

Práctica 2

5 de septiembre del 2019

Estructura de Datos

Número de control – Nombre:

18131209 – ADAME SANDOVAL JOSE MISAEAL

18131222 – CARREON REYES JOSE LUIS

18131288 – VAZQUEZ AGUIRRE CARLOS ALBERTO

Descripción:

ESCRIBE UNA APLICACION JAVA NETBEANS VISUAL QUE LEA Y VISUALICE A 2 GRUPOS DE MASCOTAS. ALMACENA A CADA GRUPO EN UN ARREGLO ESTATICO DE UNA MASCOTA INTERESA: SU NOMBRE, SU TIPO (PERRO, OSO, GATO, ETCETERA), EL NUMERO DE VACUNAS QUE LE HAN PUESTO Y EL COSTO TOTAL DE ESAS VACUNAS. ADEMAS INTERESA SI ES MACHO O HEMBRA , Y SU EDAD.

1. AGREGA UN BOTON QUE INTERCAMBIE LA PRIMER MASCOTA CON LA ULTIMA DEL GRUPO SELECCIONADO. UTILIZA EL METODO `set()` PARA OBJETOS ARRAYLIST
2. AGREGA UN BOTON QUE ELIMINE A TODAS LAS MASCOTAS CUYO COSTO DE VACUNAS SEAN MAYORES A 100 PESOS, DE AMBOS GRUPOS. UTILIZA UN METODO EN LA APLICACION.
3. AGREGA UN BOTON QUE INTERCAMBIE A LAS ULTIMAS MASCOTAS DE AMBOS GRUPOS. AYUDA: AMBOS GRUPOS DEBEN TENER AL MENOS UNA MASCOTA.

80

Desarrollo:

Interfaz Gráfica

AÑADIR MASCOTA

Nombre

Tipo

Vacunas

Costo

Sexo ☒ Macho ☐ Hembra

Edad

Grupo

NO. MASCOTAS EN CADA GRUPO:

| Grupo 1 | Grupo 2 |
|---------|---------|
| 0 | 0 |

Grupo

| # | Nombre | Tipo | # Vac | \$ Vac | Sexo | Edad |
|---|--------|------|-------|--------|------|------|
|---|--------|------|-------|--------|------|------|

1a mascota --> última / mismo grupo

Costo Vacunas > 100 / ambos grupos

últ. masc <--> últ. masc. / ambos grupos

Definición de los objetos

```
public class JFrame extends javax.swing.JFrame {
```

```
    final int MAX = 30;
```

```
    private Mascota[] grupo1 = new Mascota[MAX];
```

```
    private Mascota[] grupo2 = new Mascota[MAX];
```

```
    private int[] noMascotasGpo = {0, 0};
```

```
    /**
```

```
     * Creates new form JFrame
```

```
     */
```

```
    public JFrame() {
```

```
        initComponents();
```

```
        for(int i = 0; i < MAX; i++){
```

```
            grupo1[i] = new Mascota();
```

```
            grupo2[i] = new Mascota();
```

```
        }
```

```
    }
```

OK

Clases

```
public class Mascota {  
    // Atributos  
    private String nombre;  
    private String tipo;  
    private int noVacunas;  
    private double costoVacunas;  
    private char sexo;  
    private int edad;  
  
    // Método Copia  
    public void Copia(Mascota masc){  
        nombre = masc.nombre;  
        tipo = masc.tipo;  
        noVacunas = masc.noVacunas;  
        costoVacunas = masc.costoVacunas;  
        sexo = masc.sexo;  
        edad = masc.edad;  
    }  
  
    // Getters  
    public String getNombre(){  
        return nombre;  
    }  
  
    public String getTipo(){  
        return tipo;  
    }  
}
```

```
public int getNoVacunas(){  
    return noVacunas;  
}
```

```
public double getCostoVacunas(){  
    return costoVacunas;  
}
```

```
public char getSexo(){  
    return sexo;  
}
```

```
public int getEdad(){  
    return edad;  
}
```

// Setters

```
public void setNombre(String nom){  
    nombre = nom;  
}
```

```
public void setTipo(String tip){  
    tipo = tip;  
}
```

```
public void setNoVacunas(int noV){  
    noVacunas = noV;  
}
```

```
public void setCostoVacunas(double cosV){  
    costoVacunas = cosV;  
}
```

```
public void setSexo(char sex){  
    sexo = sex;  
}
```

```
public void setEdad(int ed){  
    edad = ed;  
}
```

