Inside of the airplane

CMPS 347 Data Structures

Hasan Atwi – Batoul Hamzi

1. Abstract

1.1 -Project Statement:

Our project is based on using 2 data structures to add people to the airplane having in each class several methods (adding a passenger, revealing information of a passenger, counting the number of passengers). In this project we tried to use the most important methods we learned throughout this course to make this project successful.

1.2 Accomplished Objectives:

We aimed to use queue for security checking because of it is structure that fits the topic knowing that the passengers stand in the line to get checked before getting on the airplane. Also we used the binary search tree for inserting passengers in the airplane and revealing some information about the passengers where we decided that it would be more efficient that using other data structures.

2.1 Overview:

The aim of this project is to see who are the people that are eligible to get on the airplane in the most efficient way possible.

And making it easy to retrieve information about passengers.

2.2 Problem Statement:

We created this project to make the adding of the passengers and organizing information more sufficient instead of writing everything as in checking who registered before who, who is eligible do get on the airplane, and when is the airplane full. But instead programming helps us save a lot of time, money and number of employees.

Person

-name: String
-age: int
-left: Person
-right: Person
-next: Person
-dayRegistered: int
-monthRegistered: int
-yearRegistered: int
-rightToTravel: boolean
-securityScreening: boolean
-luggageWeight: double
-checkIfDisabled: boolean
-isBusiness: boolean
-minimumPay: int
-extraPay: boolean
-military: boolean
-inGoodHealth: boolean

+setters and getters
+Person()
+Person(name: String)
_display():void

LinkedList

#head: Person
#tail: Person
#length: int

+setters and getters
+LinkedList()
+append(P: Person): void
+prepend(newPerson: Person): void
+removeLast(): Person
+get(index: int): Person
+remove(index: int): Person
+removeFirst(): Person

SecurityLine

-max: int

+SecurityLine()
+setters and getters
+insertAtBack(P: Person):void
+removeFromFront():Person
+removeKPerson(K: int):void
+printSecurityLine():void
+isEmpty(): boolean
+clearSecurityLine():void

Airplane
-firstPassenger: Person -deadlineDay: int -deadlineMonth: int -deadlineYear: int -MaxWeight: double -MaxSeats: int -currentNumberOfSeats: int
+getters and setters +Airplane() +isEmpty(): boolean +checkIfOnTime(P: Person): boolean +checkIfEnoughSeats():boolean +checkWeight(P: Person): boolean +CanGetOnPlane(P : Person): boolean +checkIfP1lessThanP2(P1: Person,P2: Person): boolean +addToAirplane(P: Person): boolean +height(P : Person): int +getheight(): int +displayAsBST(): void +displayAsBST(P: Person, level: int): void +countPassengers(): int +countPassengers(P: Person): int +displayOldPassengers(): void +displayOldPassengers(P: Person,level: int):void +displayPassengers(): void +displayPassengers(P: Person,level: int): void +search(P: Person): boolean

+removePerson(P: Person): void
+minValue(currentNode: Person):Person

General explanation of what we did:

We made a class called person which acts as a node that has many attributes which represent the information of each person and based on these information it will be decided weather the person will be eligible to get on the plane or not.

We implemented a linked list which is just a helper for the queue class where it has the basic methods (append, prepend, removeFirst...) those methods help directly in achieving the queue class when inserting at back and removing from front.

After the person gets checked in the security line his security attribute is being set and based on his other attributes he/she will be able or not able to get on the plane. Here comes the role of the airplane class where we use it just to represent a certain section in the airplane (business, economy,....) this class has several methods to check if the person can get on the airplane or not. Also has some methods to retrieve some information concerning the passengers in an efficient way.

Description of methods in the classes:

1-Person Class:

Display(): displays the information of the person using its attributes.

2-LinkedList Class:

Append(P: Person): adds a person at the end of the linked list.

Prepend(newPerson: Person): adds a person at the beginning of the linked list.

removeLast():removes the last person in the linked list.

Get(index: int): gets the person that has the following index in the linked list.

Remove(index: int): removes a person of a specific index in the linked list.

removeFirst():removes the first person in the linked list.

3-SecurityLine Class:

insertAtBack(P: Person): adds a person at the end of the security line.

removeKPerson(K: int): removes a person of a specific index in the security line.

removeFromFront():removes the first person in the security line.

printSecurityLine():prints the security line.

isEmpty(): returns true if the security line is empty and false otherwise.

clearSecurityLine(): makes the security line empty.

4-Airplane Class:

isEmpty(): returns true if the airplane is empty and false otherwise.

CheckIfOnTime(P: Person): checks if the person has registered before the deadline (return true) otherwise returns false.

checkIfEnoughSeats():returns true if the current number of seats is less than the maximum number of seats and false otherwise.

checkWeight(P: Person): check if the person has luggage which is less that the maximum weight he is permitted to carry.

CanGetOnPlane(P: Person): returns true if the person satisfies all the conditions to get on the plane and false otherwise.

checkIfP1lessThanP2(P1: Person,P2 : Person): returns true if P1 registerd before P2 and false otherwise.

addToAirplane(P: Person): adds a person to the airplane if he/she is eligible to get on it.

Getheight(): calls the method height(P :Person) to get the height of the binary tree that represents this specific section of the airplane.

displayAsBST(): calls the method displayAsBST(P: Person,level: int) to display this section as a binary search tree.

countPassengers(): calls the method countPassengers(P : Person) to count the number of passengers in the section of the airplane.

displayOldPassengers(): calls the method displayOldPassengers(P : Person,level: int) to display the information of the old passengers in this section.

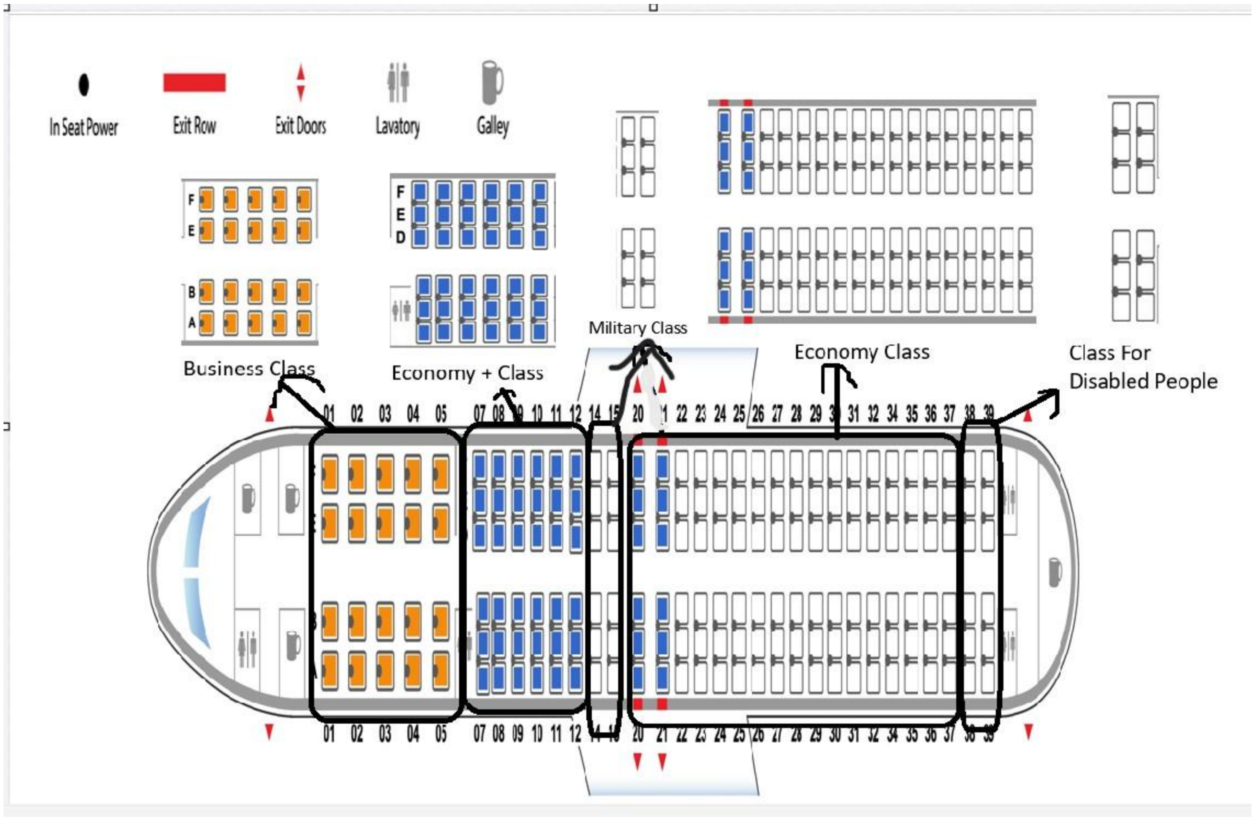
displayPassengers(): calls the method displayPassengers(P : Person,level: int) to display the information of the passengers in this section.

Search(P : Person): returns true if the person is present on the airplane and false otherwise.

removePerson(P: Person): removes a person from the airplane.

minValue(currentNode: Person): starting from this person finds the passenger who was the latest to register(in the binary search tree).

The airplane is divided as follows:



Code:

1-Person class:

```
package Project;

public class Person {
	private String name;
	private int age;
	private Person left;
	private Person right;
	private Person next;
	private int dayRegistered;
	private int monthRegistered;
```

```java
private int yearRegistered;
private boolean rightToTravel;
private boolean securityScreening;
private double luggageWeight;
private boolean checkIfDisabled;
private boolean isBusiness;
private int minimumPay;
private boolean extraPay;
private boolean military;
private boolean inGoodHealth;


public Person() {
}

public Person(String name) {
    this.name = name;
}

public Person getNextPerson() {
    return next;
}

public void setNextPerson(Person P) {
    next = P;
}

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

public void setAge(int age) {
    this.age = age;
}

public void setInGoodHealth(boolean value) {
    this.inGoodHealth = value;
}

public void setLeft(Person left) {
        this.left = left;
    }

    public void setRight(Person right) {
        this.right = right;
    }

    public void setDayRegistered(int dayRegistered) {
        this.dayRegistered = dayRegistered;
    }

    public void setMonthRegistered(int monthRegistered)
{
        this.monthRegistered = monthRegistered;
    }

    public void setYearRegistered(int yearRegistered) {
        this.yearRegistered = yearRegistered;
    }

    public void setRightToTravel(boolean rightToTravel)
{
        this.rightToTravel = rightToTravel;
    }

    public void setSecurityScreening(boolean
securityScreening) {
        this.securityScreening = securityScreening;
    }

    public void setLuggageWeight(double luggageWeight)
{
        this.luggageWeight = luggageWeight;
    }

    public void setCheckIfDisabled(boolean
checkIfDisabled) {
        this.checkIfDisabled = checkIfDisabled;
    }
    public void setPersonClass(int N) {
        if(N==1)
        this.isBusiness = true;
        else
            this.isBusiness = false;
    }

    public void setMinimumPay(int minimumPay) {
        this.minimumPay = minimumPay;
    }

    public void setExtraPay(boolean extraPay) {
        this.extraPay = extraPay;
    }

    public void setMilitary(boolean military) {
        this.military = military;
    }

    public boolean inGoodHealth() {
        return inGoodHealth;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public Person getLeft() {
        return left;
    }

    public Person getRight() {
        return right;
    }

    public int getDayRegistered() {
```

```
    return dayRegistered;
}

public int getMonthRegistered() {
    return monthRegistered;
}

public int getYearRegistered() {
    return yearRegistered;
}

public boolean isRightToTravel() {
    return rightToTravel;
}

public boolean securityScreening() {
    return securityScreening;
}

public double LuggageWeight() {
    return luggageWeight;
}

public boolean CheckIfDisabled() {
    return checkIfDisabled;
}

public boolean getBusinessClass() {
    return isBusiness;
}

public int getMinimumPay() {
    return minimumPay;
}

public boolean isExtraPay() {
    return extraPay;
}

public boolean isMilitary() {
        return military;
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Registered Date: " +
dayRegistered + "/" + monthRegistered + "/" +
yearRegistered);
        System.out.println("Right to Travel: " +
rightToTravel);
        System.out.println("Security Checked?: " +
securityScreening);
        System.out.println("Luggage Weight: " +
luggageWeight);
        System.out.println("Is he/she disabled? " +
checkIfDisabled);
        System.out.println("Business Class: " +
isBusiness);
        System.out.println("Minimum Pay: " +
minimumPay);
        System.out.println("Did he/she pay extra?: " +
extraPay);
        System.out.println("Military: " + military);
    }
}
```

2-LinkedList Class:

```
package Project;

public class LinkedList {
    protected Person head;
    protected Person tail;
    protected int length;

    public LinkedList() {

    }

    public Person getHead() {
```

```
    return head;
}

public Person getTail() {
    return tail;
}

public int getLength() {
    return length;
}
public void append(Person P)
{
    if(length==0)
    {
        head=P;
        tail=P;
    }
    else
    {
        tail.setNextPerson(P);
        tail=P;
    }
    length++;
}
public void prepend(Person newPerson) {
    if (length == 0) {
        head = newPerson;
        tail = newPerson;
    } else {
        newPerson.setNextPerson(head);
        head = newPerson;
    }
    length++;
}

public Person removeLast() {
    Person temp = head;
    Person pre = head;
    if (length == 0)
        return null;
    else {
        while (temp.getNextPerson() != null) {
            pre = temp;

temp.setNextPerson(temp.getNextPerson());
        }
        tail = pre;
        temp.setNextPerson(null);
        length--;
        if (length == 0) {
            head = null;
            tail = null;
        }
        return temp;
    }
}

public Person get(int index) {
    if (index < 0 || index >= length)
        return null;
    Person temp = head;
    for (int i = 0; i < index; i++) {
        temp=(temp.getNextPerson());
    }
    return temp;
}

public Person remove(int index) {
    index--;
    if (index < 0 || index >= length)
        return null;
    if (index == 0)
        return removeFirst();
    else if (index == length - 1)
        return removeLast();
    else {
        Person pre = get(index - 1);
        Person temp = pre.getNextPerson();
        pre.setNextPerson(temp.getNextPerson());
        length--;
```

```
            return temp;
        }
    }

    public Person removeFirst() {
        Person temp=head;
        if (length == 0)
            System.out.println("The line is empty");
        else
            if(length==1)
            {
                head=null;
                tail=null;
            }
        else {
            head=head.getNextPerson();
            length--;
        }
            return temp;
}
}
```

3-SecurityLine:

```
package Project;

import java.util.Random;

public class SecurityLine extends LinkedList {
    Random random = new Random();
    private int max;

    public SecurityLine() {
        super();
    }

    public void setMax(int max) {
        this.max = max;
    }

    public void insertAtBack(Person P) {
        System.out.println(P.getName()+" joined the
security line");
        super.append(P);
    }

    public Person removeFromFront() {
        Random random = new Random();
        int randomNumber = random.nextInt(10);
        int threshold = 9;
        boolean result = randomNumber < threshold;
        if (result)

    System.out.println(super.getHead().getName()+" has
passed the security test");
        else

    System.out.println(super.getHead().getName()+" has
not passed the security test");
        super.getHead().setSecurityScreening(result);
        return super.removeFirst();
    }

    public void removeKPerson(int K) {
        super.remove(K);
    }

    public Person getFirst() {
        return super.getHead();
    }

    public Person getLast() {
        return super.getTail();
    }

    public int getLength() {
        return super.getLength();
    }

    public int getMax() {
        return max;
    }

    public void printSecurityLine() {
        if (!this.isEmpty()) {
            Person current = this.getFirst();
            while (current.getNextPerson() != null) {
                System.out.print(current.getName() + "
(Age: " + current.getAge() + ") <--- ");
                current = current.getNextPerson();
            }
            System.out.print(current.getName() + "
(Age: " + current.getAge() + ")");
        } else
            System.out.println("The security line is
empty");
    }

    public boolean isEmpty() {
        return super.length == 0;
    }

    public void clearSecurityLine() {
        super.head = null;
        super.head = null;
        length = 0;
    }
}
```

