

Praktikum Objektorientierte Programmierung in C++ (WS 2021/2022)

[Dashboard](#) / [My courses](#) / [Wintersemester 2021/2022](#) / [Ingenieurwissenschaften](#) / [Informatik und Angewandte Kognitionswissenschaften](#)
/ [Praktikum OOP in C++ WS 2021/2022](#) / [Aufgabe 5/Task 5](#)
/ [A5 Teil 1: Hausaufgabe zur Vorbereitung auf die Präsenz-Gruppe/Part 1: Homework Task for Preparation of the Presence Group](#)

A5 Teil 1: Hausaufgabe zur Vorbereitung auf die Präsenz-Gruppe/Part 1: Homework Task for Preparation of the Presence Group

Der weltweit beobachtete Klimawandel steht im Zusammenhang mit den CO2-Emissionen der Menschen. In dieser Aufgabe sollen Klassen zur Mobilität definiert werden, über die CO2-Emissionen berechnet werden sollen, die aus der Benutzung von Verkehrsmitteln resultieren. Das nachfolgende UML-Modell zeigt die Klassen in alphabetischer Reihenfolge./

The climate change observed worldwide is related to the CO2 emissions of humans. In this task some classes are to be defined, over which CO2 emissions are to be calculated, which result from the use of means of transport. The following UML model shows the classes in alphabetic order.

Bus	Car	Cruise	Flight	Train
- local_transport : Boolean - km : Real	- l_fuel_km : Real - km : Real	- days : UnlimitedNatural - sea_days : UnlimitedNatural	- km : Real	- local_transport : Boolean - km : Real - one_way : Boolean
+ Bus() - get_co2_pkm() + get_co2() + print()	+ Car() - get_co2_km() + get_co2() + print()	+ Cruise() - get_co2_day() + get_co2() + print()	+ Flight() - get_co2_pkm() + get_co2() + print()	+ Train() - get_co2_pkm() + get_co2() + print()

1. Definieren Sie eine Klasse mit Namen **Bus** mit folgenden Mitgliedern:/
- Define a class with name **Bus** with following members:
- privates Boolesches Attribut namens **local_transport** (Nahverkehr), ob der Bus im Nahverkehr fährt oder nicht, also im Fernverkehr./

private Boolean attribute named **local_transport**, whether the bus runs on local transport or not, i.e. long-distance transport.

◦ private Gleitpunktzahl namens **km** als Attribut für die Anzahl der gefahrenen Kilometer./

private floating point number named **km** as an attribute for the number of kilometers driven.

◦ öffentlicher Standard-Konstruktor, in dessen Rumpf wie im Beispiel unten vom Standard-Zeichen-Eingabestrom der Wert für das Boolesche Attribut und die gefahrenen Kilometer eingelesen und in den beiden Attributen gespeichert werden./

public standard constructor in whose body, as in the example below, the value for the Boolean attribute and the kilometers traveled are read from the standard character input stream and stored in the two attributes.

◦ öffentlicher Konstruktor mit zwei geeigneten Parametern für die Initialisierung der beiden Attribute mit den Werten der Parameter./

public constructor with two suitable parameters for initialising the two attributes with the values of the parameters.

◦ private Member-Funktion mit Namen **get_co2_pkm** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll für einen Linienbus im Nahverkehr der Wert 0.111 kg CO2-Emission pro gefahrenem Personen-Kilometer oder für einen Bus im Fernverkehr 0.036 kg CO2-Emission pro gefahrenem Personen-Kilometer (je nach Wert des Attributs **local_transport**)/

private member function named **get_co2_pkm** without parameter and with a floating point number as return. In the body, the value 0.111 kg CO2 emission per passenger-kilometer driven should be returned for a local transport bus or 0.036 kg CO2 emission per passenger-kilometer driven for a long-distance bus (depending on the value of the attribute **local_transport**).

◦ öffentliche Member-Funktion mit Namen **get_co2** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll das Produkt aus den gefahrenen Kilometern und dem passenden Emissionswert pro Kilometer (über einen Aufruf der vorher definierten Member-Funktion) berechnet und zurück gegeben werden./

public member function named **get_co2** without parameter and with a floating point number as return. In the body, the product of the kilometers driven and the respective emission value for one kilometer (via a call to the previously defined member function) shall be calculated and returned.

