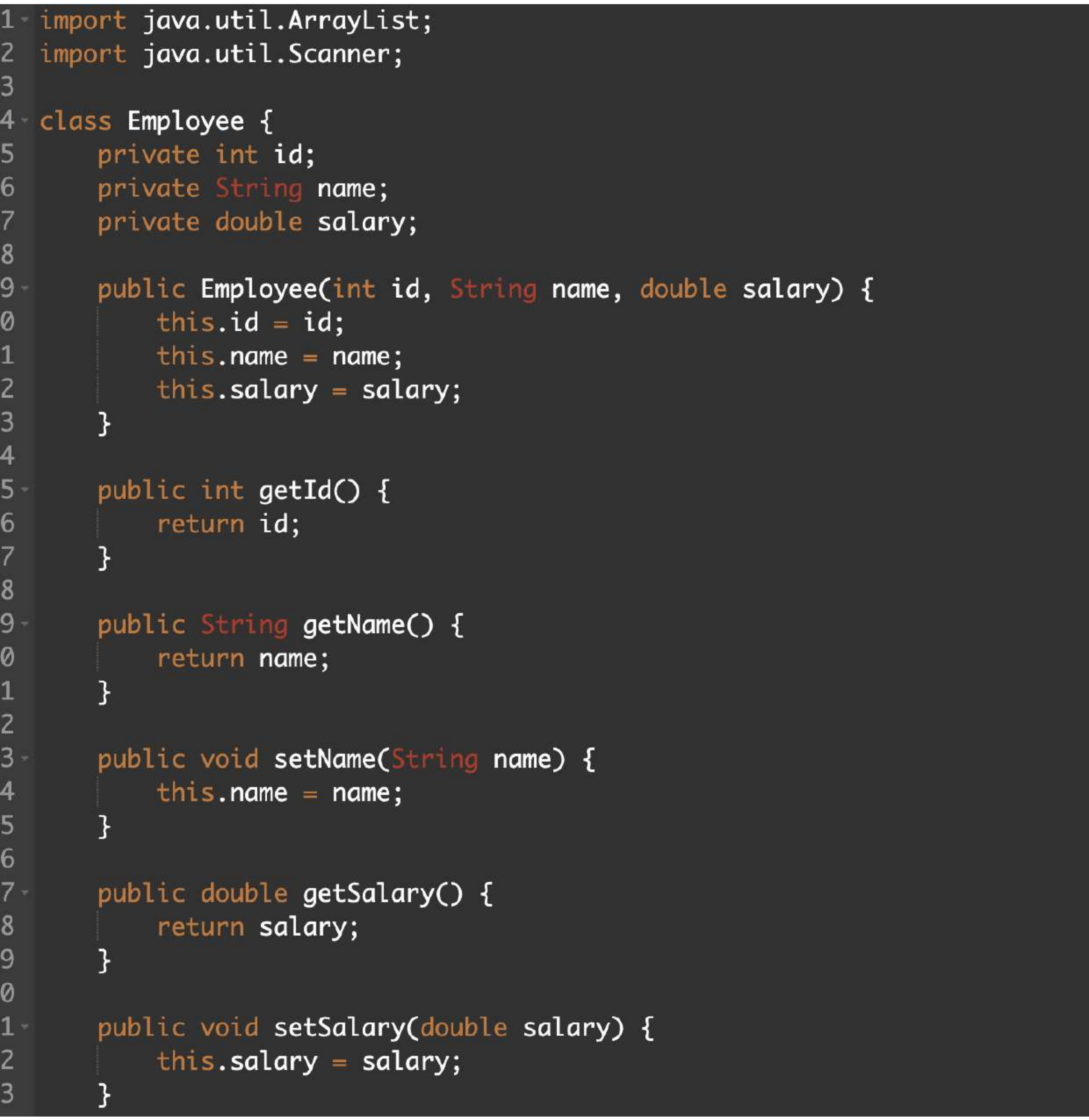
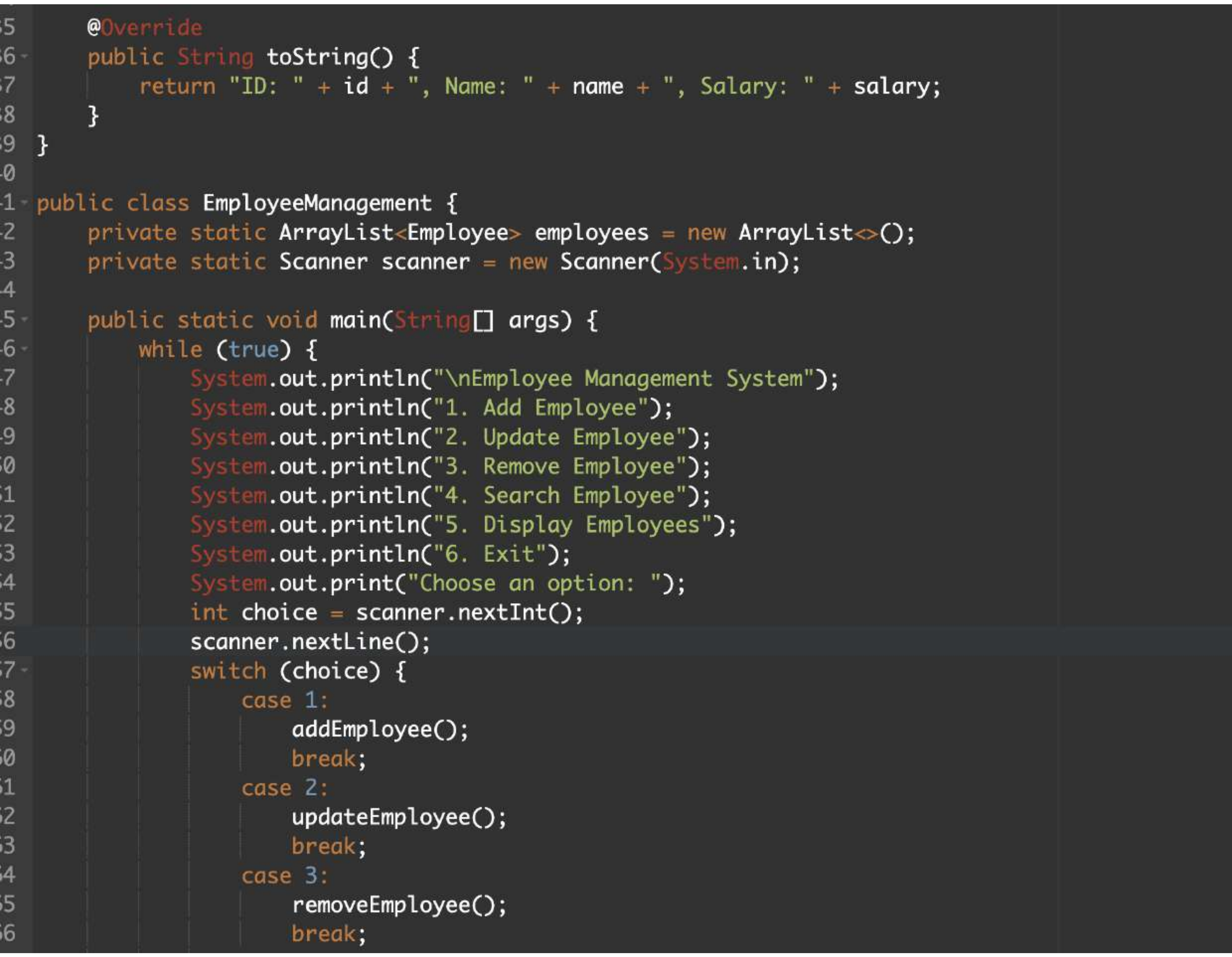
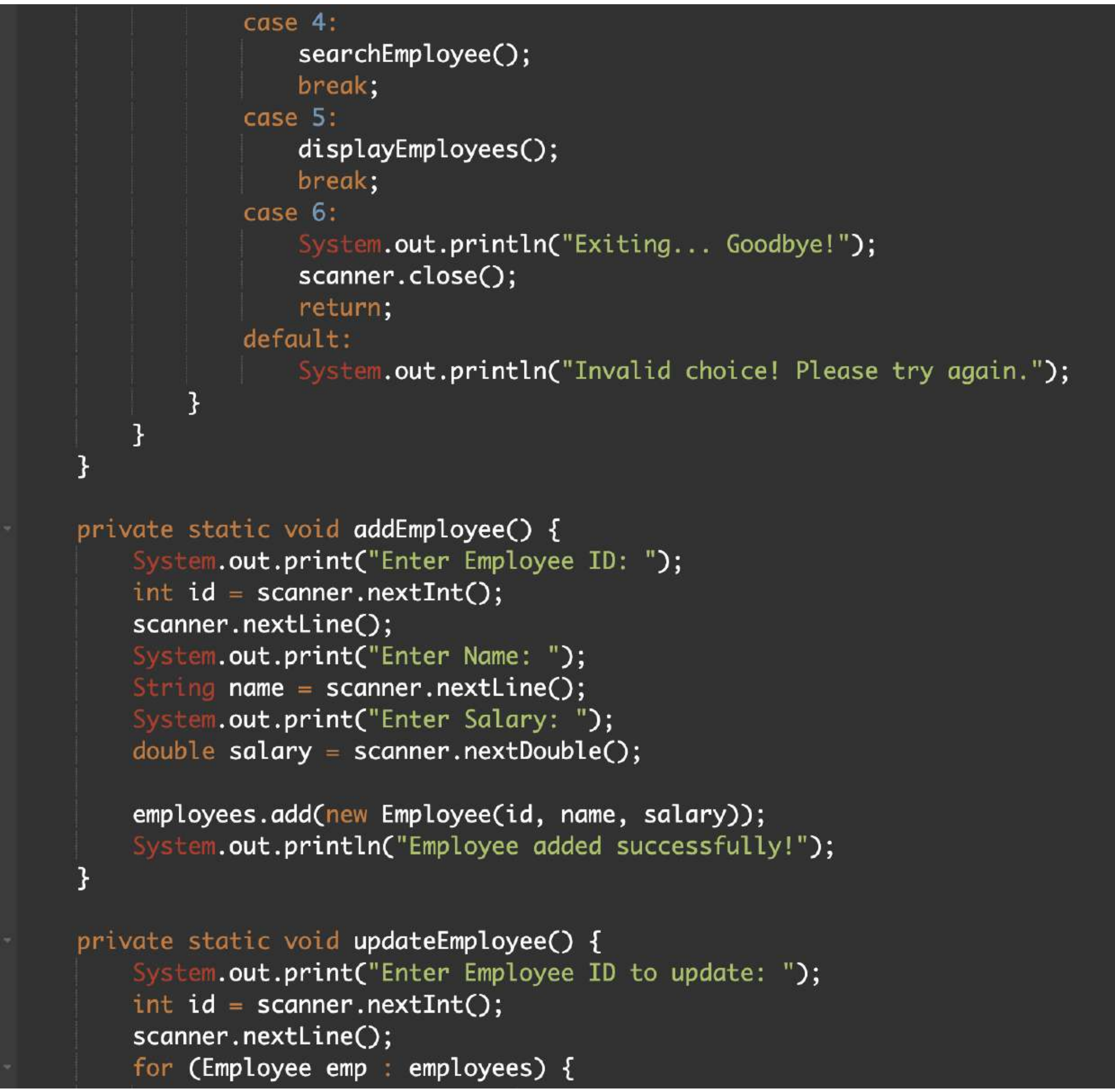


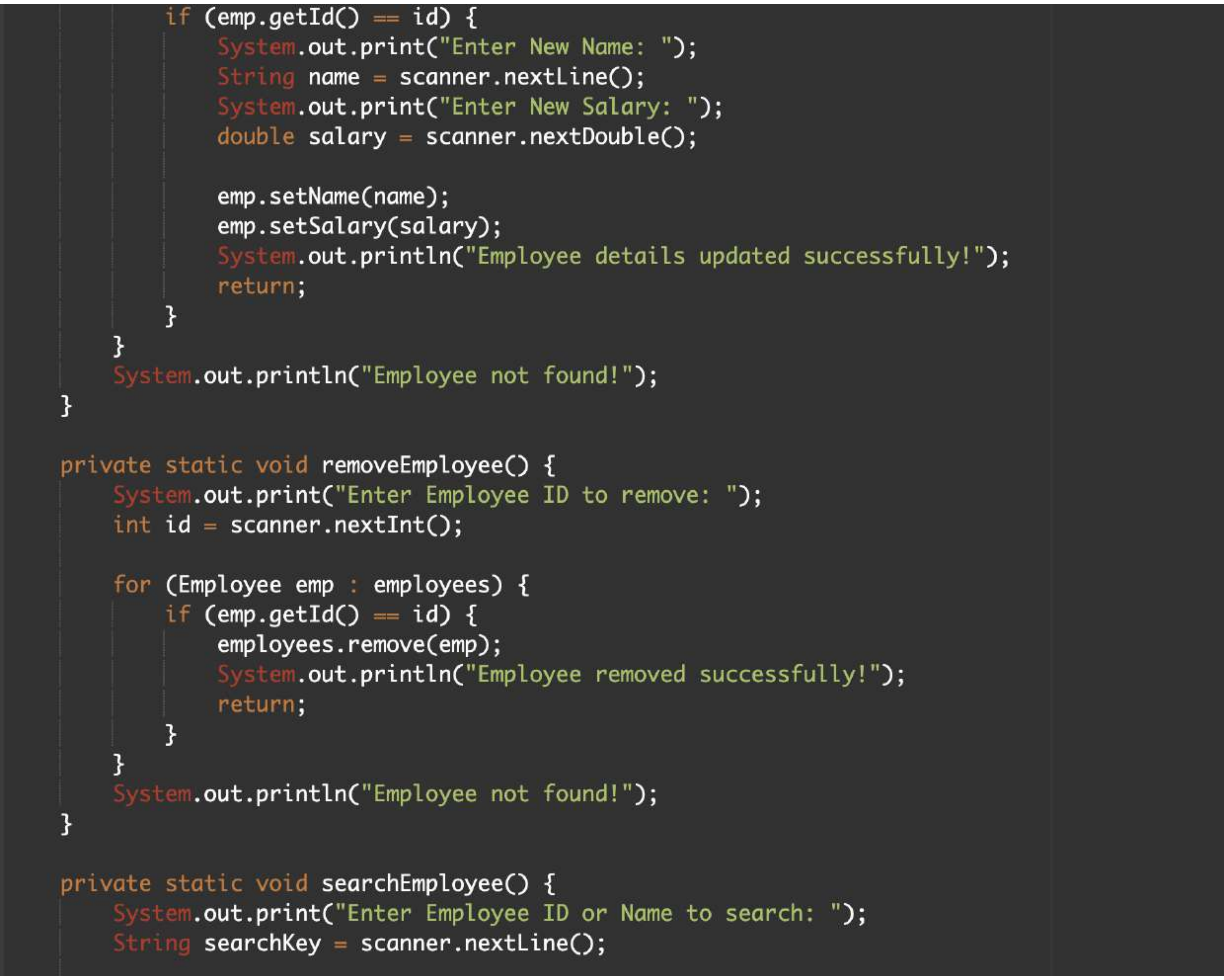
```java
import java.util.ArrayList;
import java.util.Scanner;

class Employee {
    private int id;
    private String name;
    private double salary;

    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }
```



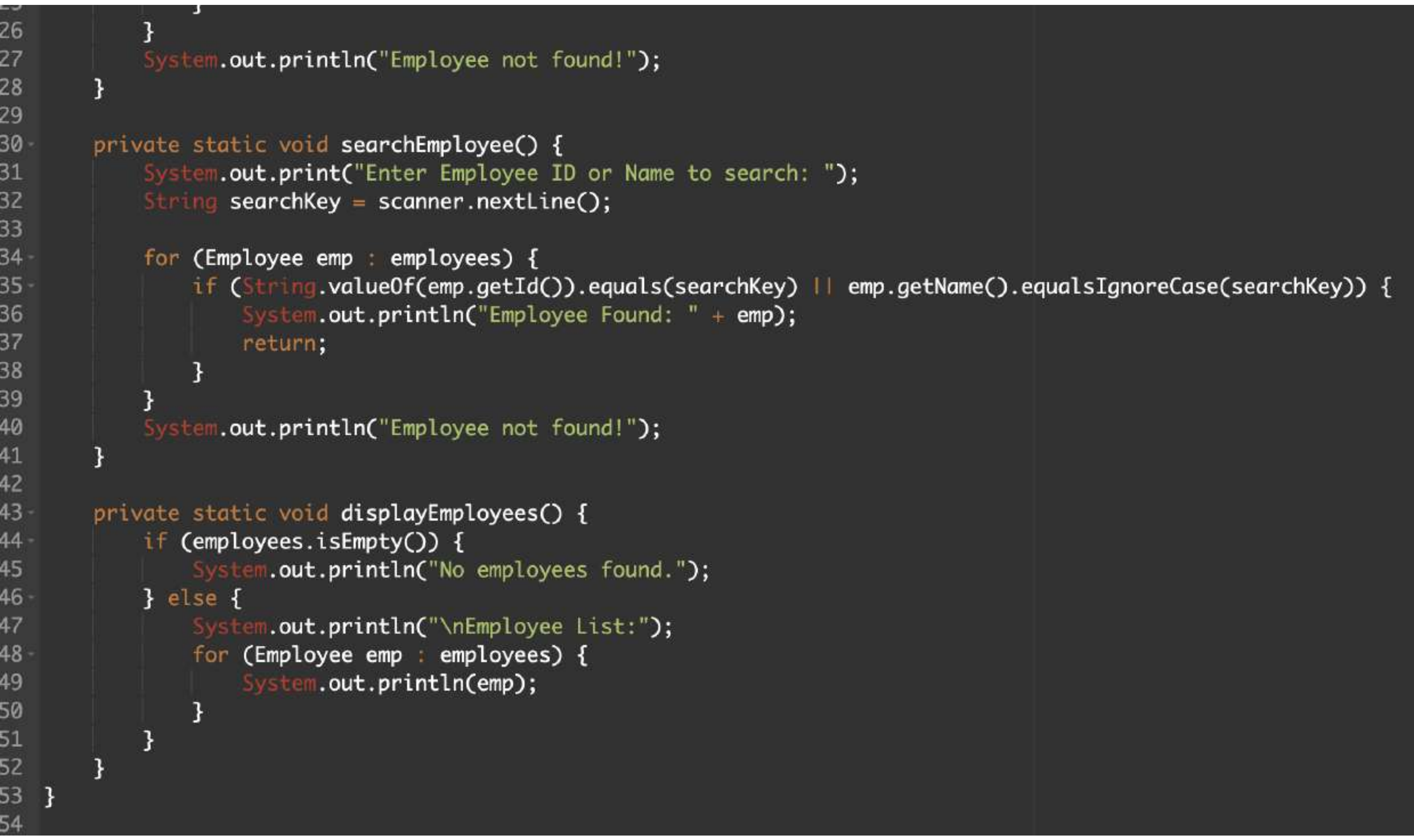
```java
    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Salary: " + salary;
    }
}

public class EmployeeManagement {
    private static ArrayList<Employee> employees = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nEmployee Management System");
            System.out.println("1. Add Employee");
