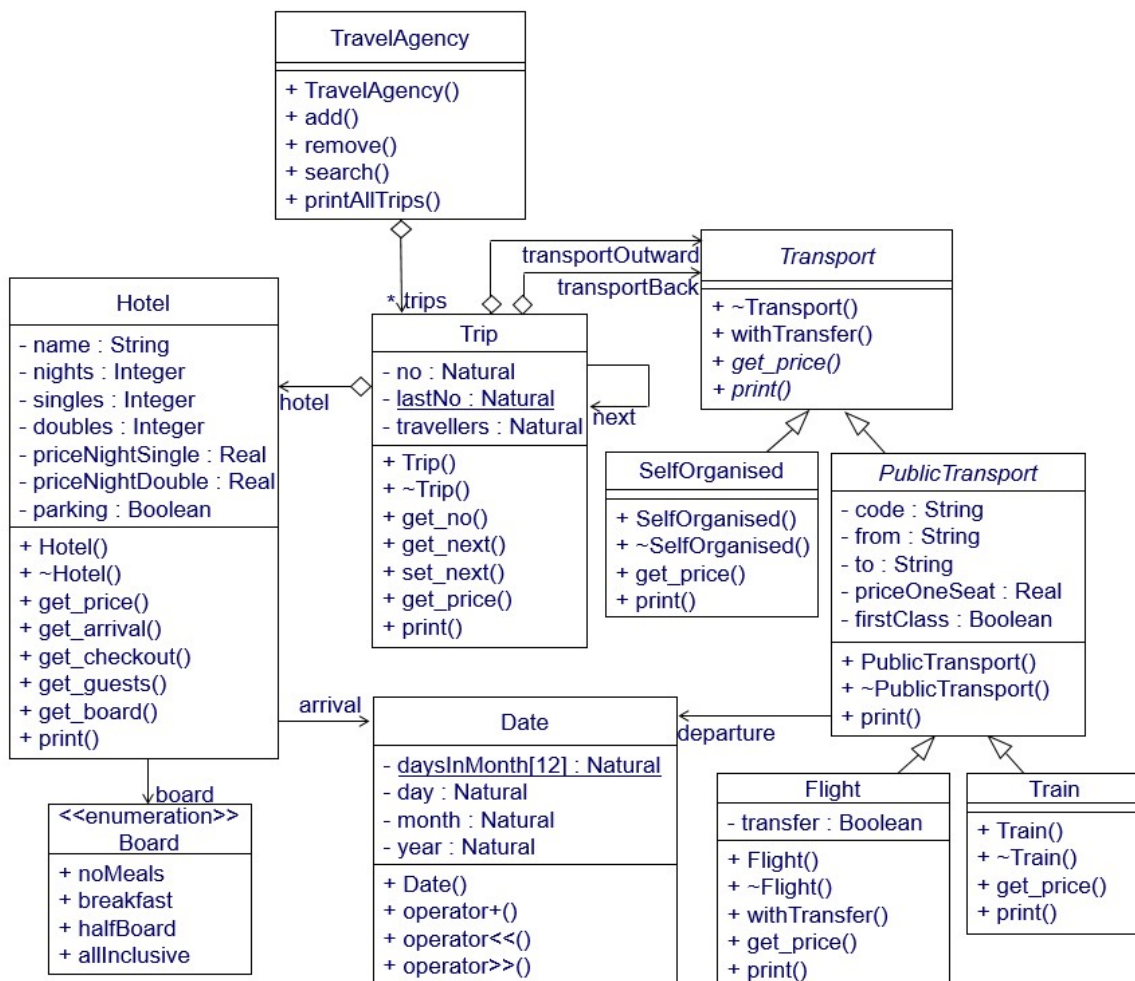


OOP/C++ Praktikum WS 2018/2019: Hotline Reisebüro/Hotline Travel Agency

Für ein Reisebüro (**TravelAgency**) soll eine vereinfachte Verwaltung von Anfragen für Reisen (**Trip**) programmiert werden, die ein **Hotel** und den **Transport** für die Hin- und Rückreise per Flug (**Flight**) oder Zug (**Train**) oder eine Selbstanreise (**SelfOrganised**) beinhalten. Das nachfolgende UML-Diagramm zeigt die zu implementierenden Klassen: /

For a **TravelAgency** a simplified management of enquiries of **Trips** shall get programmed including a **Hotel** and the **Transport** for the outward journey and the journey back by **Flight** or **Train** or **SelfOrganised**. The following UML diagram shows the to implement classes:



Teilaufgabe Klasse Date/Task class Date

Definieren Sie eine Klasse mit Namen **Date** für die Speicherung von Datumsangaben mit drei privaten ganzzahligen Attributen **day**, **month** und **year**, einem überladenen öffentlichen Konstruktor zur Initialisierung eines Datums, einen öffentlichen überladenen binären Member-Operator **+** mit einer Anzahl **n** von Tagen als Parameter, der als Ergebnis ein neues Datum **n** Tage später (nach dem 31.12. eines Jahres folgt der 1.1. des nachfolgenden Jahres) zurück liefert, sowie einen überladenen befreundeten Ein- und Ausgabeoperator.

