

Dokumentacja Projektu

Temat 5

Treść Zadania:

System rezerwacji biletów pewnej firmy lotniczej przyjmuje żądania rezerwacji od klientów indywidualnych i firm-pośredników. Firma posiada flotę wielu samolotów kilku typów (każdy o innej liczbie miejsc i zasięgu) i obsługuje różne tras między lotniskami.

Minimalny zakres funkcjonalności:

- a. zarządzanie samolotami i klientami (dodawanie, usuwanie, przegląd)
- b. zarządzanie trasami i lotniskami (dodawanie, usuwanie, przegląd)
- c. mechanizm generowania lotów (dobór samolotu w zależności od odległości, uwzględnienie czasu podróży oraz powrotu, powielanie lotów na wiele dni - codziennych, cotygodniowych) na podstawie trasy
- d. funkcjonalność rezerwacji biletów
- e. zapis i odczyt stanu systemu na dysk

Diagram UML:

Zapisany został w oddzielnym pliku PDF

Opis projektu :

Projekt pozwala na stworzenie systemu rezerwacji biletów pewnej firmy lotniczej. Przyjmuje żądania rezerwacji lotów od osób indywidualnych jak i od firm pośredników. Projekt umożliwia dodawanie, usuwanie, i przegląd: samolotów , klientów, tras i lotnisk. Posiada możliwość generowania i przeglądania lotów oraz ich rezerwacje. Dane klientów, lotnisk, samolotów, tras oraz lotnisk przechowywane są w plikach typu tmp – wczytywanie i zapis do nich przebiega automatycznie. Generowanie lotu odbywa się na podstawie utworzonej wcześniej trasy, wybranej przez użytkownika samolot wybierany jest automatycznie do długości trasy.

Działanie programu:

Po uruchomieniu programu mamy do wyboru 6 opcji:

- Loty
- Samoloty
- Trasy i Lotniska
- Klienci
- Rezerwacja biletu
- Wyjdz

Aby wybrać opcję należy wpisywać numerki odpowiadające opcjom.

1:

Po wybraniu opcji Loty możemy wybrać 4 opcje

1.mamy opcję dodania lotu (należy podać datę oraz godzinę odlotu i przylotu oraz trasę z której chcemy stworzyć lot)

2.usuwania lotu

3.przeglądania lotów

0.Powrót do menu głównego

2:

W opcji Samoloty możemy:

1.Dodać samolot(należy wybrać typ samolotu z listy wpisując jego nazwę oraz jego numer który ma się znajdować na boku pojazdu)

2.Usun samolot(usuwa samolot po nazwie)

3.Przeglądaj samoloty(wyświetla listę samolotów)

0.Powrót do menu głównego

3:

W opcji Trasy i Lotniska możemy:

1.Dodaj trasę(podaje się 2 nazwy lotnisk oraz długość trasy)

2.Usun trasę(usuwanie trasy po podaniu odpowiedniego numerku)

3.Przeglądaj trasy(wyświetla listę tras)

4.Dodaj lotnisko(dodanie lotniska poprzez podanie jego nazwy)

5.Usun lotnisko(aby usunąć lotnisko należy podać jego nazwę)

6.Przeglądaj lotnisko(wyświetla listę lotnisk)

0.Powrót do menu głównego

4:

W opcji Klienci możemy:

1.Dodaj Klienta(wybieramy czy chcemy dodać klienta indywidualnego lub jako firmę pośredniczą)

2.Usun Klienta(usuwa klienta indywidualnego poprzez podanie numeru PESEL lub firmę pośredniczą poprzez podanie nr KRS)

3.Przeglądaj Klientów(wyświetla listę klientów)

0.Powrót do menu głównego

5:

W opcji rezerwacja biletu możemy:

1.Dodaj bilet(wybieramy typ klienta i podajemy odpowiednio PESEL lub KRS oraz wybieramy lot na który chcemy zarezerwować bilet)

2.Usun bilet(należy podać numer biletu który chcemy usunąć)

3.Przeglądaj bilety(wyświetla listę zarezerwowanych biletów)

0.Powrot do menu głównego

0:

Wychodzimy z programu

Opis funkcjonalności wybranych metod:

`public void` addClient(Client c)- dodaje klienta do listy

`public void` showClients()- wyświetla listę klientów

`public void` removeClient(String PESEL) – usuwa klienta z listy

`public void` showFlight()- wyświetla listę lotów

`public void` addAirport(Airport a1)- dodaje lotnisko do listy

`public void` showAirport()- wyświetla listę lotnisk

`public void` removeAirport(String airportName)- usuwa lotnisko z listy

`public void` addAirplane(Airplane p)- dodaje samolot

`public void` showAirplane()- wyświetla listę samolotów

`public void` removeAirplane(String name, int range, int numberOfSeats, boolean status)- usuwa samolot

`public void` addRoute(Route r) - dodaje trasę

`public void` showRoute()- wyświetla listę tras

`public void` removeRoute(int numberOfRouteToRemove)- usuwa trasę

`public void` addTicket(Ticket t)- dodaje bilet do listy

`public void` removeTicket(int numberOfTicketToRemove)- usuwa bilet z listy

`public void` showTickets() - wyświetla listę biletów

Opis wybranej biblioteki:

Użyte biblioteki:

```
java.util.ArrayList;  
java.util.Scanner;  
java.time.LocalDate;  
java.time.LocalDateTime;  
java.util.InputMismatchException;  
java.util.Collections;
```

```
java.util.Comparator;  
java.util.Objects;;  
java.util.Scanner;  
java.io.*;  
  
java.time.LocalDate;
```

Klasa [LocalDate](#), która pozwala na pracę z datami jako obiektami. Takie podejście pozwala operować na datach jak na obiektach, na które można odejmować dodawać i porównywać.

Najważniejsze metody:

- `atStartOfDay()` - zwraca pełną datę wraz z północą (00:00)
- `parse()` - waliduje i tworzy obiekt `LocalDate`.
- `getDayOfMonth()` - zwraca dzień liczbowo (int)
- `getMonthValue()` - zwraca miesiąc liczbowo (int)
- `getYear()` - zwraca rok liczbowo (int)
- `isAfter()` - sprawdza czy data jest po jakiejś dacie
- `isBefore()` - sprawdza czy data jest przed jakąś datą

Link do dokumentacji:

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>

Podział ról:

Jakub Mieczkowski:

- zapis do pliku
- klasa `Flight`
- klasa `IntermediaryCompany`
- klasa `AirportCompany`
- klasa `Route`
- klasa `Ticket`
- funkcjonalność rezerwowania biletów

Marcin Marciniak:

- diagram UML
- dokumentacja
- funkcjonalność menu
- klasa `Client` oraz `IndividualClient`
- Klasa `Airplane`, `Airbus`, `Boeing`, `Tupolew`

Projekt został napisany w języku JAVA, w środowisku IntelliJ IDEA od firmy JetBrains zainstalowanym na systemie Windows 10.

