**public class Attraction** {

// gνωρισματα

private int attractionId; // μοναδικο αναγνωριστικο αξιοθεατου

private String name; // ονομα αξιοθεατου

private String description; // περιγραφή

private String location; // τοποθεσια π.χ., διεύθυνση, πόλη

private double rating; // βαθμολογια

// Constructor

public Attraction(int attractionId, String name, String description, String location, double rating) {

this.attractionId = attractionId;

this.name = name;

this.description = description;

this.location = location;

this.rating = rating;

}

// Getters

public int getAttractionId() {

return attractionId;

}

public String getName() {

return name;

}

public String getDescription() {

return description;

}

public String getLocation() {

return location;

}

public double getRating()

{

return rating;

}

// για μεθοδους - η λογικη θα προστεθε μετα

public void displayDetails() {

System.out.println("Attraction.displayDetails() placeholder called.");

}

}

**public class Route** {

private int routeId; // μοναδικο αναγνωριστικο διαδρομης

private String name; // ονομα διαδρομης

private String description; // περιγραφη της διαδρομης

private double estimatedDurationHours; // εκτιμωμενη διαρκεια

private double cost; // κοστος

private String difficultyLevel; // επιπεδο δυσκολιας

// Associations - μια διαδρομή μπορεί να περιέχει πολλά αξιοθέατα

private List<Attraction> attractions;

// Constructor

public Route(int routeId, String name, String description, double estimatedDurationHours, double cost, String difficultyLevel) {

this.routeId = routeId;

this.name = name;

this.description = description;

this.estimatedDurationHours = estimatedDurationHours;

this.cost = cost;

this.difficultyLevel = difficultyLevel;

this.attractions = new ArrayList<>(); // Αρχικοποίηση λίστας αξιοθέατων

}

// Getters

public int getRouteId() {

return routeId;

}

public String getName() {

return name;

}

public String getDescription() {

return description;

}

public double getEstimatedDurationHours() {

return estimatedDurationHours;

}

public double getCost() {

return cost;

}

public String getDifficultyLevel() {

return difficultyLevel;

}

public List<Attraction> getAttractions() {

return attractions;

}

// για μεθόδους - η λογική θα προστεθεί μετα

public void addAttraction(Attraction attraction) {

System.out.println("Route.addAttraction() placeholder called.");

this.attractions.add(attraction);

}

public void displayRouteDetails() {

System.out.println("Route.displayRouteDetails() placeholder called.");

}

}

**public class PickupPoint** {

private int pointId; // μοναδικό αναγνωριστικο σημείου

private String name; // ονομα σημείου

private String address; // διεύθυνση σημείου

private String city; // πόλη

// Constructor

public PickupPoint(int pointId, String name, String address, String city) {

this.pointId = pointId;

this.name = name;

this.address = address;

this.city = city;

}

// Getters

public int getPointId() {

return pointId;

}

public String getName() {

return name;

}

public String getAddress() {

return address;

}

public String getCity() {

return city;

}

// για μεθόδους - η λογική θα προστεθεί μετα

public void displayPickupPointDetails() {

System.out.println("PickupPoint.displayPickupPointDetails() placeholder called.");

}

}

**public class ExternalTravelService** {

private int serviceId; // μοναδικό αναγνωριστικο υπηρεσίας

private String serviceName; // ονομα υπηρεσίας )

private String apiUrl; // URL του API της υπηρεσίας

// Constructor

public ExternalTravelService(int serviceId, String serviceName, String apiUrl) {

this.serviceId = serviceId;

this.serviceName = serviceName;

this.apiUrl = apiUrl;

}

// Getters

public int getServiceId() {

return serviceId;

}

public String getServiceName() {

return serviceName;

}

public String getApiUrl() {

return apiUrl;

}

//για μεθόδους

public List<HotelOption> searchHotels(String destination, String checkInDate, String checkOutDate) {

// Χρειάζεται HotelOption

System.out.println("ExternalTravelService.searchHotels() placeholder called for destination: " + destination);

return new ArrayList<>(); // επιστροφή κενής λίστας

}

public String getBookingLink(int hotelId, String checkInDate, String checkOutDate)

System.out.println("ExternalTravelService.getBookingLink() placeholder called for hotelId: " + hotelId);

return "http://placeholder.booking.link"; // επιστροφή URL

}

}

**public class HotelOption** {

private int hotelId; // μνδκ αναγνωριστικό ξενοδοχείου

private String name; // ονομα ξενοδοχείου

private String address; // διεύθυνση

private double pricePerNight; // τιμή / διανυκτέρευση

private double rating; // βαθμολογία

// Constructor

public HotelOption(int hotelId, String name, String address, double pricePerNight, double rating) {

this.hotelId = hotelId;

this.name = name;

this.address = address;

this.pricePerNight = pricePerNight;

this.rating = rating;

}

// Getters

public int getHotelId() {

return hotelId;

}

public String getName() {

return name;

}

public String getAddress() {

return address;

}

public double getPricePerNight() {

return pricePerNight;

}

public double getRating() {

return rating;

}

// η λογική θα προστεθεί μετα

public void displayHotelDetails() {

System.out.println("HotelOption.displayHotelDetails() placeholder called.");

}}

**public class ProfileAnalyzer** {

private int analyzerId;

private String algorithmName;

private String lastRunDate; // για να ξέρουμε πότε τρέξαμε την ανάλυση

// Associations - Ενδεικτική σύνδεση, π.χ. με Customer ή User θα το δούμε μετα

// private Customer customer; // Αν αναλύει το προφίλ ενός συγκεκριμένου πελάτη

// Constructor

public ProfileAnalyzer(int analyzerId, String algorithmName) {

this.analyzerId = analyzerId;

this.algorithmName = algorithmName;

this.lastRunDate = "N/A"; // Αρχική τιμή

}

// Getters

public int getAnalyzerId() { return analyzerId; }

public String getAlgorithmName() { return algorithmName; }

public String getLastRunDate() { return lastRunDate; }

// Setters

public void setLastRunDate(String lastRunDate) {

this.lastRunDate = lastRunDate;

}

// για μεθόδους

public List<Route> analyzeUserPreferences(User user) { // Θα χρειαστεί User, Route

System.out.println("ProfileAnalyzer.analyzeUserPreferences() placeholder called for user: " + user.getUsername());

return new ArrayList<>(); // επιστροφή κενής λίστας

}

public void generateRecommendations(List<Attraction> allAttractions, List<Route> allRoutes) { // Θα χρειαστεί Attraction, Route

System.out.println("ProfileAnalyzer.generateRecommendations() placeholder called.");

}

}