Airplane Class:

```
package Project;

public class Airplane {
    private Person firstPassenger;
    private int deadlineDay;
    private int deadlineMonth;
    private int deadlineYear;
    private double MaxWeight;
    private int MaxSeats;
    private int currentNumberOfSeats;
    public Person getFirstPassenger() {
        return firstPassenger;
    }

    public void setFirstPassenger(Person
firstPassenger) {
        this.firstPassenger = firstPassenger;
    }

    public int getDeadlineDay() {
        return deadlineDay;
    }

    public void setDeadlineDay(int deadlineDay) {
        this.deadlineDay = deadlineDay;
    }

    public int getDeadlineMonth() {
        return deadlineMonth;
    }

    public void setDeadlineMonth(int deadlineMonth) {
        this.deadlineMonth = deadlineMonth;
    }

    public int getDeadlineYear() {
        return deadlineYear;
    }

    public void setDeadlineYear(int deadlineYear) {
        this.deadlineYear = deadlineYear;
    }

    public double getMaxWeight() {
        return MaxWeight;
    }

    public void setMaxWeight(int maxWeight) {
        MaxWeight = maxWeight;
    }
    public int getMaxSeats() {
        return MaxSeats;
    }

    public void setMaxSeats(int maxSeats) {
        MaxSeats = maxSeats;
    }

    public int getCurrentNumberOfSeats() {
        return currentNumberOfSeats;
    }

    public void setCurrentNumberOfSeats(int
currentNumberOfSeats) {
        this.currentNumberOfSeats =
currentNumberOfSeats;
    }

    public Airplane() {

    }

    public boolean isEmpty() {
        if (firstPassenger == null)
            return true;
        return false;
    }

    public boolean checkIfOnTime(Person P) {
        if (P.getDayRegistered() <= deadlineDay &&
P.getMonthRegistered() <= deadlineMonth
                && P.getYearRegistered() <=
deadlineYear)
            return true;
        return false;
    }

    public boolean checkIfEnoughSeats() {
        if (currentNumberOfSeats < MaxSeats)
```

```
            return true;
        return false;
    }

    public boolean checkWeight(Person P) {
        if (P.LuggageWeight() < MaxWeight)
            return true;
        return false;
    }

    public boolean CanGetOnPlane(Person P) {
        if (checkIfOnTime(P) && checkIfEnoughSeats() &&
checkWeight(P) && P.inGoodHealth() &&
P.isRightToTravel()
                && P.securityScreening())
            return true;
        return false;
    }

    public boolean checkIfP1lessThanP2(Person P1,
Person P2) {
        if (P1.getYearRegistered() <
P2.getYearRegistered())
            return true;
        else if (P1.getYearRegistered() >
P2.getYearRegistered())
            return false;
        else {
            if (P1.getMonthRegistered() <
P2.getMonthRegistered())
                return true;
            else if (P1.getMonthRegistered() >
P2.getMonthRegistered())
                return false;
            else {
                if (P1.getDayRegistered() <
P2.getDayRegistered())
                    return true;
                else if (P1.getDayRegistered() >
P2.getDayRegistered())
```

```
                    return false;
                else
                    return true;
            }
        }
    }

    public boolean addToAirplane(Person P) {
        if (CanGetOnPlane(P)) {
            if (firstPassenger == null) {
                firstPassenger = P;
                currentNumberOfSeats++;
                return true;
            }
            Person temp = firstPassenger;
            while (true) {
                if (checkIfP1lessThanP2(P, temp)) {
                    if (temp.getRight() == null) {
                        temp.setRight(P);
                        currentNumberOfSeats++;

    System.out.println(""+P.getName()+" got on the
plane");
                        return true;
                    } else {
                        temp = temp.getRight();
                    }

                } else {
                    if (temp.getLeft() == null) {
                        temp.setLeft(P);
                        currentNumberOfSeats++;

    System.out.println(""+P.getName()+" got on the
plane");
                        return true;
                    }
                    temp = temp.getLeft();
                }
            }
        } else {
            System.out.println(P.getName()+" can't get
on the airplane because:");

    if(this.currentNumberOfSeats==this.MaxSeats)
                System.out.println("This section is
full");
                if(!this.checkIfOnTime(P))
                    System.out.println("This person
was late to register");
                if(!P.isRightToTravel())
                    System.out.println("This person
doesn't have the right to travel");
                if(!P.securityScreening())
                    System.out.println("This person
got caught in the security line with a banned thing or
he/she didn't get checked in the security line");
                if(!P.inGoodHealth())
                    System.out.println("The person's
health doesn't permit to get on the plane");
            return false;
        }
    }


    public int height(Person P) {
        if (P == null)
            return 0;
        else
            return 1 + Math.max(height(P.getLeft()),
height(P.getRight()));
    }

    public int getheight() {
        return height(firstPassenger);
    }

    public void displayAsBST() {
        displayAsBST(firstPassenger, 0);
    }

    public void displayAsBST(Person P, int level) {
        if (P != null) {
            displayAsBST(P.getRight(), level + 1);
            for (int i = 0; i < level; i++)
                System.out.print("\t");
            System.out.println("Name: " + P.getName()
+ "-- Age: " + P.getAge());
            displayAsBST(P.getLeft(), level + 1);
        }
    }

    public int countPassengers() {
        return countPassengers(firstPassenger);
    }

    public int countPassengers(Person P) {
        if (P == null)
            return 0;
        else
            return 1 + countPassengers(P.getLeft()) +
countPassengers(P.getRight());
    }

    public void displayOldPassengers() {
        displayOldPassengers(firstPassenger, 0);
    }

    public void displayOldPassengers(Person P, int
level) {
        if (P != null) {
            displayOldPassengers(P.getRight(), level +
1);
            if (P.getAge() > 70) {
                P.display();
                System.out.println();
            }
            displayOldPassengers(P.getLeft(), level +
1);
```

```
        }
    }


    public void displayPassengers() {
        displayPassengers(firstPassenger, 0);
    }

    public void displayPassengers(Person P, int level)
{
        if (P != null) {
            displayPassengers(P.getRight(), level +
1);
            P.display();
            System.out.println();
            displayPassengers(P.getLeft(), level + 1);
        }
    }

    public boolean search(Person P) {
        if (firstPassenger == null)
            return false;
        else {
            Person temp = firstPassenger;
            while (true) {
                if (P.equals(temp))
                    return true;
                else if (checkIfP1lessThanP2(P, temp))
{
                    if (temp.getRight() == null) {

                        System.out.println("The
passenger is not found on the airplane");
                        return false;
                    } else
                        temp = temp.getRight();
                } else {
                    if (temp.getLeft() == null) {
                        System.out.println("The
person is not found on the airplane");
                        return false;
                    } else
                        temp = temp.getLeft();
                }
            }
        }
    }

    public void removePerson(Person P) {
        Person temp = firstPassenger;
        Person parent = firstPassenger;
        boolean flag = false;
        while (!temp.equals(P)) {
            parent = temp;
            if (checkIfP1lessThanP2(P, parent)) {
                temp = temp.getRight();
                flag = true;
            } else {
                temp = temp.getLeft();
                flag = false;
            }
            if (temp == null) {
                return;
            }
        }
        if (temp.getLeft() == null && temp.getRight()
== null) {
            if (temp.equals(firstPassenger))
                firstPassenger = null;
            else if (flag) {
                parent.setRight(null);
            } else {
                parent.setLeft(null);
            }
            currentNumberOfSeats--;
        } else if (!temp.getLeft().equals(null) &&
temp.getRight().equals(null)) {
            if (temp.equals(firstPassenger))
```

```
                firstPassenger = temp.getLeft();
            if (flag == false)
                parent.setLeft(temp.getLeft());
            else
                parent.setRight(temp.getLeft());
            currentNumberOfSeats--;
        } else if (temp.getLeft().equals(null) &&
!temp.getRight().equals(null)) {
            if (temp.equals(firstPassenger))
                firstPassenger = temp.getRight();
            if (flag == false)
                parent.setLeft(temp.getRight());
            else
                parent.setRight(temp.getRight());
            currentNumberOfSeats--;
        } else if (!temp.getLeft().equals(null) &&
!temp.getRight().equals(null)) {
            Person minParent = temp;
            Person minNode = temp.getRight();
            while (minNode.getLeft() != null) {
                minParent = minNode;
                minNode = minNode.getLeft();
            }
            if (minParent != temp) {
                minParent.setLeft(minNode.getRight());
            } else {
                temp.setRight(minNode.getRight());
            }
            minNode.setLeft(temp.getLeft());
            minNode.setRight(temp.getRight());
            if (parent == null)
                firstPassenger = minNode;
            else {
                if (flag)
                    parent.setRight(minNode);
                else
                    parent.setLeft(minNode);
            }
            currentNumberOfSeats--;
        }
```

```
    }

    public Person minValue(Person currentNode) {
        while (currentNode.getLeft() != null) {
            currentNode = currentNode.getLeft();
        }
        return currentNode;
    }
    public Person maxValue(Person currentNode) {
        while (currentNode.getRight() != null) {
            currentNode = currentNode.getRight();
        }
        return currentNode;
    }
}
```