Hinweise: zur Vereinfachung soll der Wert von n maximal 28 Tage (4 Wochen) sein. Für die Implementierung des Member-Operators + ist ein konstantes Feld **daysInMonth** als Klassenvariable hilfreich, in dem für jeden Monat die Anzahl an Tagen gespeichert ist; Schaltjahre sollen ignoriert werden./

Define a class with name `Date` to store a date with three private short unsigned integer attributes **day**, **month** and **year**, a public overloaded constructor to initialise a date, a public overloaded binary member operator `+` with a number `n` of days as parameter returning a new date `n` days later

(after December, 31st, the next day is January, 1st, of the following year), as well as an overloaded friend input and output operator.

Hints: to simplify it the value of *n* shall have a value of maximum 28 days (4 weeks). For the implementation of operator + a constant array **daysInMonth** as class variable may be helpful storing the number of days in each month; leap years shall be ignored.

Aufgabe Aufzählung Verpflegung (Board)/Task class Board

Definieren Sie eine C++ Aufzählung (**enum struct**) mit Werten für keine Verpflegung, Frühstück, Halbpension und All-Inclusive sowie einen überladenen binären Ausgabeoperator dazu./

Define a C++ enumeration (**enum struct**) with values for no meals, breakfast, half pension and all inclusive as well as a related overloaded binary output operator.

Aufgabe Klasse Hotel/Task class Hotel

Definieren Sie eine Klasse mit Namen **Hotel** mit folgenden Members:/

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

- ein privates Attribut vom Typ C++ Zeichenkette (**string**) für den **namen** des Hotels./
a private C++ string attribute for the **name** of the hotel.
- ein privates Attribut vom Typ **Date** für das Ankunftsdatum (**arrival**) im Hotel./
a private attribute of type **Date** for the **arrival** of the hotel.
- drei private ganzzahlige Attribute **nights** für die Anzahl der gebuchten Nächte, **singles** für die Anzahl der gebuchten Einzelzimmer und **doubles** für die Anzahl der gebuchten Doppelzimmer./
three private integer attributes **nights** for the number of booked nights, **singles** for the number of booked single rooms and **doubles** for the number of booked double rooms.
- ein privates Attribut vom Typ **Board** für die gebuchte Verpflegung (**board**)./
a private attribute of type **Board** for the booked **board**.
- zwei private reellwertige Attribute **priceNightSingle** für den Preis eines Einzelzimmers pro Nacht und **priceNightDouble** für den Preis eines Doppelzimmers pro Nacht./
two private real-valued attributes **priceNightSingle** for the price of a single room for one night and **priceNightDouble** for the price of a double room one night.
- ein privates Boolesches Attribut mit Namen **parking** für einen mit gebuchten Parkplatz./
a private Boolean attribute with name **parking** for an additionally booked parking place.
- ein öffentlicher überladener Standard-Konstruktor, in dem alle Werte der Attribute wie in den Beispielen eingelesen und entsprechend initialisiert werden (es sollen maximal 28 Nächte/4 Wochen gebucht werden können)./
a public overloaded standard constructor where all values of the attributes shall be inputted like in the examples below to initialise them appropriately (maximum 28 nights/4 weeks shall be possible to book).
- ein öffentlicher Destruktor, der eine Ausgabe wie in den Beispielen unten macht./
a public destructor outputting a message like shown in the examples below.
- eine öffentliche Methode mit Namen **get_price** ohne Parameter, die den Gesamtpreis berechnet aus der Anzahl der Nächte multipliziert mit den gebuchten Einzel- und Doppelzimmern plus 10 EUR pro Nacht für einen gebuchten Parkplatz./
a public method with name **get_price** without parameter returning the total price calculated by the number of nights multiplied with the booked single and double rooms plus 10 EUR for each night in case of a booked parking place.
- eine öffentliche Methode mit Namen **get_arrival** ohne Parameter, die das Ankunftsdatum zurück gibt./
a public method with name **get_arrival** without parameter returning the arrival date.
- eine öffentliche Methode mit Namen **get_checkout** ohne Parameter, die das Abreisedatum (Ankunftsdatum plus Anzahl Nächte) zurück gibt./
a public method with name **get_checkout** without parameter returning the departure date (arrival date plus number of nights).
- eine öffentliche Methode mit Namen **get_guests** ohne Parameter, die die Gesamtanzahl der Übernachtungsgäste zurück gibt, bspw. 1 bei einem Einzelzimmer, 4 bei zwei Doppelzimmern oder 9 bei drei Einzel- und drei Doppelzimmern gebucht./
a public method with name **get_guests** without parameter returning the number of sleeping guests, e.g. 1 for a booked single room, 4 for two double rooms or 9 for three single and three double rooms booked.

