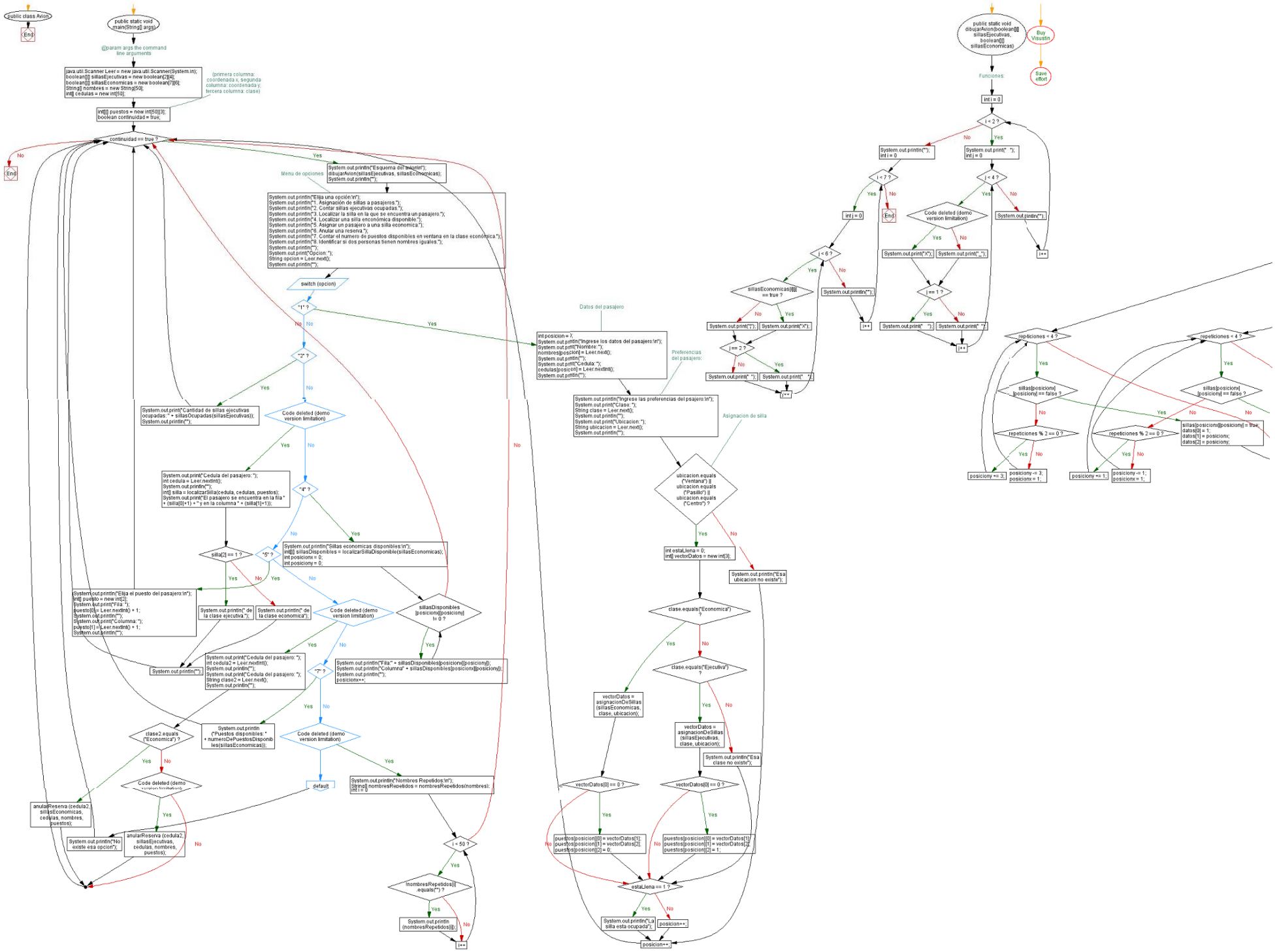
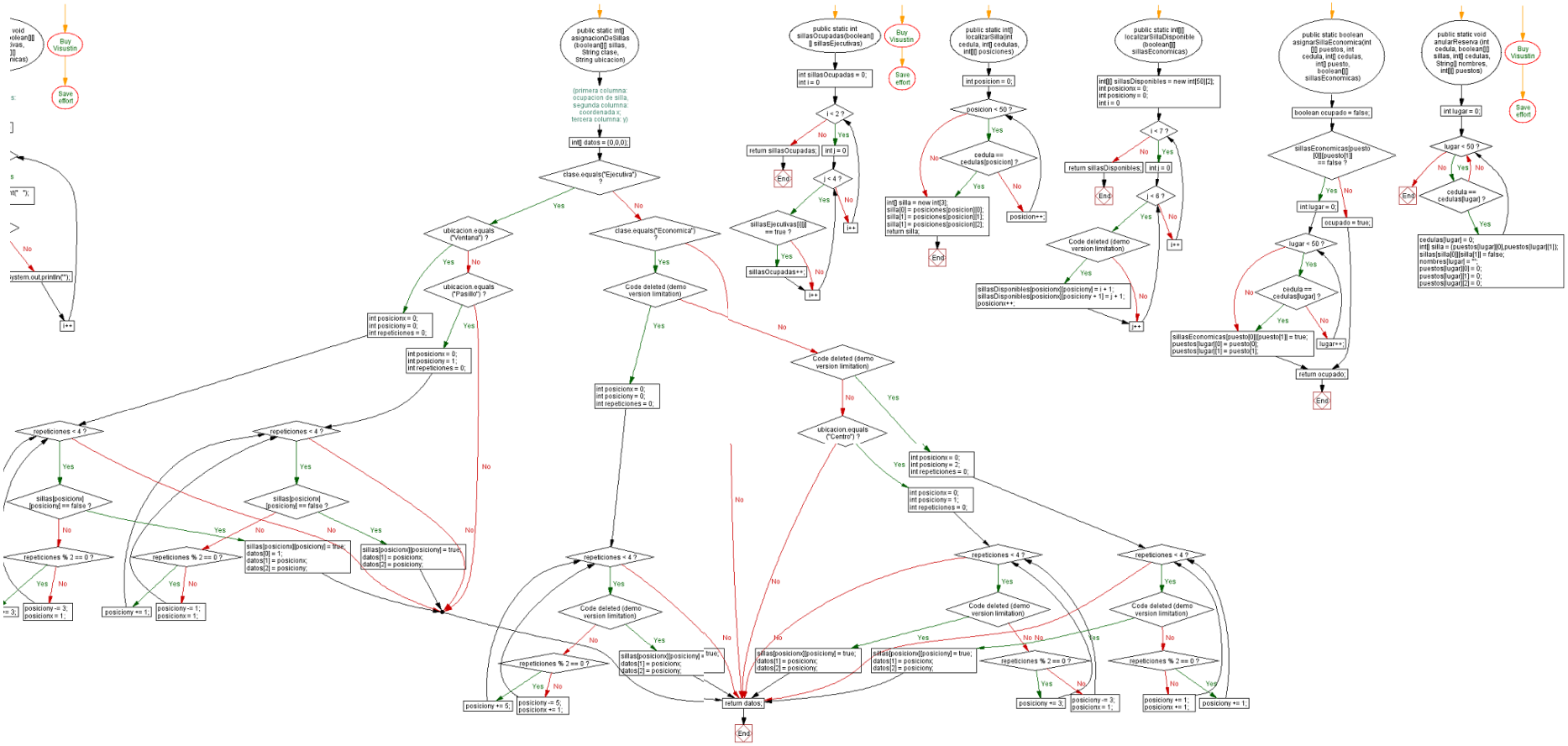
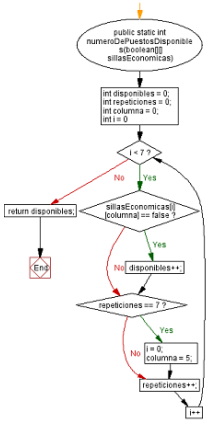
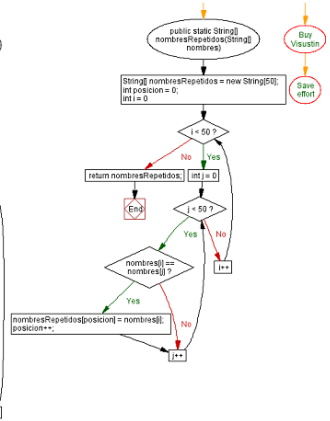

public class Avion
End
public static void main(String[] args)
@param args the command line arguments
java.util.Scanner Leer = new java.util.Scanner(System.in);
boolean[][] sillasEjecutivas = new boolean[2][4];
boolean[][] sillasEconomicas = new boolean[7][6];
String[] nombres = new String[50];
int[] cedulas = new int[50];
(primera columna: coordenada x, segunda columna: coordenada y, tercera columna: clase)
int[][] puestos = new int[50][3];
boolean continuidad = true;
continuidad == true ?