// Métodos

```
public void Leer(String nom, String tip, int noV, double cosV, char sex, int ed){  
    nombre = nom;  
    tipo = tip;  
    noVacunas = noV;  
    costoVacunas = cosV;  
    sexo = sex;  
    edad = ed;  
}
```

```
}
```

Métodos de la aplicación

```
private Mascota[] getGrupoSel() {  
    int noGrupoSel = jComboGrupo.getSelectedIndex();  
    switch(noGrupoSel){  
        case 0:  
            return grupo1;  
        case 1:  
            return grupo2;
```



```

        default:
            return grupo1;
    }
}

```

```

private void eliminarPorCosto(Mascota[] grupo, int col){
    if(noMascotasGpo[col] > 0){
        Mascota[] aux = new Mascota[noMascotasGpo[col]];
        for(int i = 0; i < noMascotasGpo[col]; i++)
            aux[i] = new Mascota();
        int noEc = 0;
        for(int i = 0; i < noMascotasGpo[col]; i++)
            if(grupo[i].getCostoVacunas() < 100)
                aux[noEc++] = Copia(grupo[i]);
        for(int i = 0; i < noEc; i++)
            grupo[i].Copia(aux[i]);
        noMascotasGpo[col] = noEc;
        jTableNoMasc.setValueAt(noMascotasGpo[col], 0, col);
    }
}

```

OK

```

private void jButtonAnadirActionPerformed(java.awt.event.ActionEvent evt) {
    Mascota[] grupoSel = getGrupoSel();
    int noGpoSel = jComboGrupo.getSelectedIndex();
    if(noMascotasGpo[noGpoSel] < MAX){
        grupoSel[noMascotasGpo[noGpoSel]++].Leer(jTextNombre.getText(),
            jTextTipo.getText(), Integer.parseInt(jTextNoVacunas.getText()),
            Double.parseDouble(jTextCosto.getText()), jRadioM.isSelected() ? 'M' : 'F',
            Integer.parseInt(jTextEdad.getText()));
        jTableNoMasc.setValueAt(noMascotasGpo[noGpoSel], 0,

```

```
jComboGrupo.getSelectedIndex());  
  
}  
  
}  
  
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
    jTabNoMasc.setValueAt(noMascotasGpo[0], 0, 0);  
    jTabNoMasc.setValueAt(noMascotasGpo[1], 0, 1);  
}  
  
private void jButtVisGpoActionPerformed(java.awt.event.ActionEvent evt) {  
    Mascota[] grupoSel = getGrupoSel();  
    jLabGrupoVis.setText("Grupo " + Integer.toString(jComboGrupo.getSelectedIndex() + 1));  
    DefaultTableModel modelo = (DefaultTableModel) jTabMascotas.getModel();  
    modelo.setRowCount(0);  
    int noGpoSel = jComboGrupo.getSelectedIndex();  
    for(int i = 0; i < noMascotasGpo[noGpoSel]; i++){  
        Object[] o = new Object[7];  
        o[0] = i + 1;  
        o[1] = grupoSel[i].getNombre();  
        o[2] = grupoSel[i].getTipo();  
        o[3] = grupoSel[i].getNoVacunas();  
        o[4] = grupoSel[i].getCostoVacunas();  
        o[5] = grupoSel[i].getSexo();  
        o[6] = grupoSel[i].getEdad();  
        modelo.addRow(o);  
    }  
}
```

if

```
private void jButtonIntercambiar1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Mascota[] grupoSel = getGrupoSel();  
    Mascota aux = grupoSel[0];  
    if(grupoSel == grupo1){  
        grupoSel[0] = grupoSel[noMascotasGpo[0] - 1];  
        grupoSel[noMascotasGpo[0] - 1] = aux;  
    } else if (grupoSel == grupo2){  
        grupoSel[0] = grupoSel[noMascotasGpo[1] - 1];  
        grupoSel[noMascotasGpo[1] - 1] = aux;  
    }  
}
```

glu referencia

if

```
private void jButtonIntercambiar3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Mascota aux = grupo1[noMascotasGpo[0] - 1];  
    grupo1[noMascotasGpo[0] - 1] = grupo2[noMascotasGpo[0] - 1];  
    grupo2[noMascotasGpo[0] - 1] = aux;  
}
```

```
private void jButtonEliminar2ActionPerformed(java.awt.event.ActionEvent evt) {  
    eliminarPorCosto(grupo1, 0);  
    eliminarPorCosto(grupo2, 1);  
}
```