- eine öffentliche Methode mit Namen **print** ohne Parameter, die das Ankunftsdatum, den Namen des Hotels, die Anzahl gebuchter Übernachtungen, die Anzahl gebuchter Einzel- und Doppelzimmer, die gebuchte Verpflegung sowie ob ein Parkplatz gebucht ist ausgibt (siehe Beispiele unten)./
a public method with name **print** without parameter writing the arrival date, the name of the hotel, the number of booked nights, the number of booked single and double rooms, the booked board as well as whether a parking place is booked (see examples below).

Aufgabe Klasse *Transport*/Task class *Transport*

Definieren Sie eine abstrakte Klasse mit Namen **Transport** mit folgenden Members:/
Define an abstract class with name **Transport** with following members:

- ein öffentlicher virtueller Destruktor, der eine entsprechende Ausgabe macht./
a public virtual destructor writing a respective output message.
- eine öffentliche abstrakte/rein virtuelle Member-Funktion mit Namen **withTransfer** ohne Parameter und mit einem Booleschen Rückgabewert./
a public abstract/pure virtual member function with name **withTransfer** without parameter and a Boolean return value.
- eine öffentliche abstrakte/rein virtuelle Member-Funktion mit Namen **get_price** ohne Parameter und mit einem reellwertigen Rückgabewert./
a public abstract/pure virtual member function with name **get_price** without parameter and a real-valued return value.
- eine öffentliche abstrakte/rein virtuelle Member-Funktion mit Namen **print** ohne Parameter und ohne Rückgabewert./
a public abstract/pure virtual member function with name **print** without parameter or return value.

Aufgabe Klasse *SelfOrganised*/Task class *SelfOrganised*

Definieren Sie eine Klasse für eine selbst organisierte An- oder Abreise mit Namen **SelfOrganised**, die öffentlich von der Klasse **Transport** erbt, mit folgenden Members:/
For self organised transport outwards or back define a class with name **SelfOrganised** inheriting public from class **Transport** with following members:

- ein öffentlicher Standard-Konstruktor./
a public standard constructor.
- ein öffentlicher virtueller Destruktor, der eine entsprechende Ausgabe macht./
a public virtual destructor writing a respective output.
- eine öffentliche virtuelle Member-Funktion mit Namen **withTransfer** ohne Parameter, die immer **false** zurück gibt./
a public virtual member function with name **withTransfer** without parameter always returning **false**.
- eine öffentliche virtuelle Member-Funktion mit Namen **get_price** ohne Parameter, die immer **0.00** EUR zurück gibt./
a public virtual member function with name **get_price** without parameter always returning **0.00** EUR.
- eine öffentliche virtuelle Methode mit Namen **print** ohne Parameter oder Rückgabewert, die die Zeichenkette **"self organised transport"** ausgibt (siehe Beispiel unten)./
a public virtual method with name **print** without parameter or return value writing **"self organised transport"** (see example below).

Aufgabe Klasse *PublicTransport*/Task class *PublicTransport*

Definieren Sie eine abstrakte Klasse mit Namen **PublicTransport**, die öffentlich von der Klasse **Transport** erbt, mit folgenden Members:/
Define an abstract class with name **PublicTransport** inheriting public from class **Transport** with following members:

- ein privates Attribut vom Typ **Date** für das Abfahrtsdatum (**departure**)./
a private attribute of type **Date** for the **departure**.
- ein privates Attribut mit Namen **code** vom Typ C++ Zeichenkette (**string**) für den Flug, Zug, ... des Verkehrsmittels./