◦ öffentliche Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabe. Im Rumpf soll wie im Beispiel unten nach der Zeichenkette "**bus travel**" je nach Wert des Booleschen Attributs "**local transport**" oder "**long distance**" auf den Standard-Zeichen-Ausgabestrom geschrieben werden, die gefahrenen Kilometer mit einer Nachkommastelle, in runden Klammern der CO2-Emissionswert pro Personen-Kilometer mit drei Nachkommastellen und die berechnete CO2-Emission mit einer Nachkommastelle (rufen Sie die vorherigen Member-Funktionen geeignet auf)/

public member function with name **print** without parameter and without return. In the body, as in the example below, after the string "**bus travel**", depending on the value of the Boolean attribute "**local transport**" or "**long distance**" shall be written onto

the standard character output stream, the kilometers driven with one decimal place, in round brackets the CO2 emission value per person-kilometer with three decimal places and the calculated CO2 emission with one decimal place (call the previous member functions suitably).

2. Definieren Sie eine Klasse mit Namen **Car** mit folgenden Mitgliedern: /

Define a class with name **Car** with following members:

- private Gleitpunktzahl namens **l_fuel_km** als Attribut für den Spritverbrauch in Litern auf 100 Kilometer. /
private floating point number named **l_fuel_km** as an attribute for the liters consumption of fuel per 100 kilometers.
- private Gleitpunktzahl namens **km** als Attribut für die Anzahl der gefahrenen Kilometer. /
private floating point number named **km** as an attribute for the number of kilometers driven.
- öffentlicher Standard-Konstruktor, in dessen Rumpf wie im Beispiel unten vom Standard-Zeichen-Eingabestrom zwei Werte für den Verbrauch und die gefahrenen Kilometer eingelesen und in den Attributen gespeichert werden soll. /
public standard constructor in whose body, as in the example below, the value for consumption of fuel and the kilometers traveled shall be read from the standard character input stream and stored in the attributes.
- öffentlicher Konstruktor mit zwei geeigneten Parametern für die Initialisierung der beiden Attribute mit den Werten der Parameter. /
public constructor with two suitable parameters for initialising the two attributes with the values of the parameters.
- private Member-Funktion mit Namen **get_co2_km** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll die CO2-Emission für einen gefahrenen Kilometer berechnet werden, wobei pro Liter Spritverbrauch 2.3 kg CO2 Emission anfallen (Hinweis: beachten Sie, dass die 2.3 kg auf 100 km anfallen). /
private member function named **get_co2_km** without parameter and with a floating point number as return. In the body, the CO2 emission is to be calculated for one kilometer driven with 2.3 kg of CO2 emission per liter of fuel consumed (hint: the 2.3 kg consumption is per 100 km).
- öffentliche Member-Funktion mit Namen **get_co2** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll das Produkt aus den gefahrenen Kilometern und dem passenden Emissionswert (über einen Aufruf der vorher definierten Member-Funktion) berechnet und zurück gegeben werden. /
public member function named **get_co2** without parameter and with a floating point number as return. In the body, the product of the kilometers driven and the respective emission value (via a call to the previously defined member function) shall be calculated and returned.
- öffentliche Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabe. Im Rumpf soll wie im Beispiel unten die Zeichenkette "**passenger car travel**" auf den Standard-Zeichen-Ausgabestrom geschrieben werden, die gefahrenen Kilometer mit einer Nachkommastelle, in runden Klammern der CO2-Emissionswert pro Kilometer mit drei Nachkommastellen und die berechnete CO2-Emission mit einer Nachkommastelle (rufen Sie die vorherigen Member-Funktionen geeignet auf). /
public member function with name **print** without parameter and without return. In the body, as in the example below the string "**passenger car travel**" shall be written onto the standard character output stream, the kilometers driven with one decimal place, in round brackets the CO2 emission value per kilometer with three decimal places and the calculated CO2 emission with one decimal place (call the previous member functions suitably).

