Card.java

```
package metrocard;

public class Card {
    private String cardid;
    private int balance;

    public Card(String cardid, int balance) {
        this.cardid = cardid;
        this.balance = balance;
    }

    public String getCardId() {
        return cardid;
    }

    public int getBalance() {
        return balance;
    }

    public void setBalance(int balance) {
        this.balance = balance;
    }
}
```

MetroCardService.java

```java
package metrocard;

import java.util.*;

public class MetroCardService {
    Map<String, Card> cards = new HashMap<>();
    Map<String, Integer> stationCollection = new HashMap<>();
    Map<String, Integer> stationDiscounts = new HashMap<>();
    Map<String, Map<String, Integer>> stationPassengerCount = new HashMap<>();
    Map<String, String> lastStationVisited = new HashMap<>();

    public void addBalance(String cardid, int amount) {
        if (cards.containsKey(cardid)) {
            Card card = cards.get(cardid);
            card.setBalance(card.getBalance() + amount);
        } else {
            Card card = new Card(cardid, amount);
            cards.put(cardid, card);
        }
    }

    public void checkIn(String cardid, String station, String passengerType) {
        Card card = cards.get(cardid);
        int fare = 0;

        switch (passengerType.toUpperCase()) {
            case "ADULT": fare = 200; break;
            case "SENIOR_CITIZEN": fare = 100; break;
            case "KID": fare = 50; break;
        }

        boolean isReturn = station.equals(lastStationVisited.get(cardid));
        int discount = isReturn ? fare / 2 : 0;
        fare -= discount;

        if (card.getBalance() >= fare) {
            card.setBalance(card.getBalance() - fare);
            stationCollection.put(station, stationCollection.getOrDefault(station, 0) + fare);
            stationDiscounts.put(station, stationDiscounts.getOrDefault(station, 0) + discount);
        } else {
            int partial = card.getBalance();
            int penalty = 50;
            card.setBalance(0);
            stationCollection.put(station, stationCollection.getOrDefault(station, 0) + partial + penalty);
        }

        lastStationVisited.put(cardid, station);

        stationPassengerCount.putIfAbsent(station, new HashMap<>());
        Map<String, Integer> typeCount = stationPassengerCount.get(station);
        typeCount.put(passengerType.toUpperCase(), typeCount.getOrDefault(passengerType.toUpperCase(), 0) + 1);
    }

    public void printSummary() {
        for (String station : stationCollection.keySet()) {
            int total = stationCollection.getOrDefault(station, 0);
            int discount = stationDiscounts.getOrDefault(station, 0);

            System.out.println("TOTAL_COLLECTION " + station + " " + total + " " + discount);
            System.out.println("PASSENGER_TYPE_SUMMARY");

            Map<String, Integer> counts = stationPassengerCount.getOrDefault(station, new HashMap<>());

            counts.entrySet().stream()
```

```
                .sorted((e1, e2) -> {
                    if (e1.getValue().equals(e2.getValue())) {
                        return e1.getKey().compareTo(e2.getKey());
                    }
                    return e2.getValue().compareTo(e1.getValue());
                })
                .forEach(entry -> {
                    System.out.println(entry.getKey() + " " + entry.getValue());
                });
        }
    }
}
```

CommandProcessor.java

```
package metrocard;

public class CommandProcessor {
    private static MetroCardService service = new MetroCardService();

    public static void process(String line) {
        String[] parts = line.split(" ");
        String command = parts[0];

        switch (command) {
            case "BALANCE":
                service.addBalance(parts[1], Integer.parseInt(parts[2]));
                break;
            case "CHECK_IN":
                service.checkIn(parts[1], parts[2], parts[3]);
                break;
            case "PRINT_SUMMARY":
                service.printSummary();
                break;
        }
    }
}
```

Main.java

```
package metrocard;

import java.io.*;

public class Main {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new FileReader("input.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                CommandProcessor.process(line);
            }
        } catch (IOException e) {
            System.out.println("Error reading input file: " + e.getMessage());
        }
    }
}
```

input.txt

```
BALANCE CARD1 500
CHECK_IN CARD1 AIRPORT ADULT
CHECK_IN CARD1 AIRPORT ADULT
BALANCE CARD2 200
CHECK_IN CARD2 CENTRAL SENIOR_CITIZEN
PRINT_SUMMARY
```