- a private C++ **string** attribute with name **code** for the flight, train, ...
- ein privates Attribut mit Namen **from** vom Typ C++ Zeichenkette (**string**) für den Abfahrtsort./
- a private C++ **string** attribute with name **from** for the place of departure.
- ein privates Attribut mit Namen **to** vom Typ C++ Zeichenkette (**string**) für den Ankunftsort./
- a private C++ **string** attribute with name **to** for the place of arrival.
- ein privates reellwertiges Attribut mit Namen **priceOneSeat** für den Preis eines Sitzplatzes./
- a private real-valued attribute with name **priceOneSeat** for the price of one seat.
- ein privates Boolesches Attribut mit Namen **firstClass**, ob die Sitzplätze in der ersten Klasse sind./
- a private Boolean attribute with name **firstClass** whether the seats are first class.
- ein öffentlicher Konstruktor mit einem Datum als Parameter für den Abfahrtstag, drei Zeichenketten-Parametern zur Initialisierung des Codes, des Abfahrts- und des Ankunftsorts, einem Parameter für den Preis eines Sitzplatzes sowie einem Booleschen Default-Parameter mit Wert **false**, ob es erste Klasse sein soll./
- a public constructor with a date as parameter for the departure day, three C++ **string** parameters to initialise the code, the places of departure and arrival, a parameter for the price of one seat as well as a Boolean default parameter with value **false** whether first class is wanted.
- ein öffentlicher virtueller Destruktor, der eine entsprechende Ausgabe mit dem Code und dem Abfahrtsdatum macht (siehe Beispiele unten)./
- a public virtual destructor writing a respective output message including the code and date of departure (see examples below).
- eine öffentliche virtuelle Methode mit Namen **print** ohne Parameter oder Rückgabewert, die das Abfahrtsdatum ausgibt, den Code, von welchem Abfahrtsort zu welchem Ankunftsort der Transport stattfindet und "**(first class)**", falls diese angefragt ist (siehe Beispiel unten)./
- a public virtual method with name **print** without parameter or return value writing the date, the code, the places of departure and of arrival and "**(first class)**" in case of wanted (see example below).

Aufgabe Klasse Flight/Task class Flight

Definieren Sie eine Klasse mit Namen **Flight**, die öffentlich von der Klasse **PublicTransport** erbt, mit folgenden Mitgliedern:/

Define a class with name **Flight** inheriting public from class **PublicTransport** with following members:

- ein privates Boolesches Attribut mit Namen **transfer**, ob auch ein Transfer angeboten wird./
- a private Boolean attribute with name **transfer** whether also a transfer is offered.
- ein öffentlicher Konstruktor mit einem Datum als Parameter für den Abflugtag, drei Zeichenketten-Parametern zur Initialisierung des Flug-Kodes, des Abflugs- und des Ankunftsflughafens, einem Parameter für den Preis eines Sitzplatzes, einem Booleschen Default-Parameter mit Wert **false**, ob es ein Sitzplatz erster Klasse sein soll, sowie einem Booleschen Default-Parameter mit Wert **true**, ob ein Transfer dabei ist./
- a public constructor with a date as parameter for the departure day, three C++ **string** parameters to initialise the code of the flight, the airports of departure and arrival, a parameter for the price of one seat, a Boolean default parameter with value **false** whether the seat shall be first class as well as a Boolean default parameter with value **true** whether a transfer is included.
- ein öffentlicher virtueller Destruktor, der eine entsprechende Ausgabe macht./
- a public virtual destructor writing a respective output message.
- eine öffentliche virtuelle Methode mit Namen **withTransfer** ohne Parameter, die den Wert des entsprechenden Attributs zurück liefert./
- a public virtual method with name **withTransfer** without parameter returning the value of the respective attribute.
- eine öffentliche virtuelle Methode mit Namen **get_price** ohne Parameter, die für einen Economy-Sitzplatz den Wert des Attributs **priceOneSeat** und für die erste Klasse dessen doppelten Wert zurück liefert./
- a public virtual method with name **get_price** without parameter returning the value of

attribute **priceOneSeat** for an economy seat and its doubled value for a first class seat.

- eine öffentliche virtuelle Methode mit Namen **print** ohne Parameter oder Rückgabewert, die "**flight**" ausgibt und danach die gleichnamige Methode der direkten Oberklasse aufruft (siehe Beispiel unten)./

a public virtual method with name **print** without parameter or return value writing "**flight**" and afterwards calling the same name method of the direct upper class (see example below).

Aufgabe Klasse Train/Task class Train

Definieren Sie eine Klasse mit Namen **Train**, die öffentlich von der Klasse **PublicTransport** erbt, mit folgenden Mitgliedern:/

Define a class with name **Train** inheriting public from class **PublicTransport** with following members:

- ein öffentlicher Konstruktor mit einem Datum als Parameter für den Abfahrtstag, drei Zeichenketten-Parametern zur Initialisierung des Zug-Kodes, des Abfahrts- und des Ankunftsbahnhofs, einem Parameter für den Preis eines Sitzplatzes und einem Booleschen Default-Parameter mit Wert **false**, ob es ein Sitzplatz erster Klasse sein soll./
a public constructor with a date as parameter for the departure day, three C++ **string** parameters to initialise the code of the train, the train stations of departure and arrival, a parameter for the price of one seat and a Boolean default parameter with value **false** whether the seat shall be first class.
- ein öffentlicher virtueller Destruktor, der eine entsprechende Ausgabe macht./
a public virtual destructor writing a respective output message.
- eine öffentliche virtuelle Member-Funktion mit Namen **withTransfer** ohne Parameter, die immer **false** zurück gibt./
a public virtual member function with name **withTransfer** without parameter always returning **false**.
- eine öffentliche virtuelle Methode mit Namen **get_price** ohne Parameter, die für einen Sitzplatz in der zweiten Klasse den Wert des Attributs **priceOneSeat** und für die erste Klasse den 1.5-fachen Wert zurück liefert./
a public virtual method with name **get_price** without parameter returning the value of attribute **priceOneSeat** for a seat in second class and its 1.5 fold value for a first class seat.
- eine öffentliche virtuelle Methode mit Namen **print** ohne Parameter oder Rückgabewert, die "**train**" ausgibt und danach die gleichnamige Methode der direkten Oberklasse aufruft (siehe Beispiel unten)./
a public virtual method with name **print** without parameter or return value writing "**train**" and afterwards calling the same name method of the direct upper class (see example below).

Aufgabe Klasse Trip/Task class Trip

Definieren Sie eine Klasse für eine geplante Urlaubsreise (**Trip**) mit folgenden Mitgliedern:/

Define a class for a holiday **Trip** planned with following members:

- ein privates konstantes vorzeichenlos ganzzahliges Attribut **no** für eine Reisebüro-interne einmalig vorkommende fortlaufend vergebene Anfragenummer für jedes Objekt./
a private constant unsigned integer attribute **no** for a travel agency internal consecutively given unique number for each object.
- eine private statische vorzeichenlos ganzzahlige Klassenvariable mit Namen **lastNo** zur Initialisierung einer fortlaufenden Nummer für jede Urlaubsreise, also die bei jeder Erzeugung eines neuen Objekts dieser Klasse inkrementiert wird, **no** initialisiert und somit die eindeutige, identifizierende Nummer aller Urlaubsreisen realisiert. Diese statische Klassenvariable soll mit dem Wert 0 - ausserhalb der Klasse - initialisiert werden./
a private static unsigned integer class variable with name **lastNo** to initialise sequential numbers **no** for each holiday trip, which shall get incremented each time a new object of this class is generated and is used to automatically get unique identifying numbers for all trips. This class variable shall be initialised - outside of the class - with value 0.
- ein privates vorzeichenlos ganzzahliges Attribut **travellers** für die Anzahl Personen dieser Urlaubsreise./
a private unsigned integer attribute **travellers** for the number of persons of this holiday trip.

- ein privates Zeiger-Attribut **hotel** auf das **Hotel** dieser Urlaubsreise./
a private pointer attribute **hotel** to the **Hotel** of this holiday trip.
- ein privates Zeiger-Attribut **transportOutward** auf die Hinreise zu dieser Urlaubsreise./
a private pointer attribute **transportOutward** to the outward journey of this holiday trip.
- ein privates Zeiger-Attribut **transportBack** auf die Rückreise zu dieser Urlaubsreise./
a private pointer attribute **transportBack** to the journey back of this holiday trip.
- ein privates Zeiger-Attribut **next** auf eine Urlaubsreise zum Aufbau einer Liste von Urlaubsreisen./
a private pointer attribute **next** to generate a list of holiday trips.
- ein öffentlicher Konstruktor mit einer Anzahl reisender Personen, Zeigern auf ein Hotel, einen Hin- und Rücktransport sowie einem voreingestellten Zeiger-Parameter mit Wert Nullzeiger für eine in der Liste folgende Urlaubsreise, also insgesamt fünf Parameter zur Initialisierung aller Attribute - die Nummer der Reise und das Inkrementieren des Klassenattributs sind ebenfalls zu berücksichtigen./
a public constructor with a number of travellers, pointers to a hotel, transport outward and back as well as a default parameter pointer with value null pointer to a next trip in the list as in total five parameters to initialise all attributes - the number of the trip and the increment of the class attribute needs to be regarded also.
- ein öffentlicher virtueller Destruktor, der das Hotel, die Hin- und Rückreise mit löscht sowie eine entsprechende Ausgabe macht./
a public virtual destructor deleting also the hotel, the outward transport and back as well as writing a respective output message.
- eine öffentliche Methode mit Namen **get_no** ohne Parameter, die die Nummer der Reise zurück liefert./
a public method with name **get_no** without parameter returning the number of the trip.
- eine öffentliche Methode mit Namen **get_next** ohne Parameter, die einen Zeiger auf die nächste Reise in der Liste zurück liefert./
a public method with name **get_next** without parameter returning a pointer to the next trip within the list.
- eine öffentliche Methode mit Namen **set_next** mit einem Zeiger auf eine Reise als Parameter, die die nächste Reise in der Liste zuweist./
a public method with name **set_next** and a pointer to a trip as parameter assigning it as next trip within the list.
- eine öffentliche Methode mit Namen **get_price** ohne Parameter, die den Preis für die gesamte Reise aller Reisenden addiert aus dem Hotelpreis, den Preisen für die Hin- und die Rückreisen und zurück liefert./
a public method with name **get_price** without parameter returning the calculated price of the complete trip for all travellers added by the price of the hotel, the outward journey and journey back.
- eine öffentliche Methode mit Namen **print** ohne Parameter oder Rückgabewert, die die Nummer der Reise, die Anzahl Personen, das Hotel, die Hin- und Rückreise, ob mit oder ohne Transfer und den Gesamtpreis ausgibt (siehe Beispiel unten)./
a public method with name **print** without parameter or return value writing the number of the trip, the number of travellers, the hotel, the outward journey and journey back, whether with or without transfer and the overall price (see example below).

