Práctica 3

26 de septiembre del 2019

Estructura de Datos

Número de Control – Nombre:

18131209 – ADAME SANDOVAL JOSE MISAEL

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 SU RESPECTIVA PILA

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.

COMPETENCIA 7

1. ESCRIBE UN METODO EN LA CLASE PILA QUE RECIBA UNA PILA DE PARAMETRO, Y QUE INTERCAMBIE A LOS ELEMENTOS EN AMBAS PILAS (LA QUE LLAMA Y EL PARAMETRO) MENOS A LOS TOPES.

NOTA: NO PUEDES USAR ARREGLOS NI PILAS AUXILIARES EN EL METODO SOLICITADO.

COMPETENCIA 8

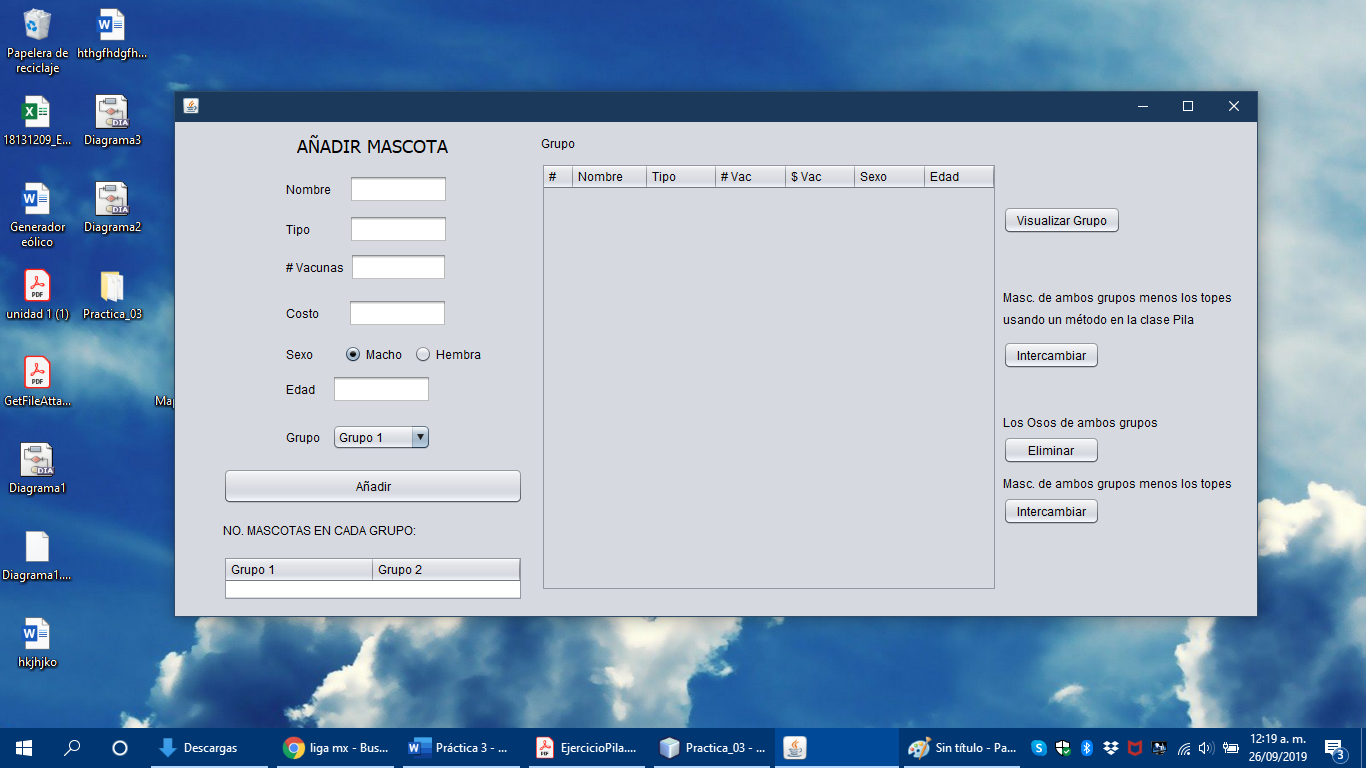
1. AGREGA UN BOTON QUE ELIMINE A TODOS LOS OSOS DE AMBOS GRUPOS DE MASCOTAS

2. AGREGA UN BOTON QUE INTERCAMBIE LOS ELEMENTOS DE AMBOS GRUPOS MENOS LOS QUE SE ENCUENTRAN EN EL TOPE.

NOTA: NO PUEDES USAR EL METODO ESCRITO EN LA COMPETENCIA 7.

