|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST DOCUMENT** | | | | |
| **Application Name:** Budget Airline | | | **Tester Name:** Juan Ignacio Maurente | |
| **Case ID** | **Test Case** | **Test Data** | **Expected Outcome** | **Actual Outcome** |
| Airline001 | Display Countries from array | Input USA | Print selection “USA” | Pass |
| Airline002 | Display Countries from array | Input all countries and wrong selection | Print country name selected and message of wrong | Pass |
| Airline003 | Display Cities for each country from array | Input cities and wrong number | Print city name selected and add to total and message of wrong | Pass |
| Airline004 | Display Cities for Domestic flights | Input domestic Cities | Print city selected and a message of wrong | Pass |
| Airline005 | Car selection display option | Select car option | Print car has been selected and add to Total | Pass |
| Airline006 | Domestic flights – No CAR | Select two cities without car | Print Cities selected and no car added | Pass |
| Airline007 | Domestic flights – With CAR | Select City twice. One with car and another one without car | Print City selected twice + Car (once) | Pass |
| Airline008 | Testing all International flights | Select all international destination possible | Print all international destination possible | Pass |

Pseudocode

START

Num destinationChoice

Num countrySelect

Num citySelect

ArrayList<String> Order = new ArrayList<String>();

IntDest.getInternationalDestinations(); initialize International destinations within an Domestic.getDomesticDestinations();Construction of our Domestic Destinations within an Array

do {

print 1. International Destinations

print 2. Domestic Destinations

print 3. Exit

intput destination choice : - “Select one option:”

switch (destinationChoice) {

case 1: INTERNATIONAL DESTINATIONS

for (int i = 0; i < (IntDest.countryArray.length); i++) {

print ((i+1)+ IntDest.countryArray[i]);

input countrySelect : - “Select one option: “

// Display of international flights and Choose cities

if (countrySelect>0 && countrySelect<=IntDest.countryArray.length)

{

String countrySelected = IntDest.countryArray[countrySelect-1];

Print countrySelected

for (int i = 0; i < intDestArray.length ; i++)

{ if (countrySelected == intDestArray[i].getcountryName()) {

Print ((i+1) + intDestArray[i].getcityName() + intDestArray[i].getcityCode()+ intDestArray[i].getflightPrice());

} else ; (nothing)

Input citySelect ; "Select the City”

Total += addInternationalFlightOrder(citySelect, intDestArray, Order);

break;

case 2: SELECTION OF DOMESTIC DESTINATIONS

Print Domestic Destinations

for (int i = 0; i < 5; i++) {

print ((i + 1 ) + domDestArray[i].getcityName()+ domDestArray[i].getflightPrice());

Input citySelect ; "Select the City”

if (citySelect == 1)

Print “Melbourne has been Selected”

Total += domDestArray[0].getflightPrice()

Total += getcarSelection(Total, domDestArray, Order, citySelect);

else if (citySelect == 2)

Print “Brisbane has been Selected

Total += domDestArray[1].getflightPrice()

Total += getcarSelection(Total, domDestArray, Order, citySelect);

else if (citySelect == 3)

Print “Cairns has been Selected”

Total += domDestArray[2].getflightPrice()

Total += getcarSelection(Total, domDestArray, Order, citySelect);

else if (citySelect == 4) {

Print “Perth has been Selected

Total += domDestArray[3].getflightPrice()

Total += getcarSelection(Total, domDestArray, Order, citySelect);

else if (citySelect == 5) {

Print “Hobart has been Selected

Total += domDestArray[4].getflightPrice()

Total += getcarSelection(Total, domDestArray, Order, citySelect);

else {

Print “Wrong Selection”

break;

case 3: {

Print “Preparing Invoice”

for (String i : Order)

Output (i);

Print Total Due Amount: + Total

break;

default:

Print “Wrong Choice”

break;

} while (destinationChoice != 3) ; To close the loop

public static int addInternationalFlightOrder(int citySelect, IntDest[] intDestArray, ArrayList<String> order)

if (citySelect > 0 && citySelect <= intDestArray.length) {

Print: intDestArray[citySelect - 1].getcityName() + AS BEEN SELECTED order.add(intDestArray[citySelect - 1].toString());

return intDestArray[citySelect - 1].getflightPrice();

} else print : Wrong Selection

return 0; }

private static int getcarSelection(int carSelection, Domestic[] domDestArray, ArrayList<String> order, int citySelect) {

Print would you like to rent a car?