Main Class:

```
package Project;

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Airplane BusinessClass = new Airplane();
        Airplane EconomyPlusClass = new Airplane();
        Airplane MilitaryClass = new Airplane();
        Airplane EconomyClass = new Airplane();
        Airplane ClassForDisabledPeople = new
Airplane();
        SecurityLine S=new SecurityLine();

        BusinessClass.setMaxWeight(20);
        BusinessClass.setDeadlineDay(31);
        BusinessClass.setDeadlineMonth(12);
        BusinessClass.setDeadlineYear(2023);
        BusinessClass.setMaxSeats(20);

        EconomyPlusClass.setMaxWeight(20);
        EconomyPlusClass.setDeadlineDay(31);
        EconomyPlusClass.setDeadlineMonth(12);
        EconomyPlusClass.setDeadlineYear(2023);
        EconomyPlusClass.setMaxSeats(36);

        MilitaryClass.setMaxWeight(20);
        MilitaryClass.setDeadlineDay(31);
        MilitaryClass.setDeadlineMonth(12);
        MilitaryClass.setDeadlineYear(2023);
        MilitaryClass.setMaxSeats(12);

        EconomyClass.setMaxWeight(20);
        EconomyClass.setDeadlineDay(31);
        EconomyClass.setDeadlineMonth(12);
        EconomyClass.setDeadlineYear(2023);
        EconomyClass.setMaxSeats(102);

        ClassForDisabledPeople.setMaxWeight(20);
        ClassForDisabledPeople.setDeadlineDay(31);
        ClassForDisabledPeople.setDeadlineMonth(12);
        ClassForDisabledPeople.setDeadlineYear(2023);
        ClassForDisabledPeople.setMaxSeats(12);

        Person[] ListOfPersons=new Person[110];

        ListOfPersons[0] = new Person("John Doe");
        ListOfPersons[1] = new Person("Jacob Adams");
        ListOfPersons[2] = new Person("Jane Smith");
        ListOfPersons[3] = new Person("Alex Johnson");
        ListOfPersons[4] = new Person("Emily Brown");
        ListOfPersons[5] = new Person("Michael Davis");
        ListOfPersons[6] = new Person("Olivia White");
        ListOfPersons[7] = new Person("Daniel Miller");
        ListOfPersons[8] = new Person("Grace Wilson");
        ListOfPersons[9] = new Person("Matthew
Taylor");
        ListOfPersons[10] = new Person("Ava Martinez");
        ListOfPersons[11] = new Person("William
Harris");
        ListOfPersons[12] = new Person("Abigail
Martin");
```

```
        ListOfPersons[13] = new Person("James
Anderson");
        ListOfPersons[14] = new Person("Avery Davis");
        ListOfPersons[15] = new Person("Emma Garcia");
        ListOfPersons[16] = new Person("Ethan Turner");
        ListOfPersons[17] = new Person("Natalie
Mitchell");
        ListOfPersons[18] = new Person("Oliver
Cooper");
        ListOfPersons[19] = new Person("Grace King");
        ListOfPersons[20] = new Person("Jackson
Moore");
        ListOfPersons[21] = new Person("Lucas Turner");
        ListOfPersons[22] = new Person("Samantha
Mitchell");
        ListOfPersons[23] = new Person("Aiden Cooper");
        ListOfPersons[24] = new Person("Hannah King");
        ListOfPersons[25] = new Person("Caleb Moore");
        ListOfPersons[26] = new Person("Daniel
Turner");
        ListOfPersons[27] = new Person("Hailey
Mitchell");
        ListOfPersons[28] = new Person("Leo Cooper");
        ListOfPersons[29] = new Person("Mia King");
        ListOfPersons[30] = new Person("Evan Moore");
        ListOfPersons[31] = new Person("Aria Turner");
        ListOfPersons[32] = new Person("Logan
Mitchell");
        ListOfPersons[33] = new Person("Lila Cooper");
        ListOfPersons[34] = new Person("Oscar King");
        ListOfPersons[35] = new Person("Ava Moore");
        ListOfPersons[36] = new Person("Nathan
Turner");
        ListOfPersons[37] = new Person("Grace
Mitchell");
        ListOfPersons[38] = new Person("Carter
Cooper");
        ListOfPersons[39] = new Person("Sophie King");
        ListOfPersons[40] = new Person("Liam Moore");
        ListOfPersons[41] = new Person("Liam Turner");
        ListOfPersons[42] = new Person("Ella
Mitchell");
        ListOfPersons[43] = new Person("Henry Cooper");
        ListOfPersons[44] = new Person("Ava King");
        ListOfPersons[45] = new Person("Mia Moore");
        ListOfPersons[46] = new Person("Logan Turner");
        ListOfPersons[47] = new Person("Sophia
Mitchell");
        ListOfPersons[48] = new Person("Ethan Cooper");
        ListOfPersons[49] = new Person("Emma King");
        ListOfPersons[50] = new Person("Oscar Moore");
        ListOfPersons[51] = new Person("Aiden Turner");
        ListOfPersons[52] = new Person("Zoe Mitchell");
        ListOfPersons[53] = new Person("Grace Cooper");
        ListOfPersons[54] = new Person("Noah King");
        ListOfPersons[55] = new Person("Chloe Moore");
        ListOfPersons[56] = new Person("Olivia
Turner");
        ListOfPersons[57] = new Person("Liam
Mitchell");
        ListOfPersons[58] = new Person("Aria Cooper");
        ListOfPersons[59] = new Person("Carter King");
        ListOfPersons[60] = new Person("Lily Moore");
        ListOfPersons[61] = new Person("Ella Turner");
        ListOfPersons[62] = new Person("Lucas
Mitchell");
        ListOfPersons[63] = new Person("Aiden Cooper");
        ListOfPersons[64] = new Person("Ava King");
        ListOfPersons[65] = new Person("Ethan Moore");
        ListOfPersons[66] = new Person("Sophie
Turner");
        ListOfPersons[67] = new Person("Daniel
Mitchell");
        ListOfPersons[68] = new Person("Grace Cooper");
        ListOfPersons[69] = new Person("Matthew King");
        ListOfPersons[70] = new Person("Ava Moore");
        ListOfPersons[71] = new Person("Olivia
Turner");
        ListOfPersons[72] = new Person("Daniel
Mitchell");
```

```
        ListOfPersons[73] = new Person("Grace Cooper");
        ListOfPersons[74] = new Person("Matthew King");
        ListOfPersons[75] = new Person("Ava Moore");
        ListOfPersons[76] = new Person("Liam Turner");
        ListOfPersons[77] = new Person("Ella
Mitchell");
        ListOfPersons[78] = new Person("Henry Cooper");
        ListOfPersons[79] = new Person("Ava King");
        ListOfPersons[80] = new Person("Mia Moore");
        ListOfPersons[81] = new Person("Logan Turner");
        ListOfPersons[82] = new Person("Sophia
Mitchell");
        ListOfPersons[83] = new Person("Ethan Cooper");
        ListOfPersons[84] = new Person("Emma King");
        ListOfPersons[85] = new Person("Oscar Moore");
        ListOfPersons[86] = new Person("Aiden Turner");
        ListOfPersons[87] = new Person("Zoe Mitchell");
        ListOfPersons[88] = new Person("Grace Cooper");
        ListOfPersons[89] = new Person("Noah King");
        ListOfPersons[90] = new Person("Chloe Moore");
        ListOfPersons[91] = new Person("Olivia
Turner");
        ListOfPersons[92] = new Person("Liam
Mitchell");
        ListOfPersons[93] = new Person("Aria Cooper");
        ListOfPersons[94] = new Person("Carter King");
        ListOfPersons[95] = new Person("Emma Turner");
        ListOfPersons[96] = new Person("Aiden
Mitchell");
        ListOfPersons[97] = new Person("Olivia
Cooper");
        ListOfPersons[98] = new Person("Lucas King");
        ListOfPersons[99] = new Person("Chloe Moore");
        ListOfPersons[100] = new Person("Mason
Turner");
        ListOfPersons[101] = new Person("Zoe
Mitchell");
        ListOfPersons[102] = new Person("Elijah
Cooper");
        ListOfPersons[103] = new Person("Lily King");
        ListOfPersons[104] = new Person("Owen Moore");
        ListOfPersons[105] = new Person("Abigail
Turner");
        ListOfPersons[106] = new Person("Jackson
Mitchell");
        ListOfPersons[107] = new Person("Sofia
Cooper");
        ListOfPersons[108] = new Person("Noah King");
        ListOfPersons[109] = new Person("Emily Moore");

        ListOfPersons[0].setAge(25);
        ListOfPersons[0].setDayRegistered(5);
        ListOfPersons[0].setMonthRegistered(3);
        ListOfPersons[0].setYearRegistered(2023);
        ListOfPersons[0].setPersonClass(1);
        ListOfPersons[0].setRightToTravel(true);
        ListOfPersons[0].setLuggageWeight(15.5);
        ListOfPersons[0].setCheckIfDisabled(false);
        ListOfPersons[0].setMinimumPay(500);
        ListOfPersons[0].setExtraPay(true);
        ListOfPersons[0].setMilitary(false);
        ListOfPersons[0].setInGoodHealth(true);

        ListOfPersons[1].setAge(30);
        ListOfPersons[1].setDayRegistered(10);
        ListOfPersons[1].setMonthRegistered(5);
        ListOfPersons[1].setYearRegistered(2023);
        ListOfPersons[1].setPersonClass(1);
        ListOfPersons[1].setRightToTravel(true);
        ListOfPersons[1].setLuggageWeight(14.0);
        ListOfPersons[1].setCheckIfDisabled(false);
        ListOfPersons[1].setMinimumPay(550);
        ListOfPersons[1].setExtraPay(false);
        ListOfPersons[1].setMilitary(false);
        ListOfPersons[1].setInGoodHealth(true);

        ListOfPersons[2].setAge(35);
        ListOfPersons[2].setDayRegistered(15);
        ListOfPersons[2].setMonthRegistered(4);
        ListOfPersons[2].setYearRegistered(2023);
```

```
ListOfPersons[2].setPersonClass(1);
ListOfPersons[2].setRightToTravel(true);
ListOfPersons[2].setLuggageWeight(13.0);
ListOfPersons[2].setCheckIfDisabled(false);
ListOfPersons[2].setMinimumPay(600);
ListOfPersons[2].setExtraPay(false);
ListOfPersons[2].setMilitary(false);
ListOfPersons[2].setInGoodHealth(true);

ListOfPersons[3].setAge(45);
ListOfPersons[3].setDayRegistered(7);
ListOfPersons[3].setMonthRegistered(8);
ListOfPersons[3].setYearRegistered(2023);
ListOfPersons[3].setPersonClass(2);
ListOfPersons[3].setRightToTravel(true);
ListOfPersons[3].setLuggageWeight(16.0);
ListOfPersons[3].setCheckIfDisabled(false);
ListOfPersons[3].setMinimumPay(700);
ListOfPersons[3].setExtraPay(false);
ListOfPersons[3].setMilitary(false);
ListOfPersons[3].setInGoodHealth(true);

ListOfPersons[4].setAge(28);
ListOfPersons[4].setDayRegistered(18);
ListOfPersons[4].setMonthRegistered(2);
ListOfPersons[4].setYearRegistered(2023);
ListOfPersons[4].setPersonClass(1);
ListOfPersons[4].setRightToTravel(true);
ListOfPersons[4].setLuggageWeight(14.5);
ListOfPersons[4].setCheckIfDisabled(false);
ListOfPersons[4].setMinimumPay(520);
ListOfPersons[4].setExtraPay(false);
ListOfPersons[4].setMilitary(false);
ListOfPersons[4].setInGoodHealth(true);

ListOfPersons[5].setAge(32);
ListOfPersons[5].setDayRegistered(22);
ListOfPersons[5].setMonthRegistered(7);
ListOfPersons[5].setYearRegistered(2023);
ListOfPersons[5].setPersonClass(1);
```

```
ListOfPersons[5].setRightToTravel(true);
ListOfPersons[5].setLuggageWeight(15.0);
ListOfPersons[5].setCheckIfDisabled(false);
ListOfPersons[5].setMinimumPay(580);
ListOfPersons[5].setExtraPay(true);
ListOfPersons[5].setMilitary(false);
ListOfPersons[5].setInGoodHealth(true);

ListOfPersons[6].setAge(40);
ListOfPersons[6].setDayRegistered(12);
ListOfPersons[6].setMonthRegistered(9);
ListOfPersons[6].setYearRegistered(2023);
ListOfPersons[6].setPersonClass(2);
ListOfPersons[6].setRightToTravel(true);
ListOfPersons[6].setLuggageWeight(16.5);
ListOfPersons[6].setCheckIfDisabled(false);
ListOfPersons[6].setMinimumPay(620);
ListOfPersons[6].setExtraPay(false);
ListOfPersons[6].setMilitary(false);
ListOfPersons[6].setInGoodHealth(true);

ListOfPersons[7].setAge(48);
ListOfPersons[7].setDayRegistered(5);
ListOfPersons[7].setMonthRegistered(11);
ListOfPersons[7].setYearRegistered(2023);
ListOfPersons[7].setPersonClass(2);
ListOfPersons[7].setRightToTravel(true);
ListOfPersons[7].setLuggageWeight(18.0);
ListOfPersons[7].setCheckIfDisabled(false);
ListOfPersons[7].setMinimumPay(700);
ListOfPersons[7].setExtraPay(false);
ListOfPersons[7].setMilitary(false);
ListOfPersons[7].setInGoodHealth(true);

ListOfPersons[8].setAge(33);
ListOfPersons[8].setDayRegistered(28);
ListOfPersons[8].setMonthRegistered(3);
ListOfPersons[8].setYearRegistered(2024);
ListOfPersons[8].setPersonClass(3);
ListOfPersons[8].setRightToTravel(true);
```

```
ListOfPersons[8].setLuggageWeight(14.0);
ListOfPersons[8].setCheckIfDisabled(false);
ListOfPersons[8].setMinimumPay(550);
ListOfPersons[8].setExtraPay(false);
ListOfPersons[8].setMilitary(false);
ListOfPersons[8].setInGoodHealth(true);

ListOfPersons[9].setAge(42);
ListOfPersons[9].setDayRegistered(19);
ListOfPersons[9].setMonthRegistered(5);
ListOfPersons[9].setYearRegistered(2023);
ListOfPersons[9].setPersonClass(1);
ListOfPersons[9].setRightToTravel(true);
ListOfPersons[9].setLuggageWeight(15.5);
ListOfPersons[9].setCheckIfDisabled(false);
ListOfPersons[9].setMinimumPay(590);
ListOfPersons[9].setExtraPay(false);
ListOfPersons[9].setMilitary(false);
ListOfPersons[9].setInGoodHealth(false);

ListOfPersons[10].setAge(37);
ListOfPersons[10].setDayRegistered(14);
ListOfPersons[10].setMonthRegistered(8);
ListOfPersons[10].setYearRegistered(2024);
ListOfPersons[10].setPersonClass(2);
ListOfPersons[10].setRightToTravel(true);
ListOfPersons[10].setLuggageWeight(16.0);
ListOfPersons[10].setCheckIfDisabled(false);
ListOfPersons[10].setMinimumPay(630);
ListOfPersons[10].setExtraPay(false);
ListOfPersons[10].setMilitary(false);
ListOfPersons[10].setInGoodHealth(true);

ListOfPersons[11].setAge(45);
ListOfPersons[11].setDayRegistered(8);
ListOfPersons[11].setMonthRegistered(12);
ListOfPersons[11].setYearRegistered(2023);
ListOfPersons[11].setPersonClass(2);
ListOfPersons[11].setRightToTravel(true);
ListOfPersons[11].setLuggageWeight(17.5);
```

```
ListOfPersons[11].setCheckIfDisabled(false);
ListOfPersons[11].setMinimumPay(680);
ListOfPersons[11].setExtraPay(false);
ListOfPersons[11].setMilitary(false);
ListOfPersons[11].setInGoodHealth(false);

ListOfPersons[12].setAge(39);
ListOfPersons[12].setDayRegistered(25);
ListOfPersons[12].setMonthRegistered(4);
ListOfPersons[12].setYearRegistered(2023);
ListOfPersons[12].setPersonClass(3);
ListOfPersons[12].setRightToTravel(true);
ListOfPersons[12].setLuggageWeight(15.0);
ListOfPersons[12].setCheckIfDisabled(false);
ListOfPersons[12].setMinimumPay(560);
ListOfPersons[12].setExtraPay(false);
ListOfPersons[12].setMilitary(false);
ListOfPersons[12].setInGoodHealth(false);

ListOfPersons[13].setAge(41);
ListOfPersons[13].setDayRegistered(17);
ListOfPersons[13].setMonthRegistered(6);
ListOfPersons[13].setYearRegistered(2023);
ListOfPersons[13].setPersonClass(1);
ListOfPersons[13].setRightToTravel(true);
ListOfPersons[13].setLuggageWeight(14.5);
ListOfPersons[13].setCheckIfDisabled(false);
ListOfPersons[13].setMinimumPay(600);
ListOfPersons[13].setExtraPay(false);
ListOfPersons[13].setMilitary(false);
ListOfPersons[13].setInGoodHealth(true);

ListOfPersons[14].setAge(38);
ListOfPersons[14].setDayRegistered(10);
ListOfPersons[14].setMonthRegistered(9);
ListOfPersons[14].setYearRegistered(2023);
ListOfPersons[14].setPersonClass(2);
ListOfPersons[14].setRightToTravel(true);
ListOfPersons[14].setLuggageWeight(16.5);
ListOfPersons[14].setCheckIfDisabled(false);
```

```
ListOfPersons[14].setMinimumPay(640);
ListOfPersons[14].setExtraPay(false);
ListOfPersons[14].setMilitary(false);
ListOfPersons[14].setInGoodHealth(true);

ListOfPersons[15].setAge(44);
ListOfPersons[15].setDayRegistered(7);
ListOfPersons[15].setMonthRegistered(1);
ListOfPersons[15].setYearRegistered(2023);
ListOfPersons[15].setPersonClass(3);
ListOfPersons[15].setRightToTravel(true);
ListOfPersons[15].setLuggageWeight(18.0);
ListOfPersons[15].setCheckIfDisabled(false);
ListOfPersons[15].setMinimumPay(690);
ListOfPersons[15].setExtraPay(false);
ListOfPersons[15].setMilitary(false);
ListOfPersons[15].setInGoodHealth(true);

ListOfPersons[16].setAge(40);
ListOfPersons[16].setDayRegistered(22);
ListOfPersons[16].setMonthRegistered(2);
ListOfPersons[16].setYearRegistered(2023);
ListOfPersons[16].setPersonClass(1);
ListOfPersons[16].setRightToTravel(true);
ListOfPersons[16].setLuggageWeight(15.0);
ListOfPersons[16].setCheckIfDisabled(false);
ListOfPersons[16].setMinimumPay(600);
ListOfPersons[16].setExtraPay(false);
ListOfPersons[16].setMilitary(false);
ListOfPersons[16].setInGoodHealth(true);

ListOfPersons[17].setAge(36);
ListOfPersons[17].setDayRegistered(19);
ListOfPersons[17].setMonthRegistered(5);
ListOfPersons[17].setYearRegistered(2023);
ListOfPersons[17].setPersonClass(2);
ListOfPersons[17].setRightToTravel(true);
ListOfPersons[17].setLuggageWeight(17.5);
ListOfPersons[17].setCheckIfDisabled(false);
ListOfPersons[17].setMinimumPay(680);
```

```
ListOfPersons[17].setExtraPay(false);
ListOfPersons[17].setMilitary(false);
ListOfPersons[17].setInGoodHealth(true);

ListOfPersons[18].setAge(42);
ListOfPersons[18].setDayRegistered(12);
ListOfPersons[18].setMonthRegistered(8);
ListOfPersons[18].setYearRegistered(2023);
ListOfPersons[18].setPersonClass(3);
ListOfPersons[18].setRightToTravel(true);
ListOfPersons[18].setLuggageWeight(16.0);
ListOfPersons[18].setCheckIfDisabled(false);
ListOfPersons[18].setMinimumPay(720);
ListOfPersons[18].setExtraPay(false);
ListOfPersons[18].setMilitary(false);
ListOfPersons[18].setInGoodHealth(true);

ListOfPersons[19].setAge(35);
ListOfPersons[19].setDayRegistered(25);
ListOfPersons[19].setMonthRegistered(3);
ListOfPersons[19].setYearRegistered(2023);
ListOfPersons[19].setPersonClass(1);
ListOfPersons[19].setRightToTravel(true);
ListOfPersons[19].setLuggageWeight(14.0);
ListOfPersons[19].setCheckIfDisabled(true);
ListOfPersons[19].setMinimumPay(620);
ListOfPersons[19].setExtraPay(true);
ListOfPersons[19].setMilitary(false);
ListOfPersons[19].setInGoodHealth(true);

ListOfPersons[20].setAge(50);
ListOfPersons[20].setDayRegistered(8);
ListOfPersons[20].setMonthRegistered(7);
ListOfPersons[20].setYearRegistered(2023);
ListOfPersons[20].setPersonClass(2);
ListOfPersons[20].setRightToTravel(true);
ListOfPersons[20].setLuggageWeight(19.5);
ListOfPersons[20].setCheckIfDisabled(true);
ListOfPersons[20].setMinimumPay(700);
ListOfPersons[20].setExtraPay(true);
```

```
   ListOfPersons[20].setMilitary(false);
   ListOfPersons[20].setInGoodHealth(true);

   ListOfPersons[21].setAge(38);
   ListOfPersons[21].setDayRegistered(17);
   ListOfPersons[21].setMonthRegistered(9);
   ListOfPersons[21].setYearRegistered(2023);
   ListOfPersons[21].setPersonClass(3);
   ListOfPersons[21].setRightToTravel(true);
   ListOfPersons[21].setLuggageWeight(15.5);
   ListOfPersons[21].setCheckIfDisabled(true);
   ListOfPersons[21].setMinimumPay(740);
   ListOfPersons[21].setExtraPay(true);
   ListOfPersons[21].setMilitary(false);
   ListOfPersons[21].setInGoodHealth(true);

   ListOfPersons[22].setAge(45);
   ListOfPersons[22].setDayRegistered(29);
   ListOfPersons[22].setMonthRegistered(1);
   ListOfPersons[22].setYearRegistered(2023);
   ListOfPersons[22].setPersonClass(1);
   ListOfPersons[22].setRightToTravel(true);
   ListOfPersons[22].setLuggageWeight(13.0);
   ListOfPersons[22].setCheckIfDisabled(false);
   ListOfPersons[22].setMinimumPay(660);
   ListOfPersons[22].setExtraPay(true);
   ListOfPersons[22].setMilitary(false);
   ListOfPersons[22].setInGoodHealth(true);

// Person 24
   ListOfPersons[23].setAge(32);
   ListOfPersons[23].setDayRegistered(14);
   ListOfPersons[23].setMonthRegistered(4);
   ListOfPersons[23].setYearRegistered(2023);
   ListOfPersons[23].setPersonClass(2);
   ListOfPersons[23].setRightToTravel(true);
   ListOfPersons[23].setLuggageWeight(18.0);
   ListOfPersons[23].setCheckIfDisabled(true);
   ListOfPersons[23].setMinimumPay(720);
   ListOfPersons[23].setExtraPay(true);
```

```
ListOfPersons[23].setMilitary(false);
ListOfPersons[23].setInGoodHealth(true);

ListOfPersons[24].setAge(34);
ListOfPersons[24].setDayRegistered(21);
ListOfPersons[24].setMonthRegistered(6);
ListOfPersons[24].setYearRegistered(2023);
ListOfPersons[24].setPersonClass(3);
ListOfPersons[24].setRightToTravel(true);
ListOfPersons[24].setLuggageWeight(16.5);
ListOfPersons[24].setCheckIfDisabled(false);
ListOfPersons[24].setMinimumPay(780);
ListOfPersons[24].setExtraPay(true);
ListOfPersons[24].setMilitary(false);
ListOfPersons[24].setInGoodHealth(true);

ListOfPersons[25].setAge(55);
ListOfPersons[25].setDayRegistered(5);
ListOfPersons[25].setMonthRegistered(11);
ListOfPersons[25].setYearRegistered(2022);
ListOfPersons[25].setPersonClass(1);
ListOfPersons[25].setRightToTravel(true);
ListOfPersons[25].setLuggageWeight(17.0);
ListOfPersons[25].setCheckIfDisabled(true);
ListOfPersons[25].setMinimumPay(650);
ListOfPersons[25].setExtraPay(true);
ListOfPersons[25].setMilitary(false);
ListOfPersons[25].setInGoodHealth(true);

ListOfPersons[26].setAge(47);
ListOfPersons[26].setDayRegistered(12);
ListOfPersons[26].setMonthRegistered(3);
ListOfPersons[26].setYearRegistered(2023);
ListOfPersons[26].setPersonClass(2);
ListOfPersons[26].setRightToTravel(true);
ListOfPersons[26].setLuggageWeight(19.0);
ListOfPersons[26].setCheckIfDisabled(false);
ListOfPersons[26].setMinimumPay(710);
ListOfPersons[26].setExtraPay(true);
ListOfPersons[26].setMilitary(false);
```

```
ListOfPersons[26].setInGoodHealth(true);

ListOfPersons[27].setAge(31);
ListOfPersons[27].setDayRegistered(18);
ListOfPersons[27].setMonthRegistered(7);
ListOfPersons[27].setYearRegistered(2023);
ListOfPersons[27].setPersonClass(3);
ListOfPersons[27].setRightToTravel(true);
ListOfPersons[27].setLuggageWeight(16.5);
ListOfPersons[27].setCheckIfDisabled(true);
ListOfPersons[27].setMinimumPay(780);
ListOfPersons[27].setExtraPay(true);
ListOfPersons[27].setMilitary(false);
ListOfPersons[27].setInGoodHealth(true);

ListOfPersons[28].setAge(58);
ListOfPersons[28].setDayRegistered(9);
ListOfPersons[28].setMonthRegistered(2);
ListOfPersons[28].setYearRegistered(2023);
ListOfPersons[28].setPersonClass(1);
ListOfPersons[28].setRightToTravel(true);
ListOfPersons[28].setLuggageWeight(15.0);
ListOfPersons[28].setCheckIfDisabled(false);
ListOfPersons[28].setMinimumPay(680);
ListOfPersons[28].setExtraPay(true);
ListOfPersons[28].setMilitary(false);
ListOfPersons[28].setInGoodHealth(true);

ListOfPersons[29].setAge(40);
ListOfPersons[29].setDayRegistered(15);
ListOfPersons[29].setMonthRegistered(6);
ListOfPersons[29].setYearRegistered(2023);
ListOfPersons[29].setPersonClass(2);
ListOfPersons[29].setRightToTravel(true);
ListOfPersons[29].setLuggageWeight(18.0);
ListOfPersons[29].setCheckIfDisabled(true);
ListOfPersons[29].setMinimumPay(720);
ListOfPersons[29].setExtraPay(true);
ListOfPersons[29].setMilitary(false);
ListOfPersons[29].setInGoodHealth(true);
```

```
// Person 31
   ListOfPersons[30].setAge(25);
   ListOfPersons[30].setDayRegistered(20);
   ListOfPersons[30].setMonthRegistered(5);
   ListOfPersons[30].setYearRegistered(2023);
   ListOfPersons[30].setPersonClass(4);