            System.out.println("2. Update Employee");
            System.out.println("3. Remove Employee");
            System.out.println("4. Search Employee");
            System.out.println("5. Display Employees");
            System.out.println("6. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                    addEmployee();
                    break;
                case 2:
                    updateEmployee();
                    break;
                case 3:
                    removeEmployee();
                    break;
```



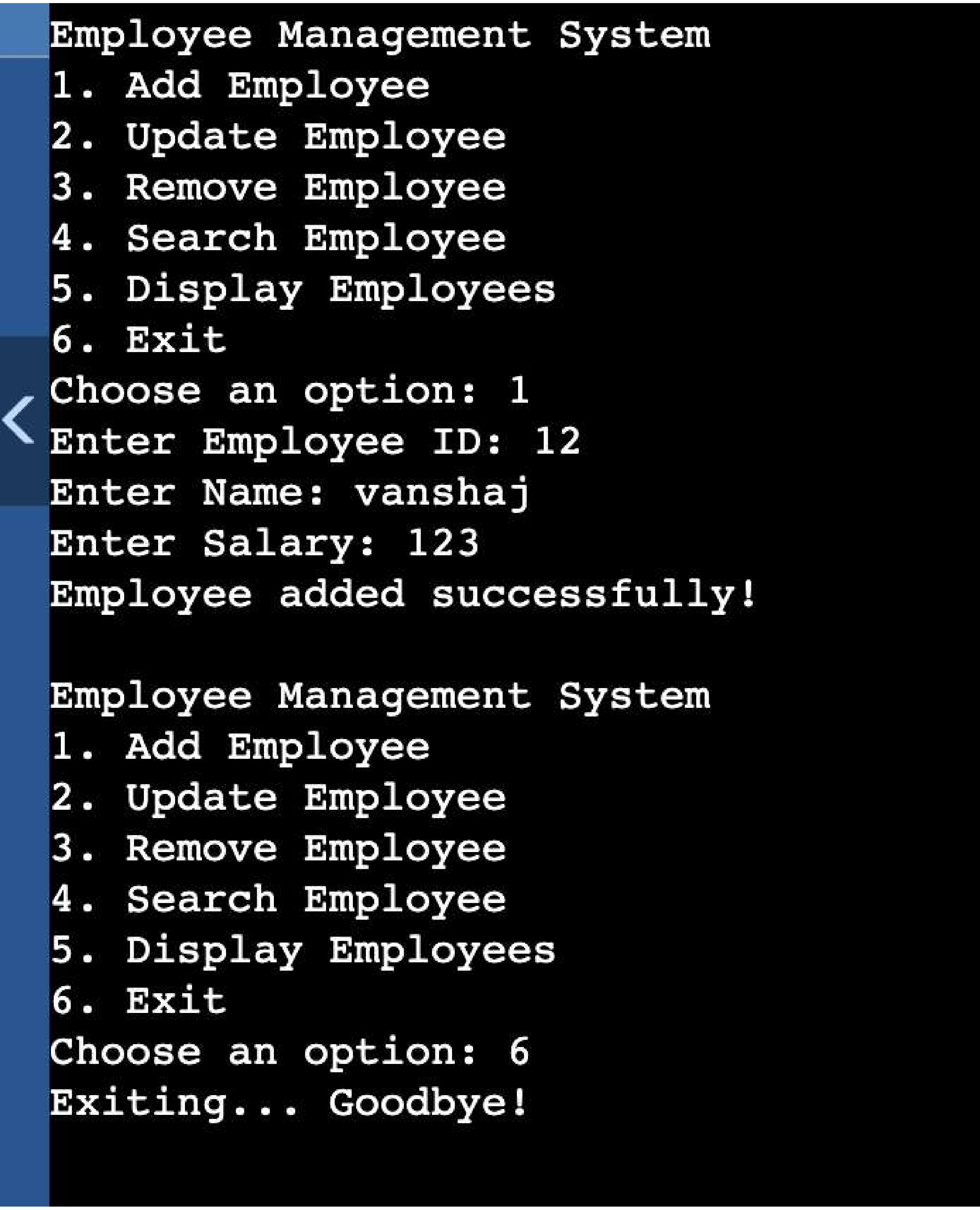
```java
                case 4:
                    searchEmployee();
                    break;
                case 5:
                    displayEmployees();
                    break;