3. Definieren Sie eine Klasse mit Namen **Cruise** (Kreuzfahrt) mit folgenden Mitgliedern: /

Define a class with name **Cruise** with following members:

- private vorzeichenlose Zahl namens **days** als Attribut für die Anzahl der Tage der Kreuzfahrt. /
private floating point number named **days** as an attribute for the number of days of the cruise.
- private Gleitpunktzahl namens **sea_days** als Attribut für die Anzahl der reinen Seetage bei der Kreuzfahrt. /
private floating point number named **sea_days** as an attribute for the number of sea days of the cruise.
- öffentlicher Standard-Konstruktor, in dessen Rumpf wie im Beispiel unten vom Standard-Zeichen-Eingabestrom die Werte für die Anzahl Tage der Kreuzfahrt und die Anzahl der Seetage eingelesen und in den beiden Attributen gespeichert werden. /
public standard constructor in whose body, as in the example below, the values for the number of days/duration of the cruise and the number of sea days are read from the standard character input stream and stored in the two attributes.
- öffentlicher Konstruktor mit zwei geeigneten Parametern für die Initialisierung der beiden Attribute mit den Werten der Parameter. /
public constructor with two suitable parameters for initialising the two attributes with the values of the parameters.
- private Member-Funktion mit Namen **get_co2_day** mit einem Booleschen Parameter für einen Seetag oder nicht und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll für einen Seetag (Parameterwert **true**) der Wert 280 kg CO2-Emission oder für einen Hafentag 190 kg CO2-Emission zurück gegeben werden. /
private member function named **get_co2_day** with a Boolean parameter for a sea day or not and with a floating point number as return. In the body, for a sea day (parameter value **true**) the value 280 kg CO2 emission or for a harbour day 190 kg CO2 emission shall be returned.
- öffentliche Member-Funktion mit Namen **get_co2** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf sollen die CO2-Emissionen für die Hafen- und Seetage - berechnet über das jeweilige Produkt aus den Tagen und dem passenden Emissionswert (über einen Aufruf der vorher definierten Member-Funktion) - addiert und zurück gegeben werden. /
public member function named **get_co2** without parameter and with a floating point number as return. In the body, the result of the addition of the CO2 emissions of the harbour and sea days calculated each by the product of the days and the respective emission value (via a call to the previously defined member function) shall be returned.
- öffentliche Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabe. Im Rumpf soll wie im Beispiel unten die Zeichenkette "**cruise holiday (without flights):**" auf den Standard-Zeichen-Ausgabestrom geschrieben werden, die Anzahlen der Tage und Seetage, in runden Klammern die beiden CO2-Emissionswerte für See- und Hafentage mit drei Nachkommastellen und die berechnete CO2-Emission mit einer Nachkommastelle (rufen Sie die vorherigen Member-Funktionen

geeignet auf)./

public member function with name **print** without parameter and without return. In the body, as in the example below, the string **"cruise holiday (without flights) :**" shall be written onto the standard character output stream, the number of days and sea days, in round brackets the CO2 emission values for sea and harbour days with three decimal places and the calculated CO2 emission with one decimal place (call the previous member functions suitably).

4. Definieren Sie eine Klasse mit Namen **Flight** mit folgenden Mitgliedern: /

Define a class with name **Flight** with following members:

- private Gleitpunktzahl namens **km** als Attribut für die Anzahl der geflogenen Kilometer. /
private floating point number named **km** as an attribute for the number of flown kilometers.
- öffentlicher Standard-Konstruktor, in dessen Rumpf wie im Beispiel unten vom Standard-Zeichen-Eingabestrom ein Wert für die geflogenen Kilometer eingelesen und im Attribut gespeichert werden soll. /
public standard constructor in whose body, as in the example below, the value for the kilometers flown shall be read from the standard character input stream and stored in the attribute.
- öffentlicher Konstruktor mit einem geeigneten Parameter für die Initialisierung des Attributs mit dem Wert des Parameters. /
public constructor with one suitable parameter for initialising the attribute with the value of the parameter.
- private Member-Funktion mit Namen **get_co2_pkm** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll ein pauschal berechneter Emissionswert von 0.27 kg CO2 für einen geflogenen Personen-Kilometer ohne Differenzierung zwischen Kurz- und Langstrecken oder Flugzeugtypen zurück gegeben werden. /
private member function named **get_co2_pkm** without parameter and with a floating point number as return. In the body, a flatly calculated emission value of 0.27 kg CO2 for a flown passenger-kilometer shall be returned without differentiation between short and long distance or aircraft types.
- öffentliche Member-Funktion mit Namen **get_co2** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll das Produkt aus den geflogenen Kilometern und dem Emissionswert (über einen Aufruf der vorher definierten Member-Funktion) berechnet und zurück gegeben werden. /
public member function named **get_co2** without parameter and with a floating point number as return. In the body, the product of the kilometers flown and the emission value (via a call to the previously defined member function) shall be calculated and returned.
- öffentliche Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabe. Im Rumpf soll wie im Beispiel unten die Zeichenkette **"flight travel"** auf den Standard-Zeichen-Ausgabestrom geschrieben werden, die geflogenen Kilometer mit einer Nachkommastelle, in runden Klammern der CO2-Emissionswert pro Personen-Kilometer mit drei Nachkommastellen und die berechnete CO2-Emission mit einer Nachkommastelle (rufen Sie die vorherigen Member-Funktionen geeignet auf). /
public member function with name **print** without parameter and without return. In the body, as in the example below the string **"flight travel"** shall be written onto the standard character output stream, the kilometers flown with one decimal place, in round brackets the CO2 emission value per person-kilometer with three decimal places and the calculated CO2 emission with one decimal place (call the previous member functions suitably).