No
End
Yes
System.out.println("Esquema del avion");
dibujarAvion(sillasEjecutivas, sillasEconomicas);
System.out.println("");
Menu de opciones
System.out.println("Elija una opción:\n");
System.out.println("1. Asignación de sillas a pasajeros:");
System.out.println("2. Contar sillas ejecutivas ocupadas:");
System.out.println("3. Localizar la silla en la que se encuentra un pasajero:");
System.out.println("4. Localizar una silla económica disponible:");
System.out.println("5. Asignar un pasajero a una silla economica:");
System.out.println("6. Anular una reserva:");
System.out.println("7. Contar el numero de puestos disponibles en ventana en la clase económica:");
System.out.println("8. Identificar si dos personas tienen nombres iguales:");
System.out.println("");
System.out.print("Opcion: ");
String opcion = Leer.next();
System.out.println("");
switch (opcion)
"1" ?
Yes
No
Datos del pasajero
int posicion = 1;
System.out.println("Ingrese los datos del pasajero:\n");
System.out.print("Nombre: ");
nombresPosicion[] = Leer.next();
System.out.println("");
System.out.print("Cedula: ");
cedulas[posicion] = Leer.nextInt();
System.out.println("");
Preferencias del pasajero:
System.out.println("Ingrese las preferencias del psajero:\n");
System.out.print("Clase: ");
String clase = Leer.next();
System.out.println("");
System.out.print("Ubicacion: ");
String ubicacion = Leer.next();
System.out.println("");
Asignacion de silla
ubicacion.equals ("Ventana") || ubicacion.equals ("Pasillo") || ubicacion.equals ("Centro") ?
Yes
No
int estaLlena = 0;
int[] vectorDatos = new int[3];
System.out.println("Esa ubicacion no existe");
clase.equals("Economica") ?
Yes
No
clase.equals("Ejecutiva") ?
vectorDatos = asignacionDeSillas (sillasEconomicas, clase, ubicacion);
vectorDatos = asignacionDeSillas (sillasEjecutivas, clase, ubicacion);
System.out.println("Esa clase no existe");
vectorDatos[0] == 0 ?
puestos[posicion][0] = vectorDatos[1];
puestos[posicion][1] = vectorDatos[2];
puestos[posicion][2] = 0;
puestos[posicion][0] = vectorDatos[1];
puestos[posicion][1] = vectorDatos[2];
puestos[posicion][2] = 1;
estaLlena == 1 ?
System.out.println("La silla esta ocupada");
posicion++;
"2" ?
System.out.print("Cantidad de sillas ejecutivas ocupadas: " + sillasOcupadas(sillasEjecutivas));
System.out.println("");
Code deleted (demo version limitation)
System.out.print("Cedula del pasajero: ");
int cedula = Leer.nextInt();
System.out.println("");
int[] silla = localizarSilla(cedula, cedulas, puestos);
System.out.print("El pasajero se encuentra en la fila " + (silla[0]+1) + " y en la columna " + (silla[1]+1));
"4" ?
silla[2] == 1 ?
"5" ?
System.out.print(" de la clase ejecutiva.");
System.out.print(" de la clase economica");
System.out.print("Elija el puesto del pasajero:\n");
int[] puesto = new int[3];
System.out.print("Fila: ");
puesto[0] = Leer.nextInt() + 1;
System.out.print("");
System.out.print("Columna: ");
puesto[1] = Leer.nextInt() + 1;
System.out.print("");
System.out.print("");
System.out.print("Sillas economicas disponibles:\n");
int[][] sillasDisponibles = localizarSillaDisponible(sillasEconomicas);
int posicionx = 0;
int posicony = 0;
sillasDisponibles [posicion][posicion] != 0 ?
System.out.print("Fila:" + sillasDisponibles[posicion][posicion]);
System.out.print("Columna" + sillasDisponibles[posicion][posicion]);
System.out.print("");
posicion++;
"6" ?
System.out.print("Cedula del pasajero: ");
int cedula2 = Leer.nextInt();
System.out.print("");
System.out.print("Cedula del pasajero: ");
String clase2 = Leer.next();
System.out.print("");
clase2.equals ("Economica") ?
anularReserva (cedula2, sillasEconomicas, cedulas, nombres, puestos);
System.out.print("No existe esa opcion");
anularReserva (cedula2, sillasEjecutivas, cedulas, nombres, puestos);
"7" ?