                case 6:
                    System.out.println("Exiting... Goodbye!");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice! Please try again.");
            }
        }
    }

    private static void addEmployee() {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Salary: ");
        double salary = scanner.nextDouble();

        employees.add(new Employee(id, name, salary));
        System.out.println("Employee added successfully!");
    }

    private static void updateEmployee() {
        System.out.print("Enter Employee ID to update: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        for (Employee emp : employees) {
```



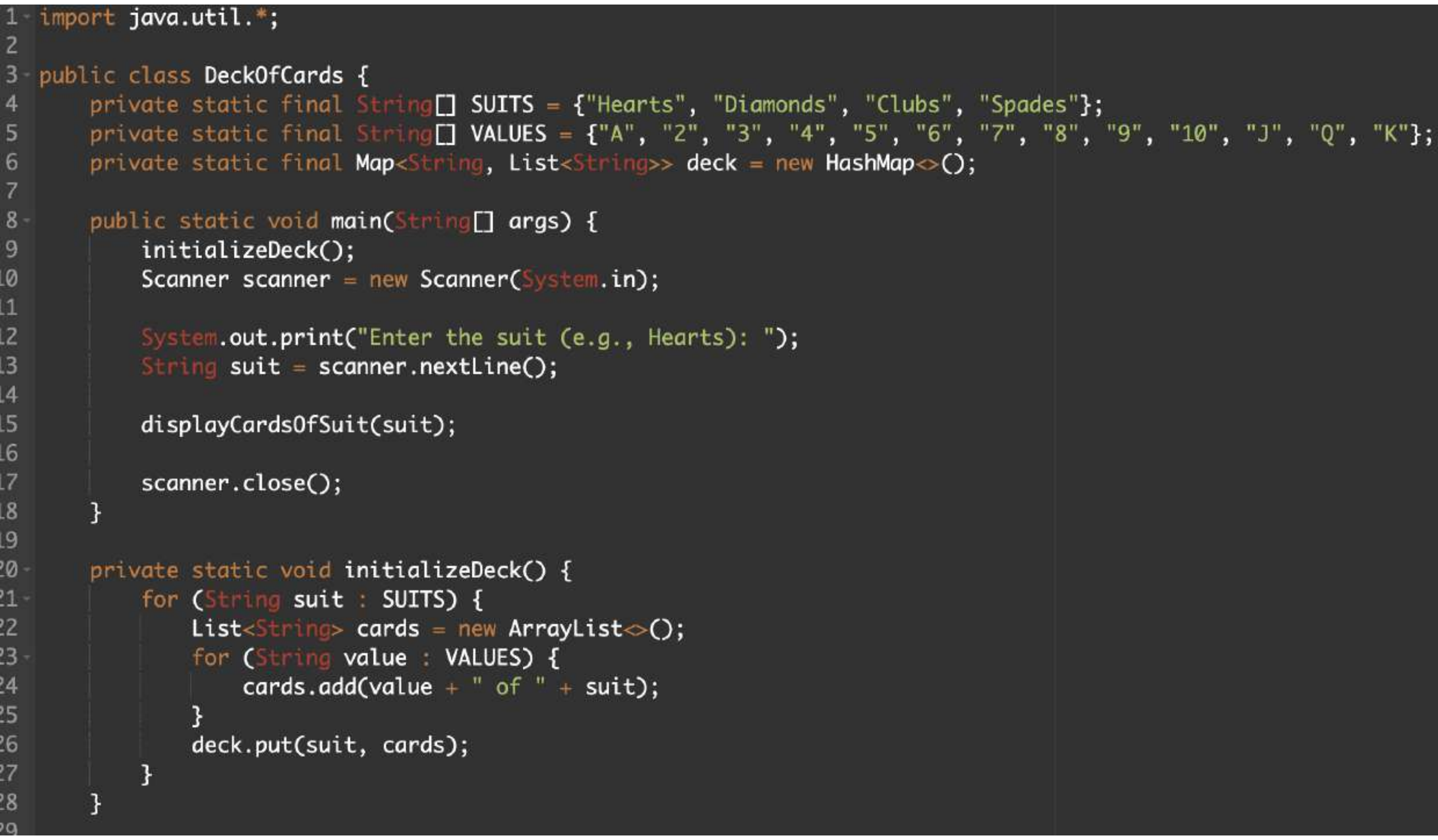
```
            if (emp.getId() == id) {
