Zuweisung von key & value in die weather\_list | Klappt |
| Test mit allen benötigten Zuweisungen von keys und values | Schon wieder Problem mit Indexierung, Zugriff auf Informationen bzgl. |
| Erneuter Test: entsprechende Indexierung als Int und nicht string | Hat geklappt (warum auch immer) |
| Test, ob die entsprechende Bedingung für die Temperatur berücksichtigt wird | Erfolgreich |
| Test, ob Eingabe von F im Konstruktor zur Umwandlung in Fahrenheit führt | Erfolgreich |
| Test, ob M im Konstruktor zu Fehlermeldung führt | Erfolgreich |
| Wie bei Testcase 2.3 wird die Umwandlung der Temperatureinheit (bzw. Einheiten allgemein) vereinfachter dargestellt; Test mit Default-Werten | Erfolgreich |
| Derselbe Test wie eins oberhalb, nun aber mit bei der Instanziierung mitgegebenen Parametern | Erfolgreich |
| Derselbe Test wie eins oberhalb, nun aber mit bei der Instanziierung mitgegebenen Parametern, wobei bei Unit ein falscher Wert mitgegeben wird | Erfolgreich, entsprechender Output wird generiert |
| Getestet mit interaktiver Applikation (User kann input geben) | Fehlermeldung bezüglich Auswahl der Einheit, obwohl ein korrekter Buchstabe eingegeben wurde |
| Erneuter Test mit interaktiver Applikation, diesmal wurde der Input ohne Leerschlag eingegeben | Funktioniert |
| Erneut interaktiver Test, diesmal wird auch die persönliche AppID abgefragt | Funktioniert |
| Test, ob mit «Leerlassen» die Default-Werte beim Instanziieren verwendet werden | Funktioniert nicht |
| Exceptionhandling aus der getcurrentTemp Methode implementiert | funktioniert |

Testcase 2.1: getTempinCelsius(self)

|  |  |
| --- | --- |
| Testgrund | Ergebnis |
| Überprüfen, ob REST Call funktioniert | funktioniert |
| Überprüfen, ob korrekte Zahl herausgegeben wird | Ja, aber als Float; Grad Celsius wird in ganzen Zahlen angegeben 🡪 Int |
| Überprüfen, ob korrekte Zahl als Ganzes herausgegeben wird | ja |

Testcase 2.2: getTempinCelsius(self, temp\_type)

|  |  |
| --- | --- |
| Testgrund | Ergebnis |
| Funktionieren die Bedingungen zu temp\_type: Default-Wert benutzt | Funktioniert ohne überschriebenen Default-Wert |
| Funktionieren die Bedingungen zu temp\_type: Falscher Wert benutzt | UnboundLocalError 🡪 return value temp\_celsius wird nach allen Bedigungen zurückgegeben, jedoch wird nach dem else temp\_celsius kein Wert assigned |
| Korrektur zu vorher: nach jeder Bedingung wurde der return value separat zugewiesen, ausser beim else; dort ist ein print statement | Kein Error wird hervorgerufen, jedoch folgt nach dem print-statement im else ein None |
| Korrektur zu vorher: print statement wurde in return umgewandelt | Erfolgreich: kein None nach dem statement |
| Methode mit «max» als temp\_type in der Funktion | Erfolgreich |
| Methode mit «min» als temp\_type in der Funktion | Erfolgreich |

* Methode getTempinCelsius wurde verworfen, da während der Bearbeitung der Methode getMeteoData die Idee aufkam, die Temperatureinheit in den Konstruktor zu verpacken. Die Test waren jedoch nützlich für das Testen des REST Calls.

Testcase 2.3: getcurrentTemp(self)

|  |  |
| --- | --- |
| Testgrund | Ergebnis |
| Überprüfen, ob Methode mit Default-Wert funktioniert | funktioniert |
| Überprüfen, ob Methode mit neu instanziierten Werten funktioniert | Funktioniert |
| Überprüfen, ob Methode mit falschem temp\_unit den entsprechenden Output gibt | Funktioniert |
| Umwandlung der Temperatureinheit (bzw. Einheiten allgemein) wird vereinfachter dargestellt (zuerst Test mit Default-Wert) | Funktioniert |
| Derselbe Test wie eins oberhalb, nun aber mit bei der Instanziierung mitgegebenen Parametern | Funktioniert |
| Derselbe Test wie eins oberhalb, nun aber mit bei der Instanziierung mitgegebenen Parametern, wobei bei Unit ein falscher Wert mitgegeben wird | Funktioniert, entsprechender Output wird generiert |
| Exceptionhandling eingebaut, falls Tippfehler oder nichtexistente Stadt als Input gegeben wird | Funktioniert nicht |
| Exceptionhandling korrigiert | Funktioniert |
| Bedingung eingebaut, falls keine oder nichtexistente AppID als Input gegeben wird | Funktioniert |

Testcase 3: getSkyDesc()

|  |  |
| --- | --- |
| Testgrund | Ergebnis |
| Überprüfen, ob erwarteter Output herausgegeben wird | Funktioniert |
| Exceptionhandling für Tippfehler oder nichtexistente Stadt als Input von oberen Methoden übernommen | Funktioniert |
| Bedingung eingebaut, falls keine oder nichtexistente AppID als Input gegeben wird | Funktioniert |