**Desarrollo:**

Interfaz Gráfica



**Definición de los objetos:**

public class JFrame extends javax.swing.JFrame {

Pila grupo1 = new Pila();

Pila grupo2 = new Pila();

DefaultTableModel modelo;

public JFrame() {

initComponents();

modelo = (DefaultTableModel)jTabMascotas.getModel();

}

**Clases:**

**Mascota**

public class Mascota {

// Atributos

private String nombre;

private String tipo;

private int noVacunas;

private double costoVacunas;

private String sexo;

private int edad;

// Constructores

public Mascota() {

nombre = "NoId";

tipo = "NoId";

noVacunas = 0;

costoVacunas = 0;

sexo = "NoId";

edad = 0;

}

public Mascota(String nombre, String tipo, int noVacunas, double costoVacunas, String sexo, int

edad) {

this.nombre = nombre;

this.tipo = tipo;

this.noVacunas = noVacunas;

this.costoVacunas = costoVacunas;

this.sexo = sexo;

this.edad = edad;

}

// Métodos get/set

public String getNombre() {

return nombre;

}

public String getTipo() {

return tipo;

}

public int getNoVacunas() {

return noVacunas;

}

public double getCostoVacunas() {

return costoVacunas;

}

public String getSexo() {

return sexo;

}

public int getEdad() {

return edad;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setTipo(String tipo) {

this.tipo = tipo;

}

public void setNoVacunas(int noVacunas) {

this.noVacunas = noVacunas;

}

public void setCostoVacunas(double costoVacunas) {

this.costoVacunas = costoVacunas;

}

public void setSexo(String sexo) {

this.sexo = sexo;

}

public void setEdad(int edad) {

this.edad = edad;

}

}

**Pila**

public class Pila {

ArrayList<Mascota> elems;

int capacidad;

public Pila() {

elems = new ArrayList<>();

capacidad = 0;

}

public Pila(int capacidad) {

elems = new ArrayList<>();

this.capacidad = capacidad <= 0 ? 0 : capacidad;

}

public boolean Full() {

return capacidad > 0 && capacidad == elems.size();

}

public boolean Empty() {

return elems.isEmpty();

}

public void Push(Mascota elem) {

if (!Full())

elems.add(elem);

}

public Mascota Pop() {

if (! Empty()) {

Mascota tope = elems.get(elems.size()-1);

elems.remove(elems.size()-1);

return tope;

}

else

return new Mascota();

}

public int Length() {

return elems.size();

}

public Mascota Tope() {

return ! Empty() ? elems.get(elems.size()-1) : new Mascota();

}

public void Invertir() {

if(elems.size() > 1)

for(int i=0; i<elems.size()-1; i++)

elems.add(i,elems.remove(elems.size()-1));

}

// Método para intercambiar las mascotas de ambos grupos menos el tope

// COMPETENCIA 7

public void IntercambiarMenosTope(Pila p) {

final int tam1 = elems.size()-1;

final int tam2 = p.elems.size();

Mascota tope1 = elems.remove(elems.size()-1);

for(int i = 0; i < tam2; i++)

elems.add(p.elems.remove(0));

for(int i = 0; i < tam1; i++)

p.elems.add(elems.remove(0));

p.elems.add(elems.remove(elems.size()-1));

elems.add(tope1);

}

}

**Métodos de la aplicación:**

private Pila getGrupoSel() {

int noGrupoSel = jComboGrupo.getSelectedIndex();

switch(noGrupoSel){

case 0:

return grupo1;

case 1:

return grupo2;

default:

return grupo1;