                System.out.print("Enter New Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter New Salary: ");
                double salary = scanner.nextDouble();

                emp.setName(name);
                emp.setSalary(salary);
                System.out.println("Employee details updated successfully!");
                return;
            }
        }
        System.out.println("Employee not found!");
    }

    private static void removeEmployee() {
        System.out.print("Enter Employee ID to remove: ");
        int id = scanner.nextInt();

        for (Employee emp : employees) {
            if (emp.getId() == id) {
                employees.remove(emp);
                System.out.println("Employee removed successfully!");
                return;
            }
        }
        System.out.println("Employee not found!");
    }

    private static void searchEmployee() {
        System.out.print("Enter Employee ID or Name to search: ");
        String searchKey = scanner.nextLine();
```



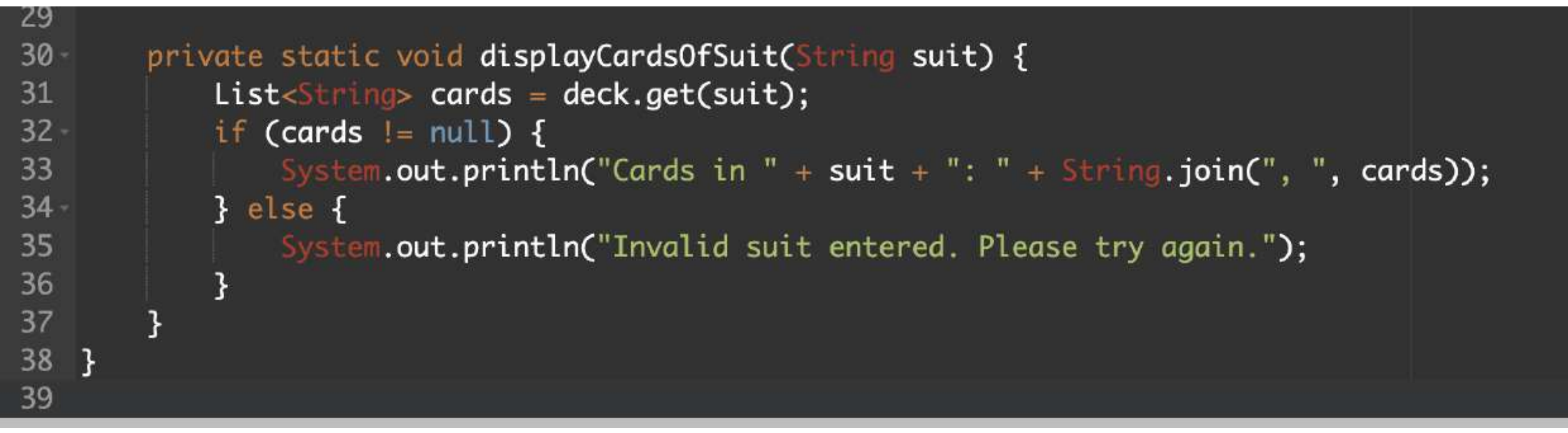
```java
            }
        }
        System.out.println("Employee not found!");
    }

    private static void searchEmployee() {
        System.out.print("Enter Employee ID or Name to search: ");
        String searchKey = scanner.nextLine();

        for (Employee emp : employees) {
            if (String.valueOf(emp.getId()).equals(searchKey) || emp.getName().equalsIgnoreCase(searchKey)) {
                System.out.println("Employee Found: " + emp);
                return;
            }
        }
        System.out.println("Employee not found!");
    }

    private static void displayEmployees() {
        if (employees.isEmpty()) {
            System.out.println("No employees found.");
        } else {
            System.out.println("\nEmployee List:");
            for (Employee emp : employees) {
                System.out.println(emp);
            }
        }
    }
}
```



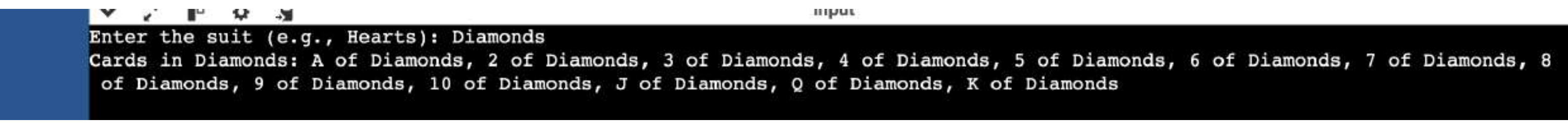
```
Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display Employees
6. Exit
Choose an option: 1
Enter Employee ID: 12
Enter Name: vanshaj
Enter Salary: 123
Employee added successfully!

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display Employees
6. Exit
Choose an option: 6
Exiting... Goodbye!