   ListOfPersons[30].setRightToTravel(true);
   ListOfPersons[30].setLuggageWeight(16.0);
   ListOfPersons[30].setCheckIfDisabled(true);
   ListOfPersons[30].setMinimumPay(690);
   ListOfPersons[30].setExtraPay(false);
   ListOfPersons[30].setMilitary(false);
   ListOfPersons[30].setInGoodHealth(true);

   // Person 32
   ListOfPersons[31].setAge(62);
   ListOfPersons[31].setDayRegistered(14);
   ListOfPersons[31].setMonthRegistered(8);
   ListOfPersons[31].setYearRegistered(2023);
   ListOfPersons[31].setPersonClass(1);
   ListOfPersons[31].setRightToTravel(true);
   ListOfPersons[31].setLuggageWeight(19.5);
   ListOfPersons[31].setCheckIfDisabled(false);
   ListOfPersons[31].setMinimumPay(710);
   ListOfPersons[31].setExtraPay(false);
   ListOfPersons[31].setMilitary(true);
   ListOfPersons[31].setInGoodHealth(true);

   // Person 33
   ListOfPersons[32].setAge(33);
   ListOfPersons[32].setDayRegistered(11);
   ListOfPersons[32].setMonthRegistered(1);
   ListOfPersons[32].setYearRegistered(2023);
   ListOfPersons[32].setPersonClass(3);
   ListOfPersons[32].setRightToTravel(true);
   ListOfPersons[32].setLuggageWeight(17.0);
   ListOfPersons[32].setCheckIfDisabled(false);
   ListOfPersons[32].setMinimumPay(730);
   ListOfPersons[32].setExtraPay(false);
```

```
ListOfPersons[32].setMilitary(true);
ListOfPersons[32].setInGoodHealth(true);

// Person 34
ListOfPersons[33].setAge(48);
ListOfPersons[33].setDayRegistered(5);
ListOfPersons[33].setMonthRegistered(7);
ListOfPersons[33].setYearRegistered(2023);
ListOfPersons[33].setPersonClass(2);
ListOfPersons[33].setRightToTravel(true);
ListOfPersons[33].setLuggageWeight(16.5);
ListOfPersons[33].setCheckIfDisabled(false);
ListOfPersons[33].setMinimumPay(700);
ListOfPersons[33].setExtraPay(false);
ListOfPersons[33].setMilitary(false);
ListOfPersons[33].setInGoodHealth(true);

// Person 35
ListOfPersons[34].setAge(41);
ListOfPersons[34].setDayRegistered(28);
ListOfPersons[34].setMonthRegistered(9);
ListOfPersons[34].setYearRegistered(2023);
ListOfPersons[34].setPersonClass(4);
ListOfPersons[34].setRightToTravel(true);
ListOfPersons[34].setLuggageWeight(18.0);
ListOfPersons[34].setCheckIfDisabled(false);
ListOfPersons[34].setMinimumPay(720);
ListOfPersons[34].setExtraPay(false);
ListOfPersons[34].setMilitary(false);
ListOfPersons[34].setInGoodHealth(true);

ListOfPersons[35].setAge(29);
ListOfPersons[35].setDayRegistered(12);
ListOfPersons[35].setMonthRegistered(2);
ListOfPersons[35].setYearRegistered(2023);
ListOfPersons[35].setPersonClass(1);
ListOfPersons[35].setRightToTravel(true);
ListOfPersons[35].setLuggageWeight(17.5);
ListOfPersons[35].setCheckIfDisabled(false);
ListOfPersons[35].setMinimumPay(710);
```

```
ListOfPersons[35].setExtraPay(false);
ListOfPersons[35].setMilitary(false);
ListOfPersons[35].setInGoodHealth(true);

// Person 37
ListOfPersons[36].setAge(54);
ListOfPersons[36].setDayRegistered(7);
ListOfPersons[36].setMonthRegistered(4);
ListOfPersons[36].setYearRegistered(2023);
ListOfPersons[36].setPersonClass(2);
ListOfPersons[36].setRightToTravel(true);
ListOfPersons[36].setLuggageWeight(18.5);
ListOfPersons[36].setCheckIfDisabled(false);
ListOfPersons[36].setMinimumPay(720);
ListOfPersons[36].setExtraPay(false);
ListOfPersons[36].setMilitary(false);
ListOfPersons[36].setInGoodHealth(true);

// Person 38
ListOfPersons[37].setAge(38);
ListOfPersons[37].setDayRegistered(25);
ListOfPersons[37].setMonthRegistered(11);
ListOfPersons[37].setYearRegistered(2023);
ListOfPersons[37].setPersonClass(3);
ListOfPersons[37].setRightToTravel(true);
ListOfPersons[37].setLuggageWeight(16.0);
ListOfPersons[37].setCheckIfDisabled(false);
ListOfPersons[37].setMinimumPay(690);
ListOfPersons[37].setExtraPay(false);
ListOfPersons[37].setMilitary(false);
ListOfPersons[37].setInGoodHealth(true);

// Person 39
ListOfPersons[38].setAge(45);
ListOfPersons[38].setDayRegistered(19);
ListOfPersons[38].setMonthRegistered(10);
ListOfPersons[38].setYearRegistered(2023);
ListOfPersons[38].setPersonClass(4);
ListOfPersons[38].setRightToTravel(true);
ListOfPersons[38].setLuggageWeight(15.5);
```

```
        ListOfPersons[38].setCheckIfDisabled(false);
        ListOfPersons[38].setMinimumPay(740);
        ListOfPersons[38].setExtraPay(false);
        ListOfPersons[38].setMilitary(false);
        ListOfPersons[38].setInGoodHealth(true);

        // Person 40
        ListOfPersons[39].setAge(31);
        ListOfPersons[39].setDayRegistered(3);
        ListOfPersons[39].setMonthRegistered(1);
        ListOfPersons[39].setYearRegistered(2023);
        ListOfPersons[39].setPersonClass(1);
        ListOfPersons[39].setRightToTravel(true);
        ListOfPersons[39].setLuggageWeight(19.0);
        ListOfPersons[39].setCheckIfDisabled(false);
        ListOfPersons[39].setMinimumPay(710);
        ListOfPersons[39].setExtraPay(false);
        ListOfPersons[39].setMilitary(false);
        ListOfPersons[39].setInGoodHealth(true);

     // Person 41
        ListOfPersons[40].setAge(33);
        ListOfPersons[40].setDayRegistered(15);
        ListOfPersons[40].setMonthRegistered(7);
        ListOfPersons[40].setYearRegistered(2023);
        ListOfPersons[40].setPersonClass(4);
        ListOfPersons[40].setRightToTravel(true);
        ListOfPersons[40].setLuggageWeight(14.5);
        ListOfPersons[40].setCheckIfDisabled(false);
        ListOfPersons[40].setMinimumPay(720);
        ListOfPersons[40].setExtraPay(false);
        ListOfPersons[40].setMilitary(false);
        ListOfPersons[40].setInGoodHealth(true);

        // Person 42
        ListOfPersons[41].setAge(57);
        ListOfPersons[41].setDayRegistered(21);
        ListOfPersons[41].setMonthRegistered(12);
        ListOfPersons[41].setYearRegistered(2023);
        ListOfPersons[41].setPersonClass(1);
```

```
ListOfPersons[41].setRightToTravel(true);
ListOfPersons[41].setLuggageWeight(17.0);
ListOfPersons[41].setCheckIfDisabled(false);
ListOfPersons[41].setMinimumPay(710);
ListOfPersons[41].setExtraPay(false);
ListOfPersons[41].setMilitary(false);
ListOfPersons[41].setInGoodHealth(true);

// Person 43
ListOfPersons[42].setAge(39);
ListOfPersons[42].setDayRegistered(9);
ListOfPersons[42].setMonthRegistered(5);
ListOfPersons[42].setYearRegistered(2023);
ListOfPersons[42].setPersonClass(2);
ListOfPersons[42].setRightToTravel(true);
ListOfPersons[42].setLuggageWeight(16.5);
ListOfPersons[42].setCheckIfDisabled(false);
ListOfPersons[42].setMinimumPay(730);
ListOfPersons[42].setExtraPay(false);
ListOfPersons[42].setMilitary(false);
ListOfPersons[42].setInGoodHealth(true);

// Person 44
ListOfPersons[43].setAge(48);
ListOfPersons[43].setDayRegistered(4);
ListOfPersons[43].setMonthRegistered(8);
ListOfPersons[43].setYearRegistered(2023);
ListOfPersons[43].setPersonClass(3);
ListOfPersons[43].setRightToTravel(true);
ListOfPersons[43].setLuggageWeight(19.5);
ListOfPersons[43].setCheckIfDisabled(false);
ListOfPersons[43].setMinimumPay(690);
ListOfPersons[43].setExtraPay(false);
ListOfPersons[43].setMilitary(false);
ListOfPersons[43].setInGoodHealth(true);

// Person 45
ListOfPersons[44].setAge(29);
ListOfPersons[44].setDayRegistered(17);
ListOfPersons[44].setMonthRegistered(10);
```

```
   ListOfPersons[44].setYearRegistered(2023);
   ListOfPersons[44].setPersonClass(4);
   ListOfPersons[44].setRightToTravel(true);
   ListOfPersons[44].setLuggageWeight(18.0);
   ListOfPersons[44].setCheckIfDisabled(false);
   ListOfPersons[44].setMinimumPay(740);
   ListOfPersons[44].setExtraPay(false);
   ListOfPersons[44].setMilitary(false);
   ListOfPersons[44].setInGoodHealth(true);

// Person 46
   ListOfPersons[45].setAge(31);
   ListOfPersons[45].setDayRegistered(5);
   ListOfPersons[45].setMonthRegistered(2);
   ListOfPersons[45].setYearRegistered(2023);
   ListOfPersons[45].setPersonClass(1);
   ListOfPersons[45].setRightToTravel(true);
   ListOfPersons[45].setLuggageWeight(15.5);
   ListOfPersons[45].setCheckIfDisabled(false);
   ListOfPersons[45].setMinimumPay(700);
   ListOfPersons[45].setExtraPay(false);
   ListOfPersons[45].setMilitary(false);
   ListOfPersons[45].setInGoodHealth(true);

   // Person 47
   ListOfPersons[46].setAge(42);
   ListOfPersons[46].setDayRegistered(14);
   ListOfPersons[46].setMonthRegistered(9);
   ListOfPersons[46].setYearRegistered(2023);
   ListOfPersons[46].setPersonClass(2);
   ListOfPersons[46].setRightToTravel(true);
   ListOfPersons[46].setLuggageWeight(16.0);
   ListOfPersons[46].setCheckIfDisabled(false);
   ListOfPersons[46].setMinimumPay(710);
   ListOfPersons[46].setExtraPay(false);
   ListOfPersons[46].setMilitary(false);
   ListOfPersons[46].setInGoodHealth(true);

   // Person 48
   ListOfPersons[47].setAge(54);
```

```
ListOfPersons[47].setDayRegistered(11);
ListOfPersons[47].setMonthRegistered(3);
ListOfPersons[47].setYearRegistered(2023);
ListOfPersons[47].setPersonClass(3);
ListOfPersons[47].setRightToTravel(true);
ListOfPersons[47].setLuggageWeight(18.5);
ListOfPersons[47].setCheckIfDisabled(false);
ListOfPersons[47].setMinimumPay(720);
ListOfPersons[47].setExtraPay(false);
ListOfPersons[47].setMilitary(false);
ListOfPersons[47].setInGoodHealth(true);

// Person 49
ListOfPersons[48].setAge(36);
ListOfPersons[48].setDayRegistered(25);
ListOfPersons[48].setMonthRegistered(6);
ListOfPersons[48].setYearRegistered(2023);
ListOfPersons[48].setPersonClass(4);
ListOfPersons[48].setRightToTravel(true);
ListOfPersons[48].setLuggageWeight(19.0);
ListOfPersons[48].setCheckIfDisabled(false);
ListOfPersons[48].setMinimumPay(730);
ListOfPersons[48].setExtraPay(false);
ListOfPersons[48].setMilitary(false);
ListOfPersons[48].setInGoodHealth(true);

// Person 50
ListOfPersons[49].setAge(45);
ListOfPersons[49].setDayRegistered(8);
ListOfPersons[49].setMonthRegistered(11);
ListOfPersons[49].setYearRegistered(2023);
ListOfPersons[49].setPersonClass(1);
ListOfPersons[49].setRightToTravel(true);
ListOfPersons[49].setLuggageWeight(14.0);
ListOfPersons[49].setCheckIfDisabled(false);
ListOfPersons[49].setMinimumPay(690);
ListOfPersons[49].setExtraPay(false);
ListOfPersons[49].setMilitary(false);
ListOfPersons[49].setInGoodHealth(true);
```

```
// Person 51
   ListOfPersons[50].setAge(28);
   ListOfPersons[50].setDayRegistered(17);
   ListOfPersons[50].setMonthRegistered(4);
   ListOfPersons[50].setYearRegistered(2023);
   ListOfPersons[50].setPersonClass(2);
   ListOfPersons[50].setRightToTravel(true);
   ListOfPersons[50].setLuggageWeight(15.8);
   ListOfPersons[50].setCheckIfDisabled(false);
   ListOfPersons[50].setMinimumPay(705);
   ListOfPersons[50].setExtraPay(true);
   ListOfPersons[50].setMilitary(false);
   ListOfPersons[50].setInGoodHealth(true);

   // Person 52
   ListOfPersons[51].setAge(33);
   ListOfPersons[51].setDayRegistered(9);
   ListOfPersons[51].setMonthRegistered(8);
   ListOfPersons[51].setYearRegistered(2023);
   ListOfPersons[51].setPersonClass(3);
   ListOfPersons[51].setRightToTravel(true);
   ListOfPersons[51].setLuggageWeight(17.2);
   ListOfPersons[51].setCheckIfDisabled(false);
   ListOfPersons[51].setMinimumPay(715);
   ListOfPersons[51].setExtraPay(true);
   ListOfPersons[51].setMilitary(false);
   ListOfPersons[51].setInGoodHealth(true);

   // Person 53
   ListOfPersons[52].setAge(40);
   ListOfPersons[52].setDayRegistered(26);
   ListOfPersons[52].setMonthRegistered(5);
   ListOfPersons[52].setYearRegistered(2023);
   ListOfPersons[52].setPersonClass(4);
   ListOfPersons[52].setRightToTravel(false);
   ListOfPersons[52].setLuggageWeight(18.0);
   ListOfPersons[52].setCheckIfDisabled(false);
   ListOfPersons[52].setMinimumPay(725);
   ListOfPersons[52].setExtraPay(true);
   ListOfPersons[52].setMilitary(false);
```

```
        ListOfPersons[52].setInGoodHealth(true);

        // Person 54
        ListOfPersons[53].setAge(29);
        ListOfPersons[53].setDayRegistered(12);
        ListOfPersons[53].setMonthRegistered(10);
        ListOfPersons[53].setYearRegistered(2023);
        ListOfPersons[53].setPersonClass(1);
        ListOfPersons[53].setRightToTravel(true);
        ListOfPersons[53].setLuggageWeight(14.5);
        ListOfPersons[53].setCheckIfDisabled(false);
        ListOfPersons[53].setMinimumPay(695);
        ListOfPersons[53].setExtraPay(true);
        ListOfPersons[53].setMilitary(false);
        ListOfPersons[53].setInGoodHealth(true);

        // Person 55
        ListOfPersons[54].setAge(48);
        ListOfPersons[54].setDayRegistered(19);
        ListOfPersons[54].setMonthRegistered(1);
        ListOfPersons[54].setYearRegistered(2023);
        ListOfPersons[54].setPersonClass(2);
        ListOfPersons[54].setRightToTravel(false);
        ListOfPersons[54].setLuggageWeight(16.5);
        ListOfPersons[54].setCheckIfDisabled(false);
        ListOfPersons[54].setMinimumPay(710);
        ListOfPersons[54].setExtraPay(true);
        ListOfPersons[54].setMilitary(false);
        ListOfPersons[54].setInGoodHealth(true);

     // Person 56
        ListOfPersons[55].setAge(36);
        ListOfPersons[55].setDayRegistered(3);
        ListOfPersons[55].setMonthRegistered(7);
        ListOfPersons[55].setYearRegistered(2023);
        ListOfPersons[55].setPersonClass(3);
        ListOfPersons[55].setRightToTravel(true);
        ListOfPersons[55].setLuggageWeight(18.5);
        ListOfPersons[55].setCheckIfDisabled(false);
        ListOfPersons[55].setMinimumPay(720);
```

```
ListOfPersons[55].setExtraPay(true);
ListOfPersons[55].setMilitary(false);
ListOfPersons[55].setInGoodHealth(true);

// Person 57
ListOfPersons[56].setAge(45);
ListOfPersons[56].setDayRegistered(14);
ListOfPersons[56].setMonthRegistered(2);
ListOfPersons[56].setYearRegistered(2023);
ListOfPersons[56].setPersonClass(4);
ListOfPersons[56].setRightToTravel(true);
ListOfPersons[56].setLuggageWeight(19.2);
ListOfPersons[56].setCheckIfDisabled(false);
ListOfPersons[56].setMinimumPay(730);
ListOfPersons[56].setExtraPay(true);
ListOfPersons[56].setMilitary(false);
ListOfPersons[56].setInGoodHealth(true);

// Person 58
ListOfPersons[57].setAge(31);
ListOfPersons[57].setDayRegistered(27);
ListOfPersons[57].setMonthRegistered(9);
ListOfPersons[57].setYearRegistered(2023);
ListOfPersons[57].setPersonClass(1);
ListOfPersons[57].setRightToTravel(false);
ListOfPersons[57].setLuggageWeight(14.8);
ListOfPersons[57].setCheckIfDisabled(false);
ListOfPersons[57].setMinimumPay(700);
ListOfPersons[57].setExtraPay(true);
ListOfPersons[57].setMilitary(false);
ListOfPersons[57].setInGoodHealth(true);

// Person 59
ListOfPersons[58].setAge(39);
ListOfPersons[58].setDayRegistered(8);
ListOfPersons[58].setMonthRegistered(6);
ListOfPersons[58].setYearRegistered(2023);
ListOfPersons[58].setPersonClass(2);
ListOfPersons[58].setRightToTravel(true);
ListOfPersons[58].setLuggageWeight(16.0);
```

```
   ListOfPersons[58].setCheckIfDisabled(false);
   ListOfPersons[58].setMinimumPay(710);
   ListOfPersons[58].setExtraPay(true);
   ListOfPersons[58].setMilitary(false);
   ListOfPersons[58].setInGoodHealth(true);

   // Person 60
   ListOfPersons[59].setAge(42);
   ListOfPersons[59].setDayRegistered(11);
   ListOfPersons[59].setMonthRegistered(11);
   ListOfPersons[59].setYearRegistered(2023);
   ListOfPersons[59].setPersonClass(3);
   ListOfPersons[59].setRightToTravel(true);
   ListOfPersons[59].setLuggageWeight(17.5);
   ListOfPersons[59].setCheckIfDisabled(false);
   ListOfPersons[59].setMinimumPay(725);
   ListOfPersons[59].setExtraPay(true);
   ListOfPersons[59].setMilitary(false);
   ListOfPersons[59].setInGoodHealth(true);

// Person 61
   ListOfPersons[60].setAge(35);
   ListOfPersons[60].setDayRegistered(18);
   ListOfPersons[60].setMonthRegistered(4);
   ListOfPersons[60].setYearRegistered(2023);
   ListOfPersons[60].setPersonClass(4);
   ListOfPersons[60].setRightToTravel(true);
   ListOfPersons[60].setLuggageWeight(19.0);
   ListOfPersons[60].setCheckIfDisabled(false);
   ListOfPersons[60].setMinimumPay(735);
   ListOfPersons[60].setExtraPay(true);
   ListOfPersons[60].setMilitary(false);
   ListOfPersons[60].setInGoodHealth(true);

   // Person 62
   ListOfPersons[61].setAge(29);
   ListOfPersons[61].setDayRegistered(2);
   ListOfPersons[61].setMonthRegistered(1);
   ListOfPersons[61].setYearRegistered(2023);
   ListOfPersons[61].setPersonClass(3);
```

```
ListOfPersons[61].setRightToTravel(true);
ListOfPersons[61].setLuggageWeight(18.2);
ListOfPersons[61].setCheckIfDisabled(false);
ListOfPersons[61].setMinimumPay(720);
ListOfPersons[61].setExtraPay(true);
ListOfPersons[61].setMilitary(false);
ListOfPersons[61].setInGoodHealth(true);

// Person 63
ListOfPersons[62].setAge(48);
ListOfPersons[62].setDayRegistered(9);
ListOfPersons[62].setMonthRegistered(8);
ListOfPersons[62].setYearRegistered(2023);
ListOfPersons[62].setPersonClass(1);
ListOfPersons[62].setRightToTravel(true);
ListOfPersons[62].setLuggageWeight(15.5);
ListOfPersons[62].setCheckIfDisabled(false);
ListOfPersons[62].setMinimumPay(705);
ListOfPersons[62].setExtraPay(true);
ListOfPersons[62].setMilitary(false);
ListOfPersons[62].setInGoodHealth(true);

// Person 64
ListOfPersons[63].setAge(38);
ListOfPersons[63].setDayRegistered(23);
ListOfPersons[63].setMonthRegistered(5);
ListOfPersons[63].setYearRegistered(2023);
ListOfPersons[63].setPersonClass(2);
ListOfPersons[63].setRightToTravel(true);
ListOfPersons[63].setLuggageWeight(16.8);
ListOfPersons[63].setCheckIfDisabled(false);
ListOfPersons[63].setMinimumPay(715);
ListOfPersons[63].setExtraPay(true);
ListOfPersons[63].setMilitary(false);
ListOfPersons[63].setInGoodHealth(true);

// Person 65
ListOfPersons[64].setAge(41);
ListOfPersons[64].setDayRegistered(14);
ListOfPersons[64].setMonthRegistered(12);
```

```
    ListOfPersons[64].setYearRegistered(2023);
    ListOfPersons[64].setPersonClass(3);
    ListOfPersons[64].setRightToTravel(true);
    ListOfPersons[64].setLuggageWeight(17.2);
    ListOfPersons[64].setCheckIfDisabled(false);
    ListOfPersons[64].setMinimumPay(730);
    ListOfPersons[64].setExtraPay(true);
    ListOfPersons[64].setMilitary(false);
    ListOfPersons[64].setInGoodHealth(true);

 // Person 66
    ListOfPersons[65].setAge(33);
    ListOfPersons[65].setDayRegistered(7);
    ListOfPersons[65].setMonthRegistered(9);
    ListOfPersons[65].setYearRegistered(2023);
    ListOfPersons[65].setPersonClass(4);
    ListOfPersons[65].setRightToTravel(true);
    ListOfPersons[65].setLuggageWeight(18.5);
    ListOfPersons[65].setCheckIfDisabled(false);
    ListOfPersons[65].setMinimumPay(725);
    ListOfPersons[65].setExtraPay(true);
    ListOfPersons[65].setMilitary(false);
    ListOfPersons[65].setInGoodHealth(true);

    // Person 67
    ListOfPersons[66].setAge(29);
    ListOfPersons[66].setDayRegistered(5);
    ListOfPersons[66].setMonthRegistered(3);
    ListOfPersons[66].setYearRegistered(2023);
    ListOfPersons[66].setPersonClass(3);
    ListOfPersons[66].setRightToTravel(true);
    ListOfPersons[66].setLuggageWeight(17.8);
    ListOfPersons[66].setCheckIfDisabled(false);
    ListOfPersons[66].setMinimumPay(720);
    ListOfPersons[66].setExtraPay(true);
    ListOfPersons[66].setMilitary(false);
    ListOfPersons[66].setInGoodHealth(true);

    // Person 68
    ListOfPersons[67].setAge(45);
```

```
ListOfPersons[67].setDayRegistered(12);
ListOfPersons[67].setMonthRegistered(6);
ListOfPersons[67].setYearRegistered(2023);
ListOfPersons[67].setPersonClass(2);
ListOfPersons[67].setRightToTravel(true);
ListOfPersons[67].setLuggageWeight(16.2);
ListOfPersons[67].setCheckIfDisabled(false);
ListOfPersons[67].setMinimumPay(710);
ListOfPersons[67].setExtraPay(true);
ListOfPersons[67].setMilitary(false);
ListOfPersons[67].setInGoodHealth(true);

// Person 69
ListOfPersons[68].setAge(37);
ListOfPersons[68].setDayRegistered(19);
ListOfPersons[68].setMonthRegistered(1);
ListOfPersons[68].setYearRegistered(2023);
ListOfPersons[68].setPersonClass(1);
ListOfPersons[68].setRightToTravel(true);
ListOfPersons[68].setLuggageWeight(15.8);
ListOfPersons[68].setCheckIfDisabled(false);
ListOfPersons[68].setMinimumPay(705);
ListOfPersons[68].setExtraPay(true);
ListOfPersons[68].setMilitary(false);
ListOfPersons[68].setInGoodHealth(true);

// Person 70
ListOfPersons[69].setAge(40);
ListOfPersons[69].setDayRegistered(25);
ListOfPersons[69].setMonthRegistered(11);
ListOfPersons[69].setYearRegistered(2023);
ListOfPersons[69].setPersonClass(3);
ListOfPersons[69].setRightToTravel(true);
ListOfPersons[69].setLuggageWeight(17.5);
ListOfPersons[69].setCheckIfDisabled(false);
ListOfPersons[69].setMinimumPay(725);
ListOfPersons[69].setExtraPay(true);
ListOfPersons[69].setMilitary(false);
ListOfPersons[69].setInGoodHealth(true);
```

```
// Person 71
   ListOfPersons[70].setAge(28);
   ListOfPersons[70].setDayRegistered(8);
   ListOfPersons[70].setMonthRegistered(2);
   ListOfPersons[70].setYearRegistered(2023);
   ListOfPersons[70].setPersonClass(4);
   ListOfPersons[70].setRightToTravel(true);
   ListOfPersons[70].setLuggageWeight(19.0);
   ListOfPersons[70].setCheckIfDisabled(false);
   ListOfPersons[70].setMinimumPay(730);
   ListOfPersons[70].setExtraPay(false);
   ListOfPersons[70].setMilitary(false);
   ListOfPersons[70].setInGoodHealth(true);

   // Person 72
   ListOfPersons[71].setAge(34);
   ListOfPersons[71].setDayRegistered(14);
   ListOfPersons[71].setMonthRegistered(5);
   ListOfPersons[71].setYearRegistered(2023);
   ListOfPersons[71].setPersonClass(3);
   ListOfPersons[71].setRightToTravel(true);
   ListOfPersons[71].setLuggageWeight(18.3);
   ListOfPersons[71].setCheckIfDisabled(false);
   ListOfPersons[71].setMinimumPay(715);
   ListOfPersons[71].setExtraPay(false);
   ListOfPersons[71].setMilitary(false);
   ListOfPersons[71].setInGoodHealth(true);

   // Person 73
   ListOfPersons[72].setAge(39);
   ListOfPersons[72].setDayRegistered(23);
   ListOfPersons[72].setMonthRegistered(10);
   ListOfPersons[72].setYearRegistered(2023);
   ListOfPersons[72].setPersonClass(2);
   ListOfPersons[72].setRightToTravel(true);
   ListOfPersons[72].setLuggageWeight(16.8);
   ListOfPersons[72].setCheckIfDisabled(false);
   ListOfPersons[72].setMinimumPay(710);
   ListOfPersons[72].setExtraPay(false);
   ListOfPersons[72].setMilitary(false);
   ListOfPersons[72].setInGoodHealth(true);

   // Person 74
   ListOfPersons[73].setAge(45);
   ListOfPersons[73].setDayRegistered(7);
   ListOfPersons[73].setMonthRegistered(4);
   ListOfPersons[73].setYearRegistered(2023);
   ListOfPersons[73].setPersonClass(1);
   ListOfPersons[73].setRightToTravel(true);
   ListOfPersons[73].setLuggageWeight(15.5);
   ListOfPersons[73].setCheckIfDisabled(false);