5. Definieren Sie eine Klasse mit Namen **Train** (Zug) mit folgenden Mitgliedern: /

Define a class with name **Train** with following members:

- privates Boolesches Attribut namens **local_transport** (Nahverkehr), ob der Zug im Nahverkehr fährt oder nicht, also im Fernverkehr. /
private Boolean attribute named **local_transport**, whether the train runs on local transport or not, i.e. long-distance transport.
- privates Boolesches Attribut namens **one_way**, ob der Zug nur hin fährt/einfache Hinfahrt oder hin und zurück. /
private Boolean attribute named **one_way**, whether the train runs one way or return/roundtrip.
- private Gleitpunktzahl namens **km** als Attribut für die Anzahl der gefahrenen Kilometer. /
private floating point number named **km** as an attribute for the number of kilometers driven.
- öffentlicher Standard-Konstruktor, in dessen Rumpf wie im Beispiel unten vom Standard-Zeichen-Eingabestrom die Werte für beide Boolesche Attribute und die gefahrenen Kilometer eingelesen und in den drei Attributen gespeichert werden. /
public standard constructor in whose body, as in the example below, the values for the two Boolean attributes and the kilometers traveled are read from the standard character input stream and stored in the three attributes.
- öffentlicher Konstruktor mit drei geeigneten Parametern für die Initialisierung der drei Attribute mit den Werten der Parameter - eine einfache Fahrt/nur Hinfahrt soll als Default-Parameter eingestellt sein. /
public constructor with two suitable parameters for initialising the two attributes with the values of the parameters - a one-way travel shall be defined as default parameter.
- private Member-Funktion mit Namen **get_co2_pkm** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll für eine Zugfahrt im Nahverkehr der Wert 0.085 kg CO2-Emission pro gefahrenem Personen-Kilometer oder für eine Zugfahrt im Fernverkehr 0.035 kg CO2-Emission pro gefahrenem Personen-Kilometer (je nach Wert des Attributs **local_transport**). /
private member function named **get_co2_pkm** without parameter and with a floating point number as return. In the body, the value 0.085 kg CO2 emission per passenger-kilometer driven should be returned for a local transport train or 0.035 kg CO2 emission per passenger-kilometer driven for a long-distance train (depending on the value of the attribute **local_transport**).
- öffentliche Member-Funktion mit Namen **get_co2** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe. Im Rumpf soll das Produkt aus den gefahrenen Kilometern und dem passenden Emissionswert (über einen Aufruf der vorher definierten Member-Funktion) berechnet und zurück gegeben werden. Bei einer Hin- und Rückfahrt soll der doppelte CO2-Emissions-Wert berechnet und zurück gegeben werden. /
public member function named **get_co2** without parameter and with a floating point number as return. In the body, the product of the

kilometers driven and the respective emission value (via a call to the previously defined member function) shall be calculated and returned. In case of a round trip, a doubled CO2 emission value shall be calculated and returned.

- öffentliche Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabe. Im Rumpf soll wie im Beispiel unten nach der Zeichenkette "**train travel**" je nach Wert des Booleschen Attributs "**local transport**" oder "**long distance**" auf den Standard-Zeichen-Ausgabestrom geschrieben werden, die gefahrenen Kilometer mit einer Nachkommastelle, in runden Klammern der CO2-Emissionswert pro Personen-Kilometer mit drei Nachkommastellen und die berechnete CO2-Emission mit einer Nachkommastelle (rufen Sie die vorherigen Member-Funktionen geeignet auf)./

public member function with name **print** without parameter and without return. In the body, as in the example below, after the string "**train travel**", depending on the value of the Boolean attribute "**local transport**" or "**long distance**" shall be written onto the standard character output stream, the kilometers driven with one decimal place, in round brackets the CO2 emission value per person-kilometer with three decimal places and the calculated CO2 emission with one decimal place (call the previous member functions suitably).

