
PROJEKT 1

Zarządzanie podróżami

1. Stwórz następującą strukturę klas:

- City
- Travel
 - AirTravel
 - NormalFlight
 - GovernmentFlight
 - RailwayTravel
 - TrainRide
 - RoadTravel
 - CarRide
 - BusRide
- Interfejs – Reservable
Implementowany przez NormalFlight, TrainRide, CarRide, BusRide
- Interfejs – Rentable
Implementowany przez GovernmentFlight i CarRide

2. Do klas dodaj następujące pola. Można dodać też inne, ale te wypisane tutaj są obowiązkowe:

- City: name, country, timeZone
- Travel: startTime, duration (w minutach), startCity, endCity, seats, occupiedSeats
- AirTravel: planeNumber
- GovernmentFlight: hirerName
- RailwayTravel: trainNumber
- RoadTravel: registrationNumber
- CarRide: hirerName

UWAGA! W klasach mogą znajdować się również inne pola i metody. Również w projekcie mogą znajdować się dodatkowe klasy.

3. Skompiluj z klasą Main (z klasy nie można nic usuwać, ale można dodawać).

4. Dodaj metody w interfejsach i zaimplementuj je w odpowiednich miejscach:

- Reservable: reserve(int numberOfSeats)
- Rentable: rent(String name)

5. Dodaj metodę w klasie Travel dodaj metodę wypisując dane o podróży i godzinę przyjazdu. Jeśli godzina jest inna w strefie docelowej i w strefie wyjazdowej to należy podać dwie.

6. Dodaj nowy wyjątek NoFreeSeatsException, który ma być rzucony przy próbie rezerwacji, gdy brak miejsc i obsłuż go w Main tak aby cały Main działał poprawnie.

7. Dodaj w klasie Travel następujące metody:

- isNationalTravel – zwracającą wartość logiczną
- doesChangeDate – sprawdzającą czy na miejscu (czasu docelowego) będziemy innego dnia niż wyruszyliśmy

8. Dodaj metody w City:

- isTheSameCountry(City city2)
 - isTheSameTime(City city2) – może się zdarzyć, że dwa miasta będą w różnych strefach czasowych, ale mimo wszystko będą miały tą samą godzinę
9. Dodaj typ wliczeniowy TravelStatus z wartościami: COMPLETED, IN_PROGRESS, NOT_STARTED
 10. Dodaj do TravelStatus metodę getStatus, która zwraca status w postaci stringa.
 11. Dodaj do klasy Travel metodę getTravelStatus zwracającą jedną z wartości TravelStatus.
 12. W klasie Main zaimplementuj metodę printAllTravels. Metoda ta ma wypisać najpierw podróże zakończone, później te w trakcie a na koniec podróże jeszcze nie rozpoczęte.
 13. Przetestuj metody isTheSameCountry i isTheSameTime z klasy City.
 14. Przetestuj metody isNationalTravel i doesChangeDate z klasy Travel.
 15. Przetestuj metodę reserve z klasy Travel.