   ListOfPersons[73].setMinimumPay(705);
   ListOfPersons[73].setExtraPay(false);
   ListOfPersons[73].setMilitary(false);
   ListOfPersons[73].setInGoodHealth(true);

   // Person 75
   ListOfPersons[74].setAge(41);
   ListOfPersons[74].setDayRegistered(18);
   ListOfPersons[74].setMonthRegistered(12);
   ListOfPersons[74].setYearRegistered(2023);
   ListOfPersons[74].setPersonClass(3);
   ListOfPersons[74].setRightToTravel(true);
   ListOfPersons[74].setLuggageWeight(17.2);
   ListOfPersons[74].setCheckIfDisabled(false);
   ListOfPersons[74].setMinimumPay(725);
   ListOfPersons[74].setExtraPay(true);
   ListOfPersons[74].setMilitary(false);
   ListOfPersons[74].setInGoodHealth(true);

// Person 76
   ListOfPersons[75].setAge(32);
   ListOfPersons[75].setDayRegistered(12);
   ListOfPersons[75].setMonthRegistered(6);
   ListOfPersons[75].setYearRegistered(2023);
   ListOfPersons[75].setPersonClass(4);
   ListOfPersons[75].setRightToTravel(true);
   ListOfPersons[75].setLuggageWeight(18.7);
   ListOfPersons[75].setCheckIfDisabled(false);
   ListOfPersons[75].setMinimumPay(720);
```

```
ListOfPersons[75].setExtraPay(true);
ListOfPersons[75].setMilitary(false);
ListOfPersons[75].setInGoodHealth(true);

// Person 77
ListOfPersons[76].setAge(38);
ListOfPersons[76].setDayRegistered(25);
ListOfPersons[76].setMonthRegistered(11);
ListOfPersons[76].setYearRegistered(2023);
ListOfPersons[76].setPersonClass(3);
ListOfPersons[76].setRightToTravel(true);
ListOfPersons[76].setLuggageWeight(17.9);
ListOfPersons[76].setCheckIfDisabled(false);
ListOfPersons[76].setMinimumPay(715);
ListOfPersons[76].setExtraPay(false);
ListOfPersons[76].setMilitary(true);
ListOfPersons[76].setInGoodHealth(true);

// Person 78
ListOfPersons[77].setAge(43);
ListOfPersons[77].setDayRegistered(9);
ListOfPersons[77].setMonthRegistered(3);
ListOfPersons[77].setYearRegistered(2023);
ListOfPersons[77].setPersonClass(2);
ListOfPersons[77].setRightToTravel(true);
ListOfPersons[77].setLuggageWeight(16.3);
ListOfPersons[77].setCheckIfDisabled(false);
ListOfPersons[77].setMinimumPay(710);
ListOfPersons[77].setExtraPay(false);
ListOfPersons[77].setMilitary(true);
ListOfPersons[77].setInGoodHealth(true);

// Person 79
ListOfPersons[78].setAge(48);
ListOfPersons[78].setDayRegistered(16);
ListOfPersons[78].setMonthRegistered(7);
ListOfPersons[78].setYearRegistered(2023);
ListOfPersons[78].setPersonClass(1);
ListOfPersons[78].setRightToTravel(true);
ListOfPersons[78].setLuggageWeight(15.8);
```

```
   ListOfPersons[78].setCheckIfDisabled(false);
   ListOfPersons[78].setMinimumPay(705);
   ListOfPersons[78].setExtraPay(false);
   ListOfPersons[78].setMilitary(true);
   ListOfPersons[78].setInGoodHealth(true);

   // Person 80
   ListOfPersons[79].setAge(36);
   ListOfPersons[79].setDayRegistered(20);
   ListOfPersons[79].setMonthRegistered(2);
   ListOfPersons[79].setYearRegistered(2023);
   ListOfPersons[79].setPersonClass(3);
   ListOfPersons[79].setRightToTravel(true);
   ListOfPersons[79].setLuggageWeight(17.5);
   ListOfPersons[79].setCheckIfDisabled(false);
   ListOfPersons[79].setMinimumPay(725);
   ListOfPersons[79].setExtraPay(false);
   ListOfPersons[79].setMilitary(false);
   ListOfPersons[79].setInGoodHealth(true);

// Person 81
   ListOfPersons[80].setAge(41);
   ListOfPersons[80].setDayRegistered(5);
   ListOfPersons[80].setMonthRegistered(10);
   ListOfPersons[80].setYearRegistered(2023);
   ListOfPersons[80].setPersonClass(2);
   ListOfPersons[80].setRightToTravel(true);
   ListOfPersons[80].setLuggageWeight(16.9);
   ListOfPersons[80].setCheckIfDisabled(false);
   ListOfPersons[80].setMinimumPay(710);
   ListOfPersons[80].setExtraPay(false);
   ListOfPersons[80].setMilitary(true);
   ListOfPersons[80].setInGoodHealth(true);

   // Person 82
   ListOfPersons[81].setAge(37);
   ListOfPersons[81].setDayRegistered(13);
   ListOfPersons[81].setMonthRegistered(4);
   ListOfPersons[81].setYearRegistered(2023);
   ListOfPersons[81].setPersonClass(1);
```

```
ListOfPersons[81].setRightToTravel(true);
ListOfPersons[81].setLuggageWeight(15.6);
ListOfPersons[81].setCheckIfDisabled(false);
ListOfPersons[81].setMinimumPay(705);
ListOfPersons[81].setExtraPay(true);
ListOfPersons[81].setMilitary(false);
ListOfPersons[81].setInGoodHealth(false);

// Person 83
ListOfPersons[82].setAge(44);
ListOfPersons[82].setDayRegistered(18);
ListOfPersons[82].setMonthRegistered(8);
ListOfPersons[82].setYearRegistered(2023);
ListOfPersons[82].setPersonClass(4);
ListOfPersons[82].setRightToTravel(true);
ListOfPersons[82].setLuggageWeight(17.2);
ListOfPersons[82].setCheckIfDisabled(false);
ListOfPersons[82].setMinimumPay(725);
ListOfPersons[82].setExtraPay(true);
ListOfPersons[82].setMilitary(false);
ListOfPersons[82].setInGoodHealth(false);

// Person 84
ListOfPersons[83].setAge(35);
ListOfPersons[83].setDayRegistered(22);
ListOfPersons[83].setMonthRegistered(1);
ListOfPersons[83].setYearRegistered(2023);
ListOfPersons[83].setPersonClass(3);
ListOfPersons[83].setRightToTravel(true);
ListOfPersons[83].setLuggageWeight(16.4);
ListOfPersons[83].setCheckIfDisabled(false);
ListOfPersons[83].setMinimumPay(720);
ListOfPersons[83].setExtraPay(true);
ListOfPersons[83].setMilitary(false);
ListOfPersons[83].setInGoodHealth(false);

// Person 85
ListOfPersons[84].setAge(38);
ListOfPersons[84].setDayRegistered(7);
ListOfPersons[84].setMonthRegistered(6);
```

```
   ListOfPersons[84].setYearRegistered(2023);
   ListOfPersons[84].setPersonClass(2);
   ListOfPersons[84].setRightToTravel(true);
   ListOfPersons[84].setLuggageWeight(16.7);
   ListOfPersons[84].setCheckIfDisabled(false);
   ListOfPersons[84].setMinimumPay(710);
   ListOfPersons[84].setExtraPay(true);
   ListOfPersons[84].setMilitary(false);
   ListOfPersons[84].setInGoodHealth(true);

// Person 86
   ListOfPersons[85].setAge(42);
   ListOfPersons[85].setDayRegistered(11);
   ListOfPersons[85].setMonthRegistered(9);
   ListOfPersons[85].setYearRegistered(2023);
   ListOfPersons[85].setPersonClass(1);
   ListOfPersons[85].setRightToTravel(true);
   ListOfPersons[85].setLuggageWeight(15.8);
   ListOfPersons[85].setCheckIfDisabled(false);
   ListOfPersons[85].setMinimumPay(705);
   ListOfPersons[85].setExtraPay(true);
   ListOfPersons[85].setMilitary(false);
   ListOfPersons[85].setInGoodHealth(true);

   // Person 87
   ListOfPersons[86].setAge(39);
   ListOfPersons[86].setDayRegistered(17);
   ListOfPersons[86].setMonthRegistered(3);
   ListOfPersons[86].setYearRegistered(2023);
   ListOfPersons[86].setPersonClass(4);
   ListOfPersons[86].setRightToTravel(true);
   ListOfPersons[86].setLuggageWeight(17.0);
   ListOfPersons[86].setCheckIfDisabled(false);
   ListOfPersons[86].setMinimumPay(725);
   ListOfPersons[86].setExtraPay(true);
   ListOfPersons[86].setMilitary(false);
   ListOfPersons[86].setInGoodHealth(true);

   // Person 88
   ListOfPersons[87].setAge(36);
```

```
ListOfPersons[87].setDayRegistered(21);
ListOfPersons[87].setMonthRegistered(7);
ListOfPersons[87].setYearRegistered(2023);
ListOfPersons[87].setPersonClass(3);
ListOfPersons[87].setRightToTravel(true);
ListOfPersons[87].setLuggageWeight(16.6);
ListOfPersons[87].setCheckIfDisabled(false);
ListOfPersons[87].setMinimumPay(720);
ListOfPersons[87].setExtraPay(true);
ListOfPersons[87].setMilitary(false);
ListOfPersons[87].setInGoodHealth(true);

// Person 89
ListOfPersons[88].setAge(43);
ListOfPersons[88].setDayRegistered(5);
ListOfPersons[88].setMonthRegistered(11);
ListOfPersons[88].setYearRegistered(2023);
ListOfPersons[88].setPersonClass(2);
ListOfPersons[88].setRightToTravel(true);
ListOfPersons[88].setLuggageWeight(16.9);
ListOfPersons[88].setCheckIfDisabled(false);
ListOfPersons[88].setMinimumPay(710);
ListOfPersons[88].setExtraPay(true);
ListOfPersons[88].setMilitary(false);
ListOfPersons[88].setInGoodHealth(true);

// Person 90
ListOfPersons[89].setAge(40);
ListOfPersons[89].setDayRegistered(15);
ListOfPersons[89].setMonthRegistered(5);
ListOfPersons[89].setYearRegistered(2023);
ListOfPersons[89].setPersonClass(1);
ListOfPersons[89].setRightToTravel(true);
ListOfPersons[89].setLuggageWeight(15.5);
ListOfPersons[89].setCheckIfDisabled(false);
ListOfPersons[89].setMinimumPay(705);
ListOfPersons[89].setExtraPay(true);
ListOfPersons[89].setMilitary(false);
ListOfPersons[89].setInGoodHealth(true);
```

```
// Person 91
   ListOfPersons[90].setAge(80);
   ListOfPersons[90].setDayRegistered(19);
   ListOfPersons[90].setMonthRegistered(2);
   ListOfPersons[90].setYearRegistered(2023);
   ListOfPersons[90].setPersonClass(4);
   ListOfPersons[90].setRightToTravel(true);
   ListOfPersons[90].setLuggageWeight(17.3);
   ListOfPersons[90].setCheckIfDisabled(false);
   ListOfPersons[90].setMinimumPay(725);
   ListOfPersons[90].setExtraPay(true);
   ListOfPersons[90].setMilitary(false);
   ListOfPersons[90].setInGoodHealth(true);

   // Person 92
   ListOfPersons[91].setAge(92);
   ListOfPersons[91].setDayRegistered(8);
   ListOfPersons[91].setMonthRegistered(6);
   ListOfPersons[91].setYearRegistered(2023);
   ListOfPersons[91].setPersonClass(3);
   ListOfPersons[91].setRightToTravel(true);
   ListOfPersons[91].setLuggageWeight(16.8);
   ListOfPersons[91].setCheckIfDisabled(false);
   ListOfPersons[91].setMinimumPay(720);
   ListOfPersons[91].setExtraPay(true);
   ListOfPersons[91].setMilitary(false);
   ListOfPersons[91].setInGoodHealth(true);

   // Person 93
   ListOfPersons[92].setAge(80);
   ListOfPersons[92].setDayRegistered(23);
   ListOfPersons[92].setMonthRegistered(10);
   ListOfPersons[92].setYearRegistered(2023);
   ListOfPersons[92].setPersonClass(2);
   ListOfPersons[92].setRightToTravel(true);
   ListOfPersons[92].setLuggageWeight(17.1);
   ListOfPersons[92].setCheckIfDisabled(false);
   ListOfPersons[92].setMinimumPay(710);
   ListOfPersons[92].setExtraPay(true);
   ListOfPersons[92].setMilitary(false);
   ListOfPersons[92].setInGoodHealth(true);

   // Person 94
   ListOfPersons[93].setAge(39);
   ListOfPersons[93].setDayRegistered(12);
   ListOfPersons[93].setMonthRegistered(4);
   ListOfPersons[93].setYearRegistered(2023);
   ListOfPersons[93].setPersonClass(1);
   ListOfPersons[93].setRightToTravel(true);
   ListOfPersons[93].setLuggageWeight(15.9);
   ListOfPersons[93].setCheckIfDisabled(false);
   ListOfPersons[93].setMinimumPay(705);
   ListOfPersons[93].setExtraPay(true);
   ListOfPersons[93].setMilitary(false);
   ListOfPersons[93].setInGoodHealth(true);

   // Person 95
   ListOfPersons[94].setAge(37);
   ListOfPersons[94].setDayRegistered(14);
   ListOfPersons[94].setMonthRegistered(8);
   ListOfPersons[94].setYearRegistered(2023);
   ListOfPersons[94].setPersonClass(4);
   ListOfPersons[94].setRightToTravel(true);
   ListOfPersons[94].setLuggageWeight(17.5);
   ListOfPersons[94].setCheckIfDisabled(false);
   ListOfPersons[94].setMinimumPay(725);
   ListOfPersons[94].setExtraPay(true);
   ListOfPersons[94].setMilitary(false);
   ListOfPersons[94].setInGoodHealth(true);

// Person 96
   ListOfPersons[95].setAge(41);
   ListOfPersons[95].setDayRegistered(7);
   ListOfPersons[95].setMonthRegistered(3);
   ListOfPersons[95].setYearRegistered(2023);
   ListOfPersons[95].setPersonClass(3);
   ListOfPersons[95].setRightToTravel(true);
   ListOfPersons[95].setLuggageWeight(16.2);
   ListOfPersons[95].setCheckIfDisabled(false);
   ListOfPersons[95].setMinimumPay(720);
```

```
ListOfPersons[95].setExtraPay(true);
ListOfPersons[95].setMilitary(false);
ListOfPersons[95].setInGoodHealth(true);

// Person 97
ListOfPersons[96].setAge(44);
ListOfPersons[96].setDayRegistered(25);
ListOfPersons[96].setMonthRegistered(7);
ListOfPersons[96].setYearRegistered(2023);
ListOfPersons[96].setPersonClass(2);
ListOfPersons[96].setRightToTravel(true);
ListOfPersons[96].setLuggageWeight(16.7);
ListOfPersons[96].setCheckIfDisabled(false);
ListOfPersons[96].setMinimumPay(710);
ListOfPersons[96].setExtraPay(true);
ListOfPersons[96].setMilitary(false);
ListOfPersons[96].setInGoodHealth(true);

// Person 98
ListOfPersons[97].setAge(40);
ListOfPersons[97].setDayRegistered(17);
ListOfPersons[97].setMonthRegistered(11);
ListOfPersons[97].setYearRegistered(2023);
ListOfPersons[97].setPersonClass(1);
ListOfPersons[97].setRightToTravel(true);
ListOfPersons[97].setLuggageWeight(15.5);
ListOfPersons[97].setCheckIfDisabled(false);
ListOfPersons[97].setMinimumPay(705);
ListOfPersons[97].setExtraPay(true);
ListOfPersons[97].setMilitary(false);
ListOfPersons[97].setInGoodHealth(true);

// Person 99
ListOfPersons[98].setAge(43);
ListOfPersons[98].setDayRegistered(9);
ListOfPersons[98].setMonthRegistered(5);
ListOfPersons[98].setYearRegistered(2023);
ListOfPersons[98].setPersonClass(4);
ListOfPersons[98].setRightToTravel(true);
ListOfPersons[98].setLuggageWeight(17.8);
```

```
   ListOfPersons[98].setCheckIfDisabled(false);
   ListOfPersons[98].setMinimumPay(725);
   ListOfPersons[98].setExtraPay(true);
   ListOfPersons[98].setMilitary(false);
   ListOfPersons[98].setInGoodHealth(true);

   // Person 100
   ListOfPersons[99].setAge(38);
   ListOfPersons[99].setDayRegistered(22);
   ListOfPersons[99].setMonthRegistered(9);
   ListOfPersons[99].setYearRegistered(2023);
   ListOfPersons[99].setPersonClass(3);
   ListOfPersons[99].setRightToTravel(true);
   ListOfPersons[99].setLuggageWeight(16.9);
   ListOfPersons[99].setCheckIfDisabled(false);
   ListOfPersons[99].setMinimumPay(720);
   ListOfPersons[99].setExtraPay(true);
   ListOfPersons[99].setMilitary(false);
   ListOfPersons[99].setInGoodHealth(true);

// Person 101
   ListOfPersons[100].setAge(41);
   ListOfPersons[100].setDayRegistered(3);
   ListOfPersons[100].setMonthRegistered(2);
   ListOfPersons[100].setYearRegistered(2023);
   ListOfPersons[100].setPersonClass(1);
   ListOfPersons[100].setRightToTravel(true);
   ListOfPersons[100].setLuggageWeight(15.3);
   ListOfPersons[100].setCheckIfDisabled(false);
   ListOfPersons[100].setMinimumPay(705);
   ListOfPersons[100].setExtraPay(true);
   ListOfPersons[100].setMilitary(false);
   ListOfPersons[100].setInGoodHealth(true);

   // Person 102
   ListOfPersons[101].setAge(39);
   ListOfPersons[101].setDayRegistered(14);
   ListOfPersons[101].setMonthRegistered(6);
   ListOfPersons[101].setYearRegistered(2023);
   ListOfPersons[101].setPersonClass(4);
```

```
ListOfPersons[101].setRightToTravel(true);
ListOfPersons[101].setLuggageWeight(17.2);
ListOfPersons[101].setCheckIfDisabled(false);
ListOfPersons[101].setMinimumPay(725);
ListOfPersons[101].setExtraPay(true);
ListOfPersons[101].setMilitary(false);
ListOfPersons[101].setInGoodHealth(true);

// Person 103
ListOfPersons[102].setAge(40);
ListOfPersons[102].setDayRegistered(9);
ListOfPersons[102].setMonthRegistered(12);
ListOfPersons[102].setYearRegistered(2023);
ListOfPersons[102].setPersonClass(3);
ListOfPersons[102].setRightToTravel(true);
ListOfPersons[102].setLuggageWeight(16.8);
ListOfPersons[102].setCheckIfDisabled(false);
ListOfPersons[102].setMinimumPay(710);
ListOfPersons[102].setExtraPay(true);
ListOfPersons[102].setMilitary(false);
ListOfPersons[102].setInGoodHealth(true);

// Person 104
ListOfPersons[103].setAge(42);
ListOfPersons[103].setDayRegistered(20);
ListOfPersons[103].setMonthRegistered(4);
ListOfPersons[103].setYearRegistered(2023);
ListOfPersons[103].setPersonClass(2);
ListOfPersons[103].setRightToTravel(true);
ListOfPersons[103].setLuggageWeight(16.4);
ListOfPersons[103].setCheckIfDisabled(false);
ListOfPersons[103].setMinimumPay(715);
ListOfPersons[103].setExtraPay(true);
ListOfPersons[103].setMilitary(false);
ListOfPersons[103].setInGoodHealth(true);

// Person 105
ListOfPersons[104].setAge(39);
ListOfPersons[104].setDayRegistered(11);
ListOfPersons[104].setMonthRegistered(7);
```

```
        ListOfPersons[104].setYearRegistered(2023);
        ListOfPersons[104].setPersonClass(1);
        ListOfPersons[104].setRightToTravel(true);
        ListOfPersons[104].setLuggageWeight(15.6);
        ListOfPersons[104].setCheckIfDisabled(false);
        ListOfPersons[104].setMinimumPay(700);
        ListOfPersons[104].setExtraPay(true);
        ListOfPersons[104].setMilitary(false);
        ListOfPersons[104].setInGoodHealth(true);

     // Person 106
        ListOfPersons[105].setAge(35);
        ListOfPersons[105].setDayRegistered(8);
        ListOfPersons[105].setMonthRegistered(11);
        ListOfPersons[105].setYearRegistered(2023);
        ListOfPersons[105].setPersonClass(4);
        ListOfPersons[105].setRightToTravel(true);
        ListOfPersons[105].setLuggageWeight(14.5);
        ListOfPersons[105].setCheckIfDisabled(false);
        ListOfPersons[105].setMinimumPay(730);
        ListOfPersons[105].setExtraPay(true);
        ListOfPersons[105].setMilitary(false);
        ListOfPersons[105].setInGoodHealth(true);

        // Person 107
        ListOfPersons[106].setAge(38);
        ListOfPersons[106].setDayRegistered(2);
        ListOfPersons[106].setMonthRegistered(9);
        ListOfPersons[106].setYearRegistered(2023);
        ListOfPersons[106].setPersonClass(3);
        ListOfPersons[106].setRightToTravel(true);
        ListOfPersons[106].setLuggageWeight(15.8);
        ListOfPersons[106].setCheckIfDisabled(false);
        ListOfPersons[106].setMinimumPay(720);
        ListOfPersons[106].setExtraPay(true);
        ListOfPersons[106].setMilitary(false);
        ListOfPersons[106].setInGoodHealth(true);

        // Person 108
        ListOfPersons[107].setAge(37);
```

```
ListOfPersons[107].setDayRegistered(17);
ListOfPersons[107].setMonthRegistered(3);
ListOfPersons[107].setYearRegistered(2023);
ListOfPersons[107].setPersonClass(2);
ListOfPersons[107].setRightToTravel(true);
ListOfPersons[107].setLuggageWeight(16.2);
ListOfPersons[107].setCheckIfDisabled(false);
ListOfPersons[107].setMinimumPay(725);
ListOfPersons[107].setExtraPay(true);
ListOfPersons[107].setMilitary(false);
ListOfPersons[107].setInGoodHealth(true);

// Person 109
ListOfPersons[108].setAge(36);
ListOfPersons[108].setDayRegistered(23);
ListOfPersons[108].setMonthRegistered(5);
ListOfPersons[108].setYearRegistered(2023);
ListOfPersons[108].setPersonClass(1);
ListOfPersons[108].setRightToTravel(true);
ListOfPersons[108].setLuggageWeight(15.3);
ListOfPersons[108].setCheckIfDisabled(false);
ListOfPersons[108].setMinimumPay(710);
ListOfPersons[108].setExtraPay(true);
ListOfPersons[108].setMilitary(false);
ListOfPersons[108].setInGoodHealth(true);

// Person 110
ListOfPersons[109].setAge(34);
ListOfPersons[109].setDayRegistered(14);
ListOfPersons[109].setMonthRegistered(8);
ListOfPersons[109].setYearRegistered(2023);
ListOfPersons[109].setPersonClass(4);
ListOfPersons[109].setRightToTravel(true);
ListOfPersons[109].setLuggageWeight(14.9);
ListOfPersons[109].setCheckIfDisabled(false);
ListOfPersons[109].setMinimumPay(735);
ListOfPersons[109].setExtraPay(true);
ListOfPersons[109].setMilitary(false);
ListOfPersons[109].setInGoodHealth(true);
for(int i=0;i<20;i++)
```

```
         S.insertAtBack(ListOfPersons[i]);
        for(int i=0;i<5;i++)
         S.removeFromFront();
        System.out.println("The security line:");
        S.printSecurityLine();
        for(int i=20;i<45;i++)
         S.insertAtBack(ListOfPersons[i]);
        for(int i=0;i<15;i++)
         S.removeFromFront();
        for(int i=45;i<80;i++)
         S.insertAtBack(ListOfPersons[i]);
        for(int i=0;i<20;i++)
         S.removeFromFront();
        System.out.println("The security line:");
        S.printSecurityLine();
        System.out.println("The security recieves a 
call that the fourth person in the line is a terrorist 
and they arrest that person");
        S.removeKPerson(3);
        System.out.println("The security line now 
becomes");
        S.printSecurityLine();
        for(int i=80;i<110;i++)
         S.insertAtBack(ListOfPersons[i]);
        for(int i=0;i<59;i++)
         S.removeFromFront();
        System.out.println("The security line now 
becomes:");
        S.printSecurityLine();
        System.out.println("\nOkay now for security 
reasons the airport decided to empty the whole line");
        S.clearSecurityLine();
        System.out.println("Test if the security line 
is empty?");
        System.out.println(S.isEmpty());


        for(int i=0;i<ListOfPersons.length;i++)
        {
         if(ListOfPersons[i].getBusinessClass())
```

```
    BusinessClass.addToAirplane(ListOfPersons[i]);
            else
                if(ListOfPersons[i].isMilitary())

    MilitaryClass.addToAirplane(ListOfPersons[i]);
                else

    if(ListOfPersons[i].CheckIfDisabled())

    ClassForDisabledPeople.addToAirplane(ListOfPersons[
i]);
                    else

    if(ListOfPersons[i].isExtraPay())

    EconomyPlusClass.addToAirplane(ListOfPersons[i]);
                        else

    EconomyClass.addToAirplane(ListOfPersons[i]);

        }

        System.out.println("The number of passengers in
the economy class is:
"+EconomyClass.countPassengers());
        System.out.println("The number of passengers in
the Business class is:
"+BusinessClass.countPassengers());
        System.out.println("The number of passengers in
the Military class is:
"+MilitaryClass.countPassengers());
        System.out.println("The number of passengers in
the Disabled people class is:
"+ClassForDisabledPeople.countPassengers());
        System.out.println("The number of passengers in
the economy plus class is:
"+EconomyPlusClass.countPassengers());
```

```
        System.out.println("\nThe binary search tree of
the economy class is:");
        EconomyClass.displayAsBST();
        System.out.println("\nThe binary search tree of
the Business class is:");
        BusinessClass.displayAsBST();
        System.out.println("\nThe binary search tree of
the Military class is:");
        MilitaryClass.displayAsBST();
        System.out.println("\nThe binary search tree of
the Disabled people class is:");
        ClassForDisabledPeople.displayAsBST();
        System.out.println("\nThe binary search tree of
the economy plus class is:");
        EconomyPlusClass.displayAsBST();
        System.out.println("The old passengers in the
economy plus section:");
        EconomyPlusClass.displayOldPassengers();
        System.out.println("What is the height of the
economy class as binary search tree?");
        System.out.println(EconomyClass.getheight());
        System.out.println("The staff wants to double
check if a passenger is on the airplane(economy
section)?");

System.out.println(EconomyClass.search(ListOfPersons[33
]));}
}
```

