# Übung: Entfernungsrechner

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial:

*L4\_2 Info CheckBox.docx*

**(I) Problemstellung**

Wir wollen ein Programm schreiben, mit dem sich ein Kunde bei einer Fluggesellschaft den Flugpreis und die für ihn entstehenden CO2-Emissionen berechnen und anzeigen kann.

Die Berechnung der Emissionen ist etwas kompliziert und wird im Programm schon vorgegeben. Der Flugpreis berechnet sich wie folgt:

* Je geflogenen Kilometer müssen 3 Cent bezahlt werden.
* Je entstandene Tonne CO2 müssen 40 Euro bezahlt werden.

Der Kunde soll auswählen können, ob er seine CO2-Emissionen finanziell ausgleichen möchte, um beispielsweise Naturschutzprojekte zu fördern.

1. Speichern Sie zunächst die Dateien *Flug.py*, *L4\_2\_Aufgabe\_Emissionsrechner.py* und *MeinDialog.ui* in einem gemeinsamen Ordner. Testen Sie, ob Ihr (noch funktionsloses) Programm läuft.
2. Implementieren Sie in der Klasse *Flug* die Methode *berechne\_flugpreis()*. Diese soll

* ... den Preis für die geflogenen Kilometer berechnen und diesen im Attribut *preis\_km* speichern.
* ... die Methode *berechne\_emission()* aufrufen und das Ergebnis im Attribut *emission* speichern.
* ... den Preis für die CO2-Emissionen berechnen und diesen im Attribut *preis\_emission* speichern, falls das Attribut *co2* auf *True* gesetzt ist.
* ... den Gesamtpreis des Fluges berechnen und diesen im Attribut *preis* speichern.

1. Gehen Sie nun in das Programm *L4\_2\_Aufgabe\_Emissionsrechner.py*.

Bei Klick auf die Schaltfläche *„Flugpreis berechnen“* soll die Methode on\_berechnen() ausgeführt werden. Erstellen Sie in dieser Methode Programmcode, der das folgende Struktogramm realisiert.

**(II) Problemanalyse**

1. Welche Ausgabedaten will man erhalten?
2. Welche Daten werden zur Bearbeitung benötigt?
3. Welche Eigenschaften haben die Eingabe-, Verarbeitungs- und Ausgabedaten? (**Variablenliste**)

|  |  |  |
| --- | --- | --- |
| **Bedeutung** | **Typ** | **Variable** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. UML-Diagramm:



1. So sieht die GUI des Programms aus:

|  |
| --- |
| *tf\_ziel*  *tf\_entfernung*  *bt\_ausgeben*  *ta\_ausgabe*  *cb\_emission*  *tf\_start* |

Verarbeitung

|  |
| --- |
|  |

**(III) Struktogramm**

**(IV) Programmcode (Python-Code)**