2

**Assignment #2**

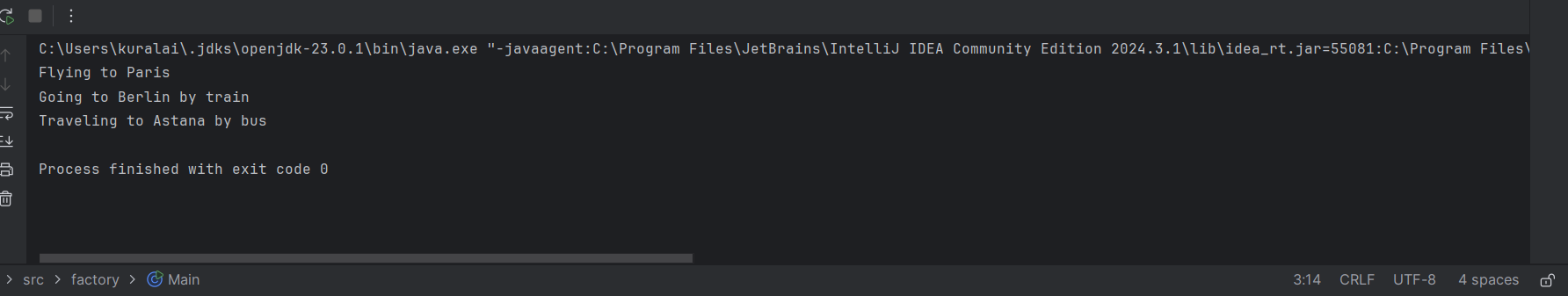
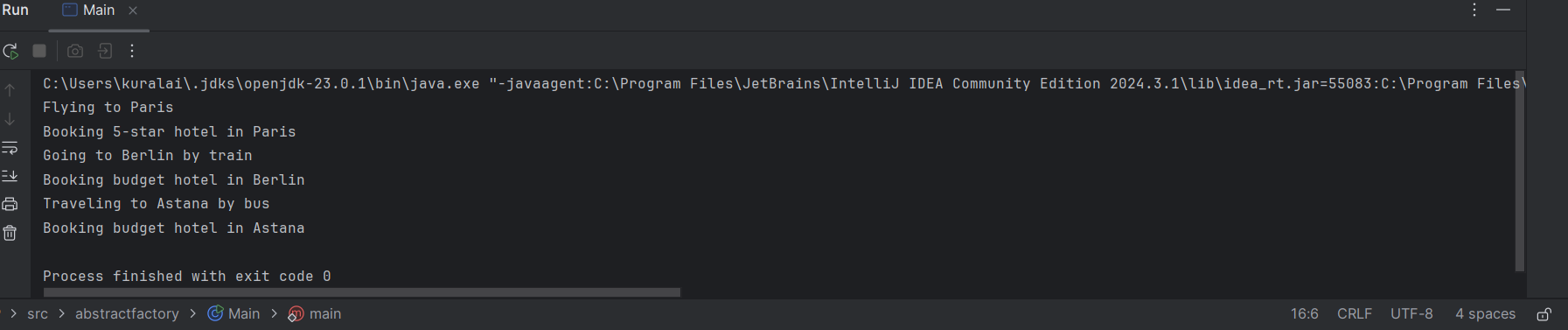
**Software Design Patterns**

**Factory Pattern**

**Student name: Quralai Baqytnur**

**Group: SE-2429**

**Practice Teacher: Mukhidenov Diyar**

**Factory   
  
**In this code I created an abstract class TravelAgency with method createTransport().  
Subclasses (AirAgency, RailAgency, BusAgency) decide which transport is used.  
The method organizeTrip() books the trip and calls the transport.  
Example: AirAgency - plane, RailAgency - train, BusAgency - bus.  
Factory Method is about creating one product, here it is transport.  
 **Abstract Factory  
  
**In this code I created an interface TravelFactory with two methods: createTransport() and createHotel().  
Different factories (AirTravelFactory, RailTravelFactory, BusTravelFactory) create both transport and hotel together.  
Example: AirTravelFactory - plane + luxury hotel, RailTravelFactory - train + budget hotel.  
Abstract Factory is about creating a family of products that match together.

**Comparison:** Factory Method → simpler, one product (transport).  
Abstract Factory → more complex, creates group of products (transport + hotel).

**Summary:** Both patterns make code flexible and reusable.  
Factory Method is good for simple cases, Abstract Factory for bigger systems.