Console:

```
John Doe joined the security line
Jacob Adams joined the security line
Jane Smith joined the security line
Alex Johnson joined the security line
Emily Brown joined the security line
Michael Davis joined the security line
Olivia White joined the security line
Daniel Miller joined the security line
Grace Wilson joined the security line
Matthew Taylor joined the security line
```

```
Ava Martinez joined the security line
William Harris joined the security line
Abigail Martin joined the security line
James Anderson joined the security line
Avery Davis joined the security line
Emma Garcia joined the security line
Ethan Turner joined the security line
Natalie Mitchell joined the security line
Oliver Cooper joined the security line
Grace King joined the security line
John Doe has passed the security test
Jacob Adams has passed the security test
Jane Smith has not passed the security test
Alex Johnson has passed the security test
Emily Brown has passed the security test
The security line:
Michael Davis (Age: 32) <--- Olivia White (Age: 40) <--
- Daniel Miller (Age: 48) <--- Grace Wilson (Age: 33)
<--- Matthew Taylor (Age: 42) <--- Ava Martinez (Age:
37) <--- William Harris (Age: 45) <--- Abigail Martin
(Age: 39) <--- James Anderson (Age: 41) <--- Avery
Davis (Age: 38) <--- Emma Garcia (Age: 44) <--- Ethan
Turner (Age: 40) <--- Natalie Mitchell (Age: 36) <---
Oliver Cooper (Age: 42) <--- Grace King (Age:
35)Jackson Moore joined the security line
Lucas Turner joined the security line
Samantha Mitchell joined the security line
Aiden Cooper joined the security line
Hannah King joined the security line
Caleb Moore joined the security line
Daniel Turner joined the security line
Hailey Mitchell joined the security line
Leo Cooper joined the security line
Mia King joined the security line
Evan Moore joined the security line
Aria Turner joined the security line
Logan Mitchell joined the security line
Lila Cooper joined the security line
Oscar King joined the security line
Ava Moore joined the security line
```

```
Nathan Turner joined the security line
Grace Mitchell joined the security line
Carter Cooper joined the security line
Sophie King joined the security line
Liam Moore joined the security line
Liam Turner joined the security line
Ella Mitchell joined the security line
Henry Cooper joined the security line
Ava King joined the security line
Michael Davis has passed the security test
Olivia White has passed the security test
Daniel Miller has passed the security test
Grace Wilson has passed the security test
Matthew Taylor has passed the security test
Ava Martinez has not passed the security test
William Harris has passed the security test
Abigail Martin has passed the security test
James Anderson has passed the security test
Avery Davis has passed the security test
Emma Garcia has not passed the security test
Ethan Turner has passed the security test
Natalie Mitchell has passed the security test
Oliver Cooper has passed the security test
Grace King has passed the security test
Mia Moore joined the security line
Logan Turner joined the security line
Sophia Mitchell joined the security line
Ethan Cooper joined the security line
Emma King joined the security line
Oscar Moore joined the security line
Aiden Turner joined the security line
Zoe Mitchell joined the security line
Grace Cooper joined the security line
Noah King joined the security line
Chloe Moore joined the security line
Olivia Turner joined the security line
Liam Mitchell joined the security line
Aria Cooper joined the security line
Carter King joined the security line
Lily Moore joined the security line
```

```
Ella Turner joined the security line
Lucas Mitchell joined the security line
Aiden Cooper joined the security line
Ava King joined the security line
Ethan Moore joined the security line
Sophie Turner joined the security line
Daniel Mitchell joined the security line
Grace Cooper joined the security line
Matthew King joined the security line
Ava Moore joined the security line
Olivia Turner joined the security line
Daniel Mitchell joined the security line
Grace Cooper joined the security line
Matthew King joined the security line
Ava Moore joined the security line
Liam Turner joined the security line
Ella Mitchell joined the security line
Henry Cooper joined the security line
Ava King joined the security line
Jackson Moore has passed the security test
Lucas Turner has passed the security test
Samantha Mitchell has not passed the security test
Aiden Cooper has passed the security test
Hannah King has passed the security test
Caleb Moore has passed the security test
Daniel Turner has passed the security test
Hailey Mitchell has passed the security test
Leo Cooper has passed the security test
Mia King has passed the security test
Evan Moore has passed the security test
Aria Turner has passed the security test
Logan Mitchell has not passed the security test
Lila Cooper has passed the security test
Oscar King has passed the security test
Ava Moore has passed the security test
Nathan Turner has passed the security test
Grace Mitchell has passed the security test
Carter Cooper has not passed the security test
Sophie King has not passed the security test
The security line:
```