6. Schreiben Sie eine Funktion **main** mit folgenden Definitionen und Anweisungen:/

Write a function **main** with following definitions and statements:

- definieren Sie eine Objekt-Variable **b1** vom Typ **Bus** und geben über den Standardkonstruktor als Beispiel Fernverkehr und 548.3 km (Entfernung Duisburg-Berlin) auf der Tastatur ein./
define an object variable **b1** of type **Bus** and input via the standard constructor as example long distance and 548.3 km (distance Duisburg-Berlin) on keyboard.
- definieren Sie eine Zeigervariable **b2** vom Typ **Bus** und initialisieren diese mit einem neuen Objekt auf dem Heap, wobei als zwei Parameter im Konstruktoraufbau ein Boolescher Wert für Nahverkehr und 2200 km (220 Präsenztage mit 2 * 5 km Busfahrt zur Arbeit/Uni) angegeben sein sollen./
define a pointer variable **b2** of type **Bus** and initialise it with a new object on heap, having as two parameters in the constructor call a Boolean value for local transport and 2200 km (example: 220 attendance days with 2 * 5 km bus ride to work/university).
- senden Sie eine Nachricht **print** an das Objekt **b1**./
send a message **print** to object **b1**.
- senden Sie eine Nachricht **print** an das Objekt, auf das Zeiger **b2** zeigt./
send a message **print** to the object pointer **b2** points to
- definieren Sie eine Objekt-Variable **c1** vom Typ **Car** und geben über den Standardkonstruktor als Beispiel einen Verbrauch von 6 l auf 100 km und 432.78 km (Entfernung Duisburg-Stuttgart) auf der Tastatur ein./
define an object variable **c1** of type **Car** and input via the standard constructor as example a consumption of 6 l fuel per 100 km and 432.78 km (distance Duisburg-Stuttgart) on keyboard.
- definieren Sie eine Zeigervariable **c2** vom Typ **Car** und initialisieren diese mit einem neuen Objekt auf dem Heap, wobei als Parameter im Konstruktoraufbau ein Verbrauch von 7.5 l auf 100 km und 5500 km (Beispiel: 220 Präsenztage mit 25 km typische Pendlerstrecke) angegeben sein sollen./
define a pointer variable **c2** of type **Car** and initialise it with a new object on heap, having as two parameters in the constructor call 7.5 l fuel consumption per 100 km and 5500 km (example: 220 attendance days with 25 km typical commuter distance).
- senden Sie eine Nachricht **print** an das Objekt **c1**./
send a message **print** to object **c1**.
- senden Sie eine Nachricht **print** an das Objekt, auf das Zeiger **c2** zeigt./
send a message **print** to the object pointer **c2** points to
- definieren Sie eine Objekt-Variable **h1** vom Typ **Cruise** und geben über den Standardkonstruktor als Beispiel 3 Tage Dauer mit 1 Seetag auf der Tastatur ein./
define an object variable **h1** of type **Cruise** and input via the standard constructor as example 3 days duration and 1 sea day on keyboard.
- definieren Sie eine Zeigervariable **h2** vom Typ **Cruise** und initialisieren diese mit einem neuen Objekt auf dem Heap, wobei als Parameter im Konstruktoraufbau 14 Tage Dauer und 3 Seetage angegeben sein sollen./
define a pointer variable **h2** of type **Cruise** and initialise it with a new object on heap, having as two parameters in the constructor call 14 days duration and 3 sea days.
- senden Sie eine Nachricht **print** an das Objekt **h1**./
send a message **print** to object **h1**.
- senden Sie eine Nachricht **print** an das Objekt, auf das Zeiger **h2** zeigt./
send a message **print** to the object pointer **h2** points to
- definieren Sie eine Objekt-Variable **f1** vom Typ **Flight** und geben über den Standardkonstruktor als Beispiel 15587.88 km (7793.94 km Luftlinie Frankfurt (Main)-Peking) auf der Tastatur ein./
define an object variable **f1** of type **Flight** and input via the standard constructor 15587.88 km (7793.94 km air distance Frankfurt (Main)-Beijing) on keyboard.
- definieren Sie eine Zeigervariable **f2** vom Typ **Flight** und initialisieren diese mit einem neuen Objekt auf dem Heap, wobei als Parameter im Konstruktoraufbau 2669.9 km (1334.95 km Luftlinie Düsseldorf-Palma de Mallorca) angegeben sein sollen./
define a pointer variable **f2** of type **Flight** and initialise it with a new object on heap, having as two parameters in the constructor call 2669.9 km (1334.95 km air distance Dusseldorf-Palma de Mallorca) .
- senden Sie eine Nachricht **print** an das Objekt **f1**./
send a message **print** to object **f1**.