Aufgabe Klasse TravelAgency/Task class TravelAgency

Definieren Sie eine Klasse für ein Reisebüro mit Namen **TravelAgency** mit folgenden Mitgliedern:/

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

- ein privates Zeiger-Attribut **trips** als Kopf einer Liste von Urlaubsreisen./
a private pointer attribute **trips** as head of a list of holiday trips.
- ein öffentlicher Standard-Konstruktor, der den Kopf der Liste mit einem Nullzeiger initialisiert./
a public standard constructor initialising the head of the list with a null pointer.
- eine öffentliche Methode mit Namen **add** mit einem Zeiger auf eine Urlaubsreise als Parameter, die diese als neuen Kopf der Liste einfügt./
a public method with name **add** with a pointer to a trip as parameter inserting it as head of the list.
- eine öffentliche Methode mit Namen **remove** mit einem Zeiger auf eine Urlaubsreise als Parameter, die diese aus der Liste löscht./

a public method with name **remove** with a pointer to a trip as parameter deleting this trip from the list.

- eine öffentliche Methode mit Namen **search** mit einer Nummer als Parameter, die die Urlaubsreise mit dieser Nummer in der Liste sucht und als Zeiger zurück gibt oder sonst einen Nullzeiger./

a public method with name **search** with a number as parameter searching for the trip with this number and returning a pointer to it or otherwise a null pointer.

- öffentliche Methode mit Namen **printAllTrips** ohne Parameter oder Rückgabewert, die alle Urlaubsreisen in der Liste der Reihe nach ausgibt (siehe Beispiel unten)./

a public method with name **printAllTrips** without parameter or return value outputting all trips within the list (see example below).

Aufgabe main/Task main

Die Hotline für das Reisebüro befindet sich noch im Aufbau, so dass Ihre Software nur eine einfache Funktionalität haben soll. Definieren Sie in **main** ein Objekt von der Klasse **TravelAgency** und weitere Objekte und Variable, die Sie benötigen, und programmieren ein kleines Menü mit drei Funktionalitäten (o für Ende):/

The hotline for the travel agency is in start phase, therefore only a simple functionality shall be offered. Define in **main** an object of class **TravelAgency** and further objects and variables, and program a small menu with three functionalities (o for end):

1. für eine neue Reise soll das Hotel über dessen Standard-Konstruktor, die Hin- und Rückreisedaten in diesem Menüpunkt wie in den Beispielen unten eingegeben werden und die neue Reise der Liste hinzugefügt werden./
for a new trip the hotel by the standard constructor, the outward journey and journey back data within this menu entry shall be inputted as shown in examples below and the new trip added to the list.
2. suchen Sie nach Eingabe einer Nummer nach der Reise mit dieser Nummer, geben diese (falls gefunden) aus und löschen diese optional wieder aus der Liste./
after inputting a number search for a trip with this number, print it (if found) and optionally delete it from the list.
3. geben Sie alle Urlaubsreisen in der Liste aus./
print all trips within the list.

Beispiel Programmlauf/Example Program Run

```
HOTLINE TRAVEL AGENCY
0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 1
name of hotel: Adlon
arrival on: 30.12.2018
how many nights: 3
how many single bed rooms: 1
how many double bed rooms: 1
a all inclusive
b breakfast
h half board
w without meals
h
price one night in single bed room: 100
price one night in double bed room: 150
with parking (y(es) or n(o)): n
please choose transport for outward journey
0 self organised
1 by flight
2 by train
your choice: 2
code of train: ICE 123
main train station of departure: Duisburg Hbf
main train station of arrival: Berlin Hbf
price for one passenger: 15
please choose transport for journey back
0 self organised
1 by flight
```

2 by train
your choice: 2
code of train: ICE 321
main train station of departure: Berlin Hbf
main train station of arrival: Duisburg Hbf
price for one passenger: 17