Liam Moore (Age: 33) <--- Liam Turner (Age: 57) <--- Ella Mitchell (Age: 39) <--- Henry Cooper (Age: 48) <--- Ava King (Age: 29) <--- Mia Moore (Age: 31) <--- Logan Turner (Age: 42) <--- Sophia Mitchell (Age: 54) <--- Ethan Cooper (Age: 36) <--- Emma King (Age: 45) <--- Oscar Moore (Age: 28) <--- Aiden Turner (Age: 33) <--- Zoe Mitchell (Age: 40) <--- Grace Cooper (Age: 29) <--- Noah King (Age: 48) <--- Chloe Moore (Age: 36) <--- Olivia Turner (Age: 45) <--- Liam Mitchell (Age: 31) <--- Aria Cooper (Age: 39) <--- Carter King (Age: 42) <--- Lily Moore (Age: 35) <--- Ella Turner (Age: 29) <--- Lucas Mitchell (Age: 48) <--- Aiden Cooper (Age: 38) <--- Ava King (Age: 41) <--- Ethan Moore (Age: 33) <--- Sophie Turner (Age: 29) <--- Daniel Mitchell (Age: 45) <--- Grace Cooper (Age: 37) <--- Matthew King (Age: 40) <--- Ava Moore (Age: 28) <--- Olivia Turner (Age: 34) <--- Daniel Mitchell (Age: 39) <--- Grace Cooper (Age: 45) <--- Matthew King (Age: 41) <--- Ava Moore (Age: 32) <--- Liam Turner (Age: 38) <--- Ella Mitchell (Age: 43) <--- Henry Cooper (Age: 48) <--- Ava King (Age: 36)The security recieves a call that the fourth person in the line is a terrorist and they arrest that person