```



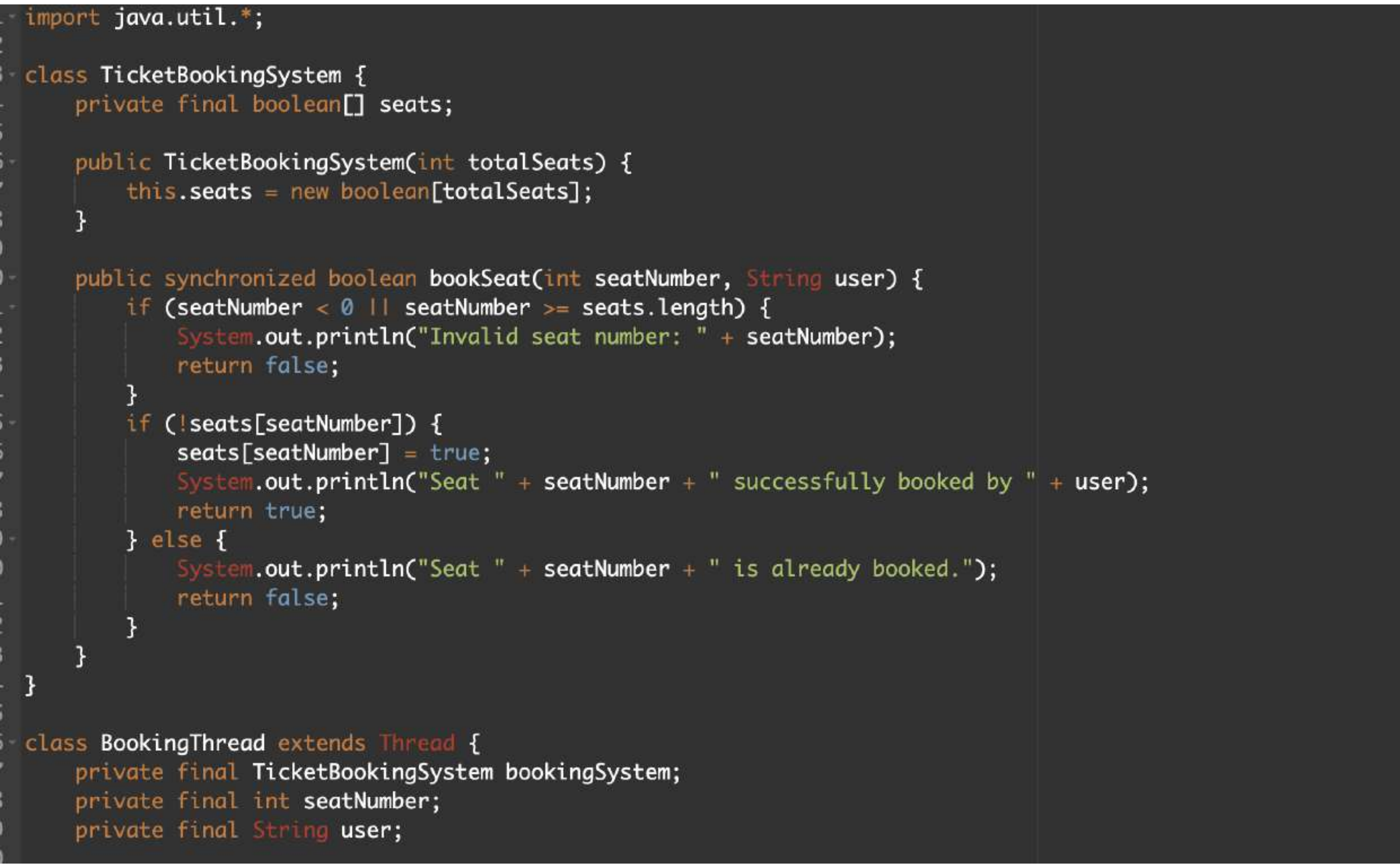
```java
import java.util.*;

public class DeckOfCards {
    private static final String[] SUITS = {"Hearts", "Diamonds", "Clubs", "Spades"};
    private static final String[] VALUES = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
    private static final Map<String, List<String>> deck = new HashMap<>();

    public static void main(String[] args) {
        initializeDeck();
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the suit (e.g., Hearts): ");
        String suit = scanner.nextLine();

        displayCardsOfSuit(suit);

        scanner.close();
    }

    private static void initializeDeck() {
        for (String suit : SUITS) {
            List<String> cards = new ArrayList<>();
            for (String value : VALUES) {
                cards.add(value + " of " + suit);
            }
            deck.put(suit, cards);
        }
    }
```



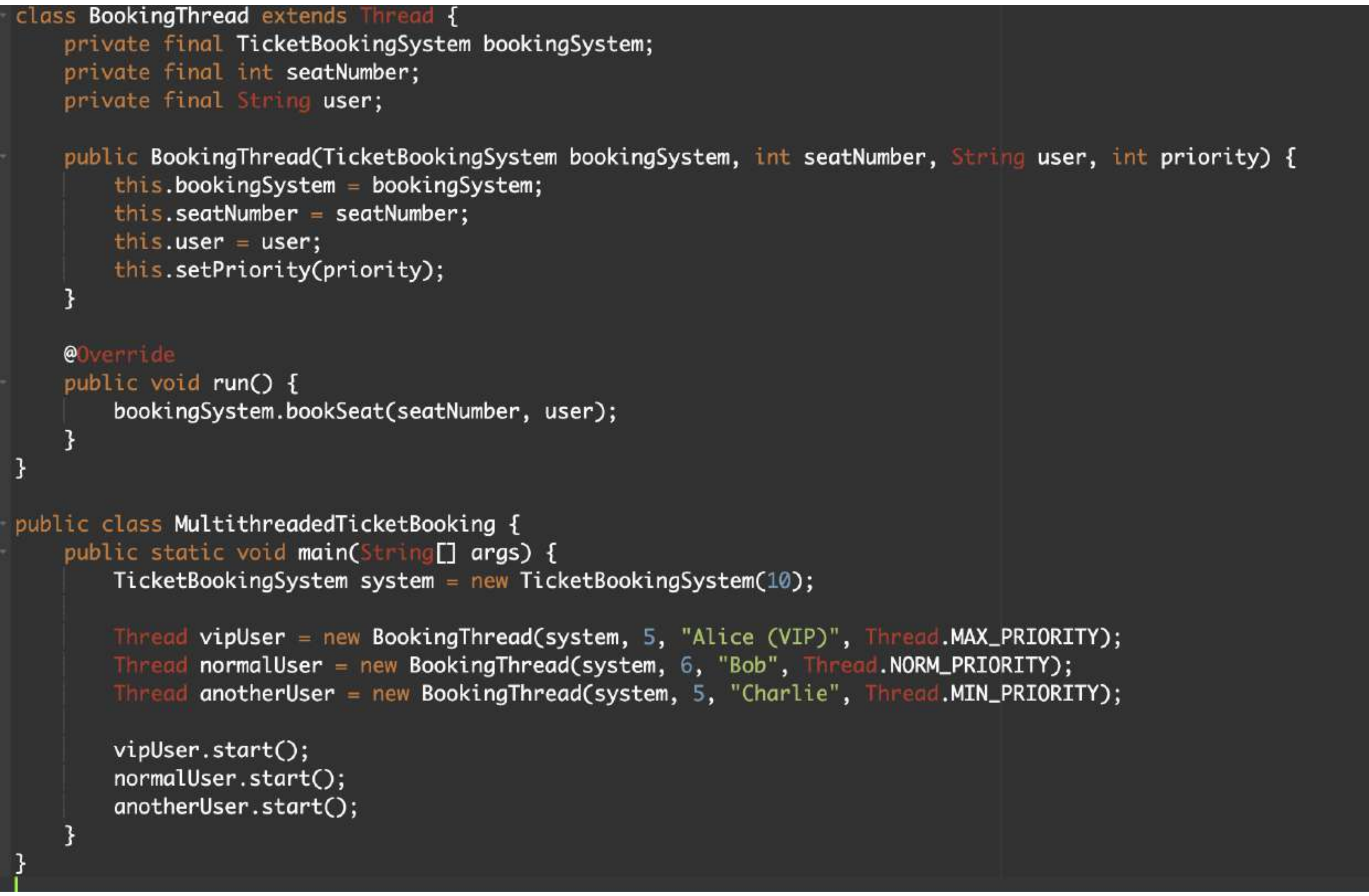
```java

    private static void displayCardsOfSuit(String suit) {
        List<String> cards = deck.get(suit);
        if (cards != null) {
            System.out.println("Cards in " + suit + ": " + String.join(", ", cards));
        } else {
            System.out.println("Invalid suit entered. Please try again.");
        }
    }
}

```



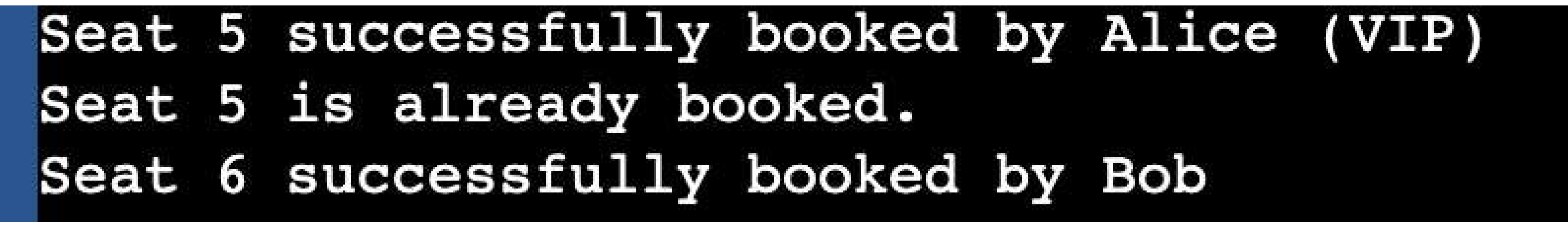
```
Enter the suit (e.g., Hearts): Diamonds
Cards in Diamonds: A of Diamonds, 2 of Diamonds, 3 of Diamonds, 4 of Diamonds, 5 of Diamonds, 6 of Diamonds, 7 of Diamonds, 8
 of Diamonds, 9 of Diamonds, 10 of Diamonds, J of Diamonds, Q of Diamonds, K of Diamonds
```



```java
import java.util.*;

class TicketBookingSystem {
    private final boolean[] seats;

    public TicketBookingSystem(int totalSeats) {
        this.seats = new boolean[totalSeats];
    }