KLASA MAIN:

```
public class Main {

    public static void main(String[] args) {

        City[] cities = new City[5];
        cities[0] = new City("Krakow", "Poland", ZoneId.of("Europe/Warsaw"));
        cities[1] = new City("Warsaw", "Poland", ZoneId.of("Europe/Warsaw"));
        cities[2] = new City("Berlin", "Germany", ZoneId.of("Europe/Berlin"));
        cities[3] = new City("Moskwa", "Russia", ZoneId.of("Europe/Moscow"));
        cities[4] = new City("Rejkiawik", "Iceland", ZoneId.of("Iceland"));

        Travel[] travels = new Travel[5];
        LocalDateTime now = LocalDateTime.now();
        travels[0] = new BusRide(now.plusHours(1), 120, cities[0], cities[1], 60, 10, "EL99999");
        travels[1] = new CarRide(now.minusMonths(1), 390, cities[0], cities[1], 5, 5, "WW88888");
        travels[2] = new GovernmentFlight(now.minusMinutes(10), 165, cities[0], cities[1], 0, 0,
"F123");
        travels[3] = new NormalFlight(now.minusMinutes(30), 240, cities[0],
cities[1], 200, 0, "F456");
        travels[4] = new TrainRide(now.plusHours(2), 1200, cities[3], cities[2], 1000, 10, "T2365");
        //można dodać więcej

        printAllTravels(travels);
        for(int i=0; i<travels.length; i++)
            if(travels[i] instanceof Reservable)
                ((Reservable)travels[i]).reserve(10);
            ((Rentable)travels[2]).rent("Kowalski");

        System.out.println("The same time (" + cities[2].getName() + ", " + cities[1].getName() + "): "
+ cities[2].isTheSameCountry(cities[1]));
        System.out.println("The same country (" + cities[2].getName() + ", " + cities[1].getName() + "): "
+ cities[2].isTheSameTime(cities[1]));

        printAllTravels(travels);

        System.out.println("The date change - train ride: " + travels[4].doesChangeDate());
    }
}
```

```
        System.out.println("National - bus ride: "+travels[0].isNational());
    }
}
```

PRZYKŁAD WYKONANEGO MAIN:

*****Complete*****

Car ride:

Registration Number: ww88888

Seats: 5/5

START: Krakow 2023-08-29T16:34:51.481906

END: Warsaw 2023-08-29T23:04:51.481906 (2023-08-29T23:04:51.481906)

Hirer: null

*****In progress*****

Goverment flight:

Plane number: F123

Seats: 0/0

START: Krakow 2023-09-29T16:24:51.481906

END: Warsaw 2023-09-29T19:09:51.481906 (2023-09-29T19:09:51.481906)

Hirer: null

Normal flight:

Plane number: F456

Seats: 0/200

START: Krakow 2023-09-29T16:04:51.481906

END: Warsaw 2023-09-29T20:04:51.481906 (2023-09-29T20:04:51.481906)

*****Not started*****

Bus ride:

Registration Number: EL99999

Seats: 10/60

START: Krakow 2023-09-29T17:34:51.481906

END: Warsaw 2023-09-29T19:34:51.481906 (2023-09-29T19:34:51.481906)

Train ride:

Train number: T2365

Seats: 10/1000

START: Moskwa 2023-09-29T18:34:51.481906

END: Berlin 2023-09-30T13:34:51.481906 (2023-09-30T14:34:51.481906)

No seats

The same country (Berlin,warsaw): false

The same country (Berlin,warsaw): true

*****Complete*****

Car ride:

Registration Number: ww88888

Seats: 5/5

START: Krakow 2023-08-29T16:34:51.481906

END: Warsaw 2023-08-29T23:04:51.481906 (2023-08-29T23:04:51.481906)

Hirer: null

*****In progress*****

Government flight:

Plane number: F123

Seats: 0/0

START: Krakow 2023-09-29T16:24:51.481906

END: Warsaw 2023-09-29T19:09:51.481906 (2023-09-29T19:09:51.481906)

Hirer: Kowalski

Normal flight:

Plane number: F456

Seats: 10/200

START: Krakow 2023-09-29T16:04:51.481906

END: Warsaw 2023-09-29T20:04:51.481906 (2023-09-29T20:04:51.481906)

*****Not started*****

Bus ride:

Registration Number: EL99999

Seats: 20/60

START: Krakow 2023-09-29T17:34:51.481906

END: Warsaw 2023-09-29T19:34:51.481906 (2023-09-29T19:34:51.481906)

Train ride:

Train number: T2365

Seats: 20/1000

START: Moskwa 2023-09-29T18:34:51.481906

END: Berlin 2023-09-30T13:34:51.481906 (2023-09-30T14:34:51.481906)

The same date - train ride: true

National - bus ride: true