System.out.print("Puestos disponibles: " + numeroDePuestosDisponibles(sillasEconomicas));
default
Code deleted (demo version limitation)
System.out.print("Nombres Repetidos:\n");
String[] nombresRepetidos = nombresRepetidos(nombres);
int i = 0
i < 50 ?
nombresRepetidos[i] .equals("") ?
System.out.print (nombresRepetidos[i]);
i++
public static void dibujarAvion(boolean[][] sillasEjecutivas, boolean[][] sillasEconomicas)
Buy Visustin
Save effort
Funciones:
int i = 0
i < 2 ?
System.out.print("");
int j = 0
i < 7 ?
i < 4 ?
Code deleted (demo version limitation)
System.out.print("*");
System.out.print("_");
System.out.print(" ");
System.out.print(" ");
j == 1 ?
i < 6 ?
sillasEconomicas[i][j] == true ?
System.out.print("[");
System.out.print("x");
j == 2 ?
System.out.print("]");
repeticiones < 4 ?
sillas[posicion][posicion] == false ?
repeticiones % 2 == 0 ?
posicion += 3;
posicion -= 3;
posicion += 1;
posicion -= 1;
sillas[posicion][posicion] = true;
datos[0] = 1;
datos[1] = posicion;
datos[2] = posicion;


Buy Visustin
Save effort
public static int[] asignacionDeSillas (boolean[][] sillas, String clase, String ubicacion)
(primera columna: ocupacion de silla, segunda columna: coordenada x; tercera columna: y)
int[] datos = {0,0,0};
clase.equals("Ejecutiva") ?
ubicacion.equals ("Ventana") ?
ubicacion.equals ("Pasillo") ?
clase.equals("Economica") ?
Code deleted (demo version limitation)
int posicionx = 0; int posiciony = 0; int repeticiones = 0;
int posicionx = 0; int posiciony = 1; int repeticiones = 0;
repeticiones < 4 ?
sillas[posicionx][posiciony] == false ?
repeticiones % 2 == 0 ?
sillas[posicionx][posiciony] = true; datos[0] = 1; datos[1] = posicionx; datos[2] = posiciony;
posiciony -= 3; posicionx = 1;
posiciony += 1;
posiciony -= 1; posicionx = 1;
posiciony += 5;
posiciony -= 5; posicionx += 1;
return datos;
End
public static int sillasOcupadas(boolean[][] sillasEjecutivas)
int sillasOcupadas = 0; int i = 0
i < 2 ?
return sillasOcupadas;
int j = 0
i < 4 ?
sillasEjecutivas[i][j] == true ?
sillasOcupadas++;
i++
public static int[] localizarSilla(int cedula, int[] cedulas, int[][] posiciones)
int posicion = 0;
posicion < 50 ?
cedula == cedulas[posicion] ?
int[] silla = new int[3]; silla[0] = posiciones[posicion][0]; silla[1] = posiciones[posicion][1]; silla[1] = posiciones[posicion][2]; return silla;
posicion++;
public static int[][] localizarSillaDisponible (boolean[][] sillasEconomicas)
int[][] sillasDisponibles = new int[50][2]; int posicionx = 0; int posiciony = 0; int i = 0
i < 7 ?
return sillasDisponibles;
j < 6 ?
Code deleted (demo version limitation)
sillasDisponibles[posicionx][posiciony] = i + 1; sillasDisponibles[posicionx][posiciony + 1] = j + 1; posicionx++;
public static boolean asignarSillaEconomica(int[][] puestos, int cedula, int[] cedulas, int[] puesto, boolean[][] sillasEconomicas)
boolean ocupado = false;
sillasEconomicas[puesto[0]][puesto[1]] == false ?
int lugar = 0;
ocupado = true;
lugar < 50 ?
cedula == cedulas[lugar] ?
sillasEconomicas[puesto[0]][puesto[1]] = true; puestos[lugar][0] = puesto[0]; puestos[lugar][1] = puesto[1];
lugar++;
return ocupado;
public static void anularReserva (int cedula, boolean[][] sillas, int[] cedulas, String[] nombres, int[][] puestos)
int lugar = 0;
cedulas[lugar] = 0; int[] silla = {puestos[lugar][0],puestos[lugar][1]}; sillas[silla[0]][silla[1]] = false; nombres[lugar] = ""; puestos[lugar][0] = 0; puestos[lugar][1] = 0; puestos[lugar][2] = 0;





Buy Visustin
Save effort