- senden Sie eine Nachricht **print** an das Objekt, auf das Zeiger **f2** zeigt./
send a message print to the object pointer f2 points to
- definieren Sie eine Objekt-Variable **t1** vom Typ **Train** und geben über den Standardkonstruktor als Beispiel Nahverkehr, 53 km (Entfernung Duisburg-Dortmund) und Hin- und Rückfahrt auf der Tastatur ein./
define an object variable t1 of type Train and input via the standard constructor as example local transport, 53 km (distance Duisburg-Dortmund) and roundtrip/return travel on keyboard.
- definieren Sie eine Zeigervariable **t2** vom Typ **Train** und initialisieren diese mit einem neuen Objekt auf dem Heap, wobei als drei Parameter im Konstruktorauf Ruf ein Boolescher Wert für Fernverkehr, 628.8 km (Entfernung Duisburg-München) und Hin- und Rückfahrt angegeben sein sollen./
define a pointer variable t2 of type Train and initialise it with a new object on heap, having as three parameters in the constructor call a Boolean value for long distance, 628.8 km (distance Duisburg-Munich) and roundtrip/return travel.
- senden Sie eine Nachricht **print** an das Objekt **t1**./
send a message print to object t1.
- senden Sie eine Nachricht **print** an das Objekt, auf das Zeiger **t2** zeigt./
send a message print to the object pointer t2 points to.
- Summieren Sie die CO2-Emissionen der 10 Beispiel-Objekte auf, geben den Summenwert wie im Beispiel unten in den Einheiten Kilogramm und Tonnen aus sowie einen abschließenden Satz **"Bei einer rein an der CO2-Absorptionskapazität der Natur orientierten Pro-Kopf-Rechnung dürfte sich derzeit jeder Weltbürger max. 2.3 t pro Jahr leisten."/**
Sum up all CO2-emissions of the 10 example objects, output the sum value as in the example below in the units kilograms and tonnes as well as a concluding sentence "Based purely on the CO2 absorption capacity of nature, each world citizen should currently afford a maximum of around 2.3 tons per year."

Beispiel Programmlauf/Example Program Run

```

bus travel data input:
(1) local transport
(2) long distance
your choice: 2
kilometers: 548.3
bus travel long distance: 548.3 km (0.036 kg CO2/pkm) 19.7 kg CO2
bus travel local transport: 2200.0 km (0.111 kg CO2/pkm) 244.2 kg CO2

passenger car travel data input:
consumption liter petrol per 100 kilometers: 6
kilometers: 432.78
passenger car travel: 432.8 km (0.138 kg CO2/km) 59.7 kg CO2
passenger car travel: 5500.0 km (0.172 kg CO2/km) 948.8 kg CO2

cruise holiday data input:
number of days of the whole cruise: 3
number of sea days: 1
cruise holiday (without flights): 3 days duration with 1 sea days (sea day 280.000 kg CO2/pkm, harbour day 190.000 kg CO2/pkm) 660.0 kg CO2
cruise holiday (without flights): 14 days duration with 3 sea days (sea day 280.000 kg CO2/pkm, harbour day 190.000 kg CO2/pkm) 2930.0 kg CO2

flight travel data input:
kilometers: 15587.88
flight travel: 15587.9 km (0.270 kg CO2/pkm) 4208.7 kg CO2
flight travel: 2669.9 km (0.270 kg CO2/pkm) 720.9 kg CO2

train travel data input:
(1) local transport
(2) long distance
your choice: 1
kilometers: 53
(1) one-way
(2) return
your choice: 1
train travel local transport: 53.0 km (one way, 0.085 kg CO2/pkm) 4.5 kg CO2
train travel long distance: 628.8 km (return, 0.035 kg CO2/pkm) 44.0 kg CO2

the added total of all CO2 emissions is 9840.5 kg = 9.8 tons

Based purely on the CO2 absorption capacity of nature, each world citizen
should currently afford a maximum of around 2.3 tons per year.

```