}

}

private void actualizarNoMascotasGpos() {

jTabNoMasc.setValueAt(grupo1.Length(), 0, 0);

jTabNoMasc.setValueAt(grupo2.Length(), 0, 1);

}

private void eliminarOsos(Pila p) {

Pila aux = new Pila();

for(int i = p.Length(); i > 0; i--) {

Mascota m = p.Pop();

if(m.getTipo().equals("oso") || m.getTipo().equals("OSO")

|| m.getTipo().equals("Oso")) {

}

else {

aux.Push(m);

}

}

while(!aux.Empty()) {

p.Push(aux.Pop());

}

}

private void jButtAnadirActionPerformed(java.awt.event.ActionEvent evt) {

Pila grupoSel = getGrupoSel();

if(jTextNombre.getText().isEmpty() || jTextTipo.getText().isEmpty() ||

jTextNoVacunas.getText().isEmpty() || jTextCosto.getText().isEmpty()

|| jTextEdad.getText().isEmpty()) {

JOptionPane.showMessageDialog(this, "NO SE PUDO GUARDAR LA MASCOTA"

+ " , DEBE LLENAR TODOS LOS CAMPOS",

"ERROR", JOptionPane.ERROR\_MESSAGE);

} else if(! grupoSel.Full()) {

Mascota m = new Mascota(jTextNombre.getText(), jTextTipo.getText(),

Integer.parseInt(jTextNoVacunas.getText()),

Double.parseDouble(jTextCosto.getText()),

jRadioM.isSelected() ? "M" : "F",

Integer.parseInt(jTextEdad.getText()));

grupoSel.Push(m);

jTabNoMasc.setValueAt(grupoSel.Length(), 0, jComboGrupo.getSelectedIndex());

} else {

JOptionPane.showMessageDialog(this, "YA ESTA LLENO EL GRUPO",

"ERROR", JOptionPane.PLAIN\_MESSAGE);

}

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

jTabNoMasc.setValueAt(grupo1.Length(), 0, 0);

jTabNoMasc.setValueAt(grupo2.Length(), 0, 1);

}

private void jButtVisGpoActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Pila grupoSel = getGrupoSel();

Pila aux = new Pila(grupoSel.Length());

jLabGrupoVis.setText("Grupo " + Integer.toString(jComboGrupo.getSelectedIndex() + 1));

while (modelo.getRowCount() > 0)

modelo.removeRow(0);

while(! grupoSel.Empty()) {

Mascota m = grupoSel.Pop();

aux.Push(m);

Object[] o = new Object[6];

o[0] = m.getNombre();

o[1] = m.getTipo();

o[2] = m.getNoVacunas();

o[3] = m.getCostoVacunas();

o[4] = m.getSexo();

o[5] = m.getEdad();

modelo.addRow(o);

}

while(! aux.Empty())

grupoSel.Push(aux.Pop());

}

private void jButtIntercambiar1ActionPerformed(java.awt.event.ActionEvent evt) {

if(grupo1.Length() > 1 && grupo2.Length() > 1) {

grupo1.IntercambiarMenosTope(grupo2);

actualizarNoMascotasGpos();

}

}

private void jButtIntercambiar3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Mascota tope1 = grupo1.Pop();

Mascota tope2 = grupo2.Pop();

int tamGpo2 = grupo2.Length();

while(!grupo1.Empty())

grupo2.Push(grupo1.Pop());

grupo2.Invertir();

for(int j = tamGpo2; j>0; j--)

grupo1.Push(grupo2.Pop());

grupo1.Push(tope1);

grupo2.Push(tope2);

actualizarNoMascotasGpos();

}

private void jButtEliminar2ActionPerformed(java.awt.event.ActionEvent evt) {

if(grupo1.Length() > 0 && grupo2.Length() > 0) {

eliminarOsos(grupo1);

eliminarOsos(grupo2);

actualizarNoMascotasGpos();

jButtVisGpoActionPerformed(evt);

}

}