Input 0 for yes or 1 for no

if (carSelection == 0) {

Print : “Your car has been added”

order.add(domDestArray[citySelect - 1].toStringWithCAR());

return Domestic.getcarHirePrice();

} else if (carSelection == 1) {

Print: "You did not hire any car"

order.add(domDestArray[citySelect - 1].toStringWithoutCAR());

} else Print: “Wrong Selection”

return 0;

STOP

START CLASS

public class Destination

Declarations

private String cityName;

private String cityCode;

private int flightPrice;

public Destination (String cityName, String cityCode, int flightPrice) {

this.cityName = cityName;

this.cityCode = cityCode;

this.flightPrice = flightPrice;

}

public String getcityName() {

return cityName; }

public String getcityCode() {

return cityCode; }

public int getflightPrice() {

return flightPrice; }

public String toString() {

return cityName + cityCode + flightPrice;

ENDCLASS

STARTCLASS

public class IntDest extends Destination {

Declarations

private String countryName;

public static String countryArray[] = { "USA", "SPAIN", "CANADA", "FRANCE", "GREAT BRITAIN" };

public IntDest(String countryName, String cityName, String cityCode, int flightPrice) {

super(cityName, cityCode, flightPrice);

this.countryName = countryName; }

public static IntDest[] getInternationalDestinations() {

IntDest [] intDestArray = new IntDest[10];

intDestArray[0]= new IntDest (countryArray[0], "NEW YORK CITY", "NYC", 1257);

intDestArray[1]= new IntDest (countryArray[0], "LOS ANGELES","LAX",960);

intDestArray[2]= new IntDest (countryArray[1], "MADRID","MAD",1510);

intDestArray[3]= new IntDest (countryArray[1], "BARCELONA","BCN",1504);

intDestArray[4]= new IntDest (countryArray[2], "TORONTO","YTO",1752);

intDestArray[5]= new IntDest (countryArray[2], "VANCOUVER","YVR",1737);

intDestArray[6]= new IntDest (countryArray[3], "PARIS","PAR",1521);

intDestArray[7]= new IntDest (countryArray[3], "MARSEILLE","MRS",2044);

intDestArray[8]= new IntDest (countryArray[4], "LONDON","LON",1493);

intDestArray[9]= new IntDest (countryArray[4], "MANCHESTER","MAN",1573);

return intDestArray; }

public String getcountryName() {

return countryName; }

public String toString() {

return "Country: "+ countryName+ "City: "+ getcityName() + getcityCode() + "Price: $" + getflightPrice();

ENDCLASS

STARTCLASS

public class Domestic extends IntDest{

DECLARATIONS

private static String carSelection = "TOYOTA RAV4";

private static int carHirePrice = 100;

public Domestic(String cityName, String cityCode, int flightPrice) {

super("Australia", cityName, cityCode, flightPrice); }

public static Domestic[] getDomesticDestinations() {

Domestic[] domDestArray = new Domestic[5];

domDestArray[0] = new Domestic("MELBOURNE", "MEL", 330);

domDestArray[1] = new Domestic("BRISBANE", "BNE", 202);

domDestArray[2] = new Domestic("CAIRNS", "CNS", 283);

domDestArray[3] = new Domestic("PERTH", "PER", 451);

domDestArray[4] = new Domestic("HOBART", "HBA", 502);

return domDestArray; }

public static String getcarSelection() {

return carSelection; }

public static int getcarHirePrice() {

return carHirePrice;}

public String toStringWithCAR() {

return "City: "+ getcityName()+ getcityCode()+ "Price: $" + getflightPrice() + "Car Hired: "+ carSelection + "Price Car: $" + carHirePrice;

public String toStringWithoutCAR()

return "City: "+ getcityName()+getcityCode()+ "Price: $" + getflightPrice()