    public synchronized boolean bookSeat(int seatNumber, String user) {
        if (seatNumber < 0 || seatNumber >= seats.length) {
            System.out.println("Invalid seat number: " + seatNumber);
            return false;
        }
        if (!seats[seatNumber]) {
            seats[seatNumber] = true;
            System.out.println("Seat " + seatNumber + " successfully booked by " + user);
            return true;
        } else {
            System.out.println("Seat " + seatNumber + " is already booked.");
            return false;
        }
    }
}

class BookingThread extends Thread {
    private final TicketBookingSystem bookingSystem;
    private final int seatNumber;
    private final String user;
```



```java
class BookingThread extends Thread {
    private final TicketBookingSystem bookingSystem;
    private final int seatNumber;
    private final String user;

    public BookingThread(TicketBookingSystem bookingSystem, int seatNumber, String user, int priority) {
        this.bookingSystem = bookingSystem;
        this.seatNumber = seatNumber;
        this.user = user;
        this.setPriority(priority);
    }

    @Override
    public void run() {
        bookingSystem.bookSeat(seatNumber, user);
    }
}

public class MultithreadedTicketBooking {
    public static void main(String[] args) {
        TicketBookingSystem system = new TicketBookingSystem(10);

        Thread vipUser = new BookingThread(system, 5, "Alice (VIP)", Thread.MAX_PRIORITY);
        Thread normalUser = new BookingThread(system, 6, "Bob", Thread.NORM_PRIORITY);
        Thread anotherUser = new BookingThread(system, 5, "Charlie", Thread.MIN_PRIORITY);

        vipUser.start();
        normalUser.start();
        anotherUser.start();
    }
}
```



```
Seat 5 successfully booked by Alice (VIP)
Seat 5 is already booked.
Seat 6 successfully booked by Bob
```