The security line now becomes

Liam Moore (Age: 33) <--- Liam Turner (Age: 57) <--- Henry Cooper (Age: 48) <--- Ava King (Age: 29) <--- Mia Moore (Age: 31) <--- Logan Turner (Age: 42) <--- Sophia Mitchell (Age: 54) <--- Ethan Cooper (Age: 36) <--- Emma King (Age: 45) <--- Oscar Moore (Age: 28) <--- Aiden Turner (Age: 33) <--- Zoe Mitchell (Age: 40) <--- Grace Cooper (Age: 29) <--- Noah King (Age: 48) <--- Chloe Moore (Age: 36) <--- Olivia Turner (Age: 45) <--- Liam Mitchell (Age: 31) <--- Aria Cooper (Age: 39) <--- Carter King (Age: 42) <--- Lily Moore (Age: 35) <--- Ella Turner (Age: 29) <--- Lucas Mitchell (Age: 48) <--- Aiden Cooper (Age: 38) <--- Ava King (Age: 41) <--- Ethan Moore (Age: 33) <--- Sophie Turner (Age: 29) <--- Daniel Mitchell (Age: 45) <--- Grace Cooper (Age: 37) <--- Matthew King (Age: 40) <--- Ava Moore (Age: 28) <--- Olivia Turner (Age: 34) <--- Daniel Mitchell (Age: 39) <--- Grace Cooper (Age: 45) <--- Matthew King (Age:

```
41) <--- Ava Moore (Age: 32) <--- Liam Turner (Age: 38)
<--- Ella Mitchell (Age: 43) <--- Henry Cooper (Age:
48) <--- Ava King (Age: 36)Mia Moore joined the
security line
Logan Turner joined the security line
Sophia Mitchell joined the security line
Ethan Cooper joined the security line
Emma King joined the security line
Oscar Moore joined the security line
Aiden Turner joined the security line
Zoe Mitchell joined the security line
Grace Cooper joined the security line
Noah King joined the security line
Chloe Moore joined the security line
Olivia Turner joined the security line
Liam Mitchell joined the security line
Aria Cooper joined the security line
Carter King joined the security line
Emma Turner joined the security line
Aiden Mitchell joined the security line
Olivia Cooper joined the security line
Lucas King joined the security line
Chloe Moore joined the security line
Mason Turner joined the security line
Zoe Mitchell joined the security line
Elijah Cooper joined the security line
Lily King joined the security line
Owen Moore joined the security line
Abigail Turner joined the security line
Jackson Mitchell joined the security line
Sofia Cooper joined the security line
Noah King joined the security line
Emily Moore joined the security line
Liam Moore has passed the security test
Liam Turner has passed the security test
Henry Cooper has passed the security test
Ava King has passed the security test
Mia Moore has passed the security test
Logan Turner has passed the security test
Sophia Mitchell has not passed the security test
```

```
Ethan Cooper has passed the security test
Emma King has passed the security test
Oscar Moore has passed the security test
Aiden Turner has passed the security test
Zoe Mitchell has passed the security test
Grace Cooper has passed the security test
Noah King has passed the security test
Chloe Moore has passed the security test
Olivia Turner has passed the security test
Liam Mitchell has passed the security test
Aria Cooper has passed the security test
Carter King has passed the security test
Lily Moore has passed the security test
Ella Turner has passed the security test
Lucas Mitchell has not passed the security test
Aiden Cooper has passed the security test
Ava King has passed the security test
Ethan Moore has passed the security test
Sophie Turner has passed the security test
Daniel Mitchell has not passed the security test
Grace Cooper has passed the security test
Matthew King has passed the security test
Ava Moore has passed the security test
Olivia Turner has passed the security test
Daniel Mitchell has passed the security test
Grace Cooper has passed the security test
Matthew King has passed the security test
Ava Moore has passed the security test
Liam Turner has passed the security test
Ella Mitchell has passed the security test
Henry Cooper has not passed the security test
Ava King has passed the security test
Mia Moore has passed the security test
Logan Turner has passed the security test
Sophia Mitchell has passed the security test
Ethan Cooper has passed the security test
Emma King has passed the security test
Oscar Moore has passed the security test
Aiden Turner has passed the security test
Zoe Mitchell has not passed the security test
```

```
Grace Cooper has passed the security test
Noah King has passed the security test
Chloe Moore has passed the security test
Olivia Turner has passed the security test
Liam Mitchell has passed the security test
Aria Cooper has not passed the security test
Carter King has passed the security test
Emma Turner has passed the security test
Aiden Mitchell has passed the security test
Olivia Cooper has passed the security test
Lucas King has passed the security test
Chloe Moore has passed the security test
The security line now becomes:
Mason Turner (Age: 41) <--- Zoe Mitchell (Age: 39) <---
Elijah Cooper (Age: 40) <--- Lily King (Age: 42) <---
Owen Moore (Age: 39) <--- Abigail Turner (Age: 35) <---
Jackson Mitchell (Age: 38) <--- Sofia Cooper (Age: 37)
<--- Noah King (Age: 36) <--- Emily Moore (Age: 34)
Okay now for security reasons the airport decided to
empty the whole line
Test if the security line is empty?
true
Jacob Adams got on the plane
Jane Smith can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Emily Brown got on the plane
Michael Davis got on the plane
Olivia White got on the plane
Daniel Miller got on the plane
Grace Wilson can't get on the airplane because:
This person was late to register
Matthew Taylor can't get on the airplane because:
The person's health doesn't permit to get on the plane
Ava Martinez can't get on the airplane because:
This person was late to register
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
```

```
William Harris can't get on the airplane because:
The person's health doesn't permit to get on the plane
Abigail Martin can't get on the airplane because:
The person's health doesn't permit to get on the plane
James Anderson got on the plane
Avery Davis got on the plane
Emma Garcia can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Ethan Turner got on the plane
Natalie Mitchell got on the plane
Oliver Cooper got on the plane
Grace King got on the plane
Lucas Turner got on the plane
Samantha Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Aiden Cooper got on the plane
Caleb Moore got on the plane
Daniel Turner got on the plane
Hailey Mitchell got on the plane
Leo Cooper got on the plane
Mia King got on the plane
Evan Moore got on the plane
Aria Turner got on the plane
Logan Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Lila Cooper got on the plane
Oscar King got on the plane
Ava Moore got on the plane
Nathan Turner got on the plane
Grace Mitchell got on the plane
Carter Cooper can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
```

```
Sophie King can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Liam Moore got on the plane
Liam Turner got on the plane
Ella Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Henry Cooper got on the plane
Ava King got on the plane
Mia Moore got on the plane
Logan Turner got on the plane
Sophia Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Ethan Cooper got on the plane
Emma King got on the plane
Oscar Moore got on the plane
Aiden Turner got on the plane
Zoe Mitchell can't get on the airplane because:
This person doesn't have the right to travel
Grace Cooper got on the plane
Noah King can't get on the airplane because:
This person doesn't have the right to travel
Chloe Moore got on the plane
Olivia Turner got on the plane
Liam Mitchell can't get on the airplane because:
This person doesn't have the right to travel
Aria Cooper got on the plane
Carter King got on the plane
Lily Moore got on the plane
Ella Turner got on the plane
Lucas Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Aiden Cooper got on the plane
```

```
Ava King got on the plane
Ethan Moore got on the plane
Sophie Turner got on the plane
Daniel Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Grace Cooper got on the plane
Matthew King got on the plane
Ava Moore got on the plane
Olivia Turner got on the plane
Daniel Mitchell got on the plane
Grace Cooper got on the plane
Matthew King got on the plane
Ava Moore got on the plane
Ella Mitchell got on the plane
Henry Cooper can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Ava King got on the plane
Mia Moore got on the plane
Logan Turner can't get on the airplane because:
The person's health doesn't permit to get on the plane
Sophia Mitchell can't get on the airplane because:
The person's health doesn't permit to get on the plane
Ethan Cooper can't get on the airplane because:
The person's health doesn't permit to get on the plane
Emma King got on the plane
Oscar Moore got on the plane
Aiden Turner got on the plane
Zoe Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Grace Cooper got on the plane
Noah King got on the plane
Chloe Moore got on the plane
Olivia Turner got on the plane
Liam Mitchell got on the plane
```

```
Aria Cooper can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Carter King got on the plane
Emma Turner got on the plane
Aiden Mitchell got on the plane
Olivia Cooper got on the plane
Lucas King got on the plane
Chloe Moore got on the plane
Mason Turner can't get on the airplane because:
This section is full
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Zoe Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Elijah Cooper can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Lily King can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Owen Moore can't get on the airplane because:
This section is full
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Abigail Turner can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Jackson Mitchell can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
```

```
Sofia Cooper can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Noah King can't get on the airplane because:
This section is full
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
Emily Moore can't get on the airplane because:
This person got caught in the security line with a
banned thing or he/she didn't get checked in the
security line
The number of passengers in the economy class is: 19
The number of passengers in the Business class is: 20
The number of passengers in the Military class is: 3
The number of passengers in the Disabled people class
is: 6
The number of passengers in the economy plus class is:
28

The binary search tree of the economy class is:
            Name: Ava Moore-- Age: 28
                Name: Ava King-- Age: 36
        Name: Nathan Turner-- Age: 54
            Name: Olivia Turner-- Age: 34
    Name: Natalie Mitchell-- Age: 36
            Name: Ethan Cooper-- Age: 36
        Name: Lila Cooper-- Age: 48
            Name: Liam Moore-- Age: 33
                Name: Henry Cooper-- Age: 48
Name: Alex Johnson-- Age: 45
            Name: Oliver Cooper-- Age: 42
        Name: Avery Davis-- Age: 38
    Name: Olivia White-- Age: 40
                Name: Logan Turner-- Age: 42
            Name: Oscar King-- Age: 41
                Name: Ava King-- Age: 29
                    Name: Daniel Mitchell-- Age: 39
        Name: Daniel Miller-- Age: 48
```

```
            Name: Grace Mitchell-- Age: 38

The binary search tree of the Business class is:
        Name: Caleb Moore-- Age: 55
                    Name: Grace Cooper-- Age: 37
                Name: Mia Moore-- Age: 31
            Name: Leo Cooper-- Age: 58
                Name: Ava Moore-- Age: 29
    Name: Emily Brown-- Age: 28
        Name: Ethan Turner-- Age: 40
Name: John Doe-- Age: 25
        Name: Grace King-- Age: 35
            Name: Grace Cooper-- Age: 45
    Name: Jacob Adams-- Age: 30
                Name: Noah King-- Age: 40
            Name: James Anderson-- Age: 41
        Name: Michael Davis-- Age: 32
            Name: Aria Turner-- Age: 62
                            Name: Oscar Moore-- Age:
42
                        Name: Grace Cooper-- Age: 29
                    Name: Emma King-- Age: 45
                        Name: Olivia Cooper-- Age: 40
                Name: Liam Turner-- Age: 57

The binary search tree of the Military class is:
    Name: Ella Mitchell-- Age: 43
        Name: Mia Moore-- Age: 41
Name: Liam Turner-- Age: 38

The binary search tree of the Disabled people class is:
    Name: Aiden Cooper-- Age: 32
            Name: Evan Moore-- Age: 25
        Name: Mia King-- Age: 40
Name: Jackson Moore-- Age: 50
        Name: Hailey Mitchell-- Age: 31
    Name: Lucas Turner-- Age: 38

The binary search tree of the economy plus class is:
            Name: Ella Turner-- Age: 29
        Name: Olivia Turner-- Age: 45
                Name: Chloe Moore-- Age: 80
            Name: Sophie Turner-- Age: 29
                Name: Emma Turner-- Age: 41
    Name: Daniel Turner-- Age: 47
            Name: Aiden Turner-- Age: 39
        Name: Oscar Moore-- Age: 28
                Name: Lily Moore-- Age: 35
                        Name: Lucas King-- Age: 43
                    Name: Aiden Cooper-- Age: 38
                        Name: Emma King-- Age: 38
                            Name: Olivia Turner--
Age: 92
            Name: Aria Cooper-- Age: 39
                Name: Ava Moore-- Age: 32
Name: Hannah King-- Age: 34
        Name: Chloe Moore-- Age: 36
            Name: Aiden Mitchell-- Age: 44
    Name: Aiden Turner-- Age: 33
                Name: Carter King-- Age: 37
            Name: Ethan Moore-- Age: 33
                        Name: Chloe Moore-- Age: 38
                    Name: Liam Mitchell-- Age: 80
                Name: Grace Cooper-- Age: 43
        Name: Carter King-- Age: 42
                Name: Matthew King-- Age: 40
            Name: Ava King-- Age: 41
                Name: Matthew King-- Age: 41
The old passengers in the economy plus section:
Name: Chloe Moore
Age: 80
Registered Date: 19/2/2023
Right to Travel: true
Security Checked?: true
Luggage Weight: 17.3
Is he/she disabled? false
Business Class: false
Minimum Pay: 725
Did he/she pay extra?: true
Military: false
```

```
Name: Olivia Turner
Age: 92
Registered Date: 8/6/2023
Right to Travel: true
Security Checked?: true
Luggage Weight: 16.8
Is he/she disabled? false
Business Class: false
Minimum Pay: 720
Did he/she pay extra?: true
Military: false

Name: Liam Mitchell
Age: 80
Registered Date: 23/10/2023
Right to Travel: true
Security Checked?: true
Luggage Weight: 17.1
Is he/she disabled? false
Business Class: false
Minimum Pay: 710
Did he/she pay extra?: true
Military: false

What is the height of the economy class as binary
search tree?
6
The staff wants to double check if a passenger is on
the airplane(economy section)?
true
```

Conclusion:

To sum up, doing such projects is to ease the way of organizing data instead of using paper, employees that do calculation on their own. Writing a program and using different data structures and choosing the best data structure with least time complexity for each kind of problem is the best way to reach out our goals to save time and money.

References:

Data Structures course(BAU)