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 3
trip inquiry 1 for 3 person(s)
check-in: hotel 30.12.2018 Adlon for 3 night(s) 1 double bed room(s) 1 single bed room(s)
half board
outward journey: train 30.12.2018 ICE 123 from Duisburg Hbf to Berlin Hbf
journey back: train 02.01.2019 ICE 321 from Berlin Hbf to Duisburg Hbf
no transfer
price: 846.00 EUR

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 1
name of hotel: Bavaria
arrival on: 25.09.2019
how many nights: 1
how many single bed rooms: 0
how many double bed rooms: 1
a all inclusive
b breakfast
h half board
w without meals
b
price one night in single bed room: 50
price one night in double bed room: 80
with parking (y(es) or n(o)): y
please choose transport for outward journey
0 self organised
1 by flight
2 by train
your choice: 0
please choose transport for journey back
0 self organised
1 by flight
2 by train
your choice: 0

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 3
trip inquiry 2 for 2 person(s)
check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
with breakfast, parking included
outward journey: self organised transport
journey back: self organised transport
no transfer
price: 90.00 EUR

trip inquiry 1 for 3 person(s)
check-in: hotel 30.12.2018 Adlon for 3 night(s) 1 double bed room(s) 1 single bed room(s)
half board
outward journey: train 30.12.2018 ICE 123 from Duisburg Hbf to Berlin Hbf
journey back: train 02.01.2019 ICE 321 from Berlin Hbf to Duisburg Hbf
no transfer
price: 846.00 EUR

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers


```

your choice: 1
name of hotel: RIU
arrival on: 25.05.2019
how many nights: 10
how many single bed rooms: 5
how many double bed rooms: 5
a all inclusive
b breakfast
h half board
w without meals
a
price one night in single bed room: 50
price one night in double bed room: 80
with parking (y(es) or n(o)): n
please choose transport for outward journey
0 self organised
1 by flight
2 by train
your choice: 1
code of flight: KLM 1234
airport of departure: Amsterdam
airport of arrival: Palma
price for one passenger: 29
please choose transport for journey back
0 self organised
1 by flight
2 by train
your choice: 1
code of flight: KLM 4567
airport of departure: Palma
airport of arrival: Duesseldorf
price for one passenger: 39

HOTLINE TRAVEL AGENCY
0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 3
trip inquiry 3 for 15 person(s)
check-in:          hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
                  all inclusive
outward journey:  flight 25.05.2019 KLM 1234 from Amsterdam to Palma
journey back:     flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
transfer included
                  price: 7520.00 EUR

trip inquiry 2 for 2 person(s)
check-in:          hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
                  with breakfast, parking included
outward journey:  self organised transport
journey back:     self organised transport
no transfer
                  price: 90.00 EUR

trip inquiry 1 for 3 person(s)
check-in:          hotel 30.12.2018 Adlon for 3 night(s) 1 double bed room(s) 1 single bed room(s)
                  half board
outward journey:  train 30.12.2018 ICE 123 from Duisburg Hbf to Berlin Hbf
journey back:     train 02.01.2019 ICE 321 from Berlin Hbf to Duisburg Hbf
no transfer
                  price: 846.00 EUR

HOTLINE TRAVEL AGENCY
0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 2
number of trip: 1
trip inquiry 1 for 3 person(s)
check-in:          hotel 30.12.2018 Adlon for 3 night(s) 1 double bed room(s) 1 single bed room(s)
                  half board
outward journey:  train 30.12.2018 ICE 123 from Duisburg Hbf to Berlin Hbf
journey back:     train 02.01.2019 ICE 321 from Berlin Hbf to Duisburg Hbf
no transfer
                  price: 846.00 EUR

```

(d)delete or 🍌ot: d
destructor hotel Adlon at 30.12.2018 for 3 guests done
destructor Train done
destructor PublicTransport ICE 123 at 30.12.2018 done
destructor Transport done
destructor Train done
destructor PublicTransport ICE 321 at 02.01.2019 done
destructor Transport done
destructor Trip done

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 3
trip inquiry 3 for 15 person(s)
check-in: hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
all inclusive
outward journey: flight 25.05.2019 KLM 1234 from Amsterdam to Palma
journey back: flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
transfer included
price: 7520.00 EUR

trip inquiry 2 for 2 person(s)
check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
with breakfast, parking included
outward journey: self organised transport
journey back: self organised transport
no transfer
price: 90.00 EUR

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 2
number of trip: 3
trip inquiry 3 for 15 person(s)
check-in: hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
all inclusive
outward journey: flight 25.05.2019 KLM 1234 from Amsterdam to Palma
journey back: flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
transfer included
price: 7520.00 EUR

(d)delete or 🍌ot: n

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 3
trip inquiry 3 for 15 person(s)
check-in: hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
all inclusive
outward journey: flight 25.05.2019 KLM 1234 from Amsterdam to Palma
journey back: flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
transfer included
price: 7520.00 EUR

trip inquiry 2 for 2 person(s)
check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
with breakfast, parking included
outward journey: self organised transport
journey back: self organised transport
no transfer
price: 90.00 EUR

HOTLINE TRAVEL AGENCY

0 end
1 new trip
2 search trip
3 show all trip offers
your choice: 1
name of hotel: Atlantic

arrival on: 28.2.2019
 how many nights: 1
 how many single bed rooms: 1
 how many double bed rooms: 0
 a all inclusive
 b breakfast
 h half board
 w without meals
 w
 price one night in single bed room: 100
 price one night in double bed room: 200
 with parking (y(es) or n(o)): n
 please choose transport for outward journey
 0 self organised
 1 by flight
 2 by train
 your choice: 0
 please choose transport for journey back
 0 self organised
 1 by flight
 2 by train
 your choice: 0

HOTLINE TRAVEL AGENCY
 0 end
 1 new trip
 2 search trip
 3 show all trip offers
 your choice: 3
 trip inquiry 4 for 1 person(s)
 check-in: hotel 28.02.2019 Atlantic for 1 night(s) 1 single bed room(s)
 without meals
 outward journey: self organised transport
 journey back: self organised transport
 no transfer
 price: 100.00 EUR

trip inquiry 3 for 15 person(s)
 check-in: hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
 all inclusive
 outward journey: flight 25.05.2019 KLM 1234 from Amsterdam to Palma
 journey back: flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
 transfer included
 price: 7520.00 EUR

trip inquiry 2 for 2 person(s)
 check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
 with breakfast, parking included
 outward journey: self organised transport
 journey back: self organised transport
 no transfer
 price: 90.00 EUR

HOTLINE TRAVEL AGENCY
 0 end
 1 new trip
 2 search trip
 3 show all trip offers
 your choice: 2
 number of trip: 1
 trip not found

HOTLINE TRAVEL AGENCY
 0 end
 1 new trip
 2 search trip
 3 show all trip offers
 your choice: 2
 number of trip: 3
 trip inquiry 3 for 15 person(s)
 check-in: hotel 25.05.2019 RIU for 10 night(s) 5 double bed room(s) 5 single bed room(s)
 all inclusive
 outward journey: flight 25.05.2019 KLM 1234 from Amsterdam to Palma
 journey back: flight 04.06.2019 KLM 4567 from Duesseldorf to Palma
 transfer included
 price: 7520.00 EUR

(d)delete or 🍌ot: d

destructor hotel RIU at 25.05.2019 for 15 guests done
destructor Flight done
destructor PublicTransport KLM 1234 at 25.05.2019 done
destructor Transport done
destructor Flight done
destructor PublicTransport KLM 4567 at 04.06.2019 done
destructor Transport done
destructor Trip done

HOTLINE TRAVEL AGENCY

0 end

1 new trip

2 search trip

3 show all trip offers

your choice: 3

trip inquiry 4 for 1 person(s)

check-in: hotel 28.02.2019 Atlantic for 1 night(s) 1 single bed room(s)
without meals

outward journey: self organised transport

journey back: self organised transport

no transfer

price: 100.00 EUR

trip inquiry 2 for 2 person(s)

check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
with breakfast, parking included

outward journey: self organised transport

journey back: self organised transport

no transfer

price: 90.00 EUR

HOTLINE TRAVEL AGENCY

0 end

1 new trip

2 search trip

3 show all trip offers

your choice: 2

number of trip: 4

trip inquiry 4 for 1 person(s)

check-in: hotel 28.02.2019 Atlantic for 1 night(s) 1 single bed room(s)
without meals

outward journey: self organised transport

journey back: self organised transport

no transfer

price: 100.00 EUR

(d)delete or 🗑ot: d

destructor hotel Atlantic at 28.02.2019 for 1 guests done

destructor SelfOrganised done

destructor Transport done

destructor SelfOrganised done

destructor Transport done

destructor Trip done

HOTLINE TRAVEL AGENCY

0 end

1 new trip

2 search trip

3 show all trip offers

your choice: 3

trip inquiry 2 for 2 person(s)

check-in: hotel 25.09.2019 Bavaria for 1 night(s) 1 double bed room(s)
with breakfast, parking included

outward journey: self organised transport

journey back: self organised transport

no transfer

price: 90.00 EUR

HOTLINE TRAVEL AGENCY

0 end

1 new trip

2 search trip

3 show all trip offers

your choice: