**IAnimal Interface**

namespace Ana\_AnimalLib

{

public interface IAnimal

{

//methods

void Voice();

bool IsCarnivore();

bool IsHerbivore();

bool IsOmnivore();

Image GetPicture();

}

}

**Animal Class**

namespace Ana\_AnimalLib

{

public abstract class Animal : IAnimal

{

//fields

private double \_weight; /\* lb \*/

private string \_imageFileName;

private string \_soundFile;

private string \_id;

//constructor

public Animal(double weight, string imageFileName, string soundFile, string id)

{

\_weight = weight;

\_imageFileName = imageFileName;

\_soundFile = soundFile;

\_id = id;

}

//properties

public double Weight { get => \_weight; }

public string ImageFileName { get => \_imageFileName; }

public string SoundFile { get => \_soundFile; }

public string ID { get => \_id; }

//methods

public abstract void Voice();

public abstract bool IsCarnivore();

public abstract bool IsHerbivore();

public abstract bool IsOmnivore();

public virtual Image GetPicture()

{

return Image.FromFile(\_imageFileName);

}

}

}

**Eagle Class**

namespace Ana\_AnimalLib

{

public class Eagle : Animal

{

//constructor

public Eagle(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override void Voice()

{

System.Media.SoundPlayer player = new System.Media.SoundPlayer(SoundFile);

player.Play();

}

public override bool IsCarnivore()

{

return true;

}

public override bool IsHerbivore()

{

return false;

}

public override bool IsOmnivore()

{

return false;

}

}

}

**Elephant Class**

namespace Ana\_AnimalLib

{

public class Elephant : Animal

{

//constructor

public Elephant(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override void Voice()

{

System.Media.SoundPlayer player = new System.Media.SoundPlayer(SoundFile);

player.Play();

}

public override bool IsCarnivore()

{

return false;

}

public override bool IsHerbivore()

{

return true;

}

public override bool IsOmnivore()

{

return false;

}

}

}

**Whale Class**

namespace Ana\_AnimalLib

{

public class Whale : Animal

{

//constructor

public Whale(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override void Voice()

{

System.Media.SoundPlayer player = new System.Media.SoundPlayer(SoundFile);

player.Play();

}

public override bool IsCarnivore()

{

return true;

}

public override bool IsHerbivore()

{

return false;

}

public override bool IsOmnivore()

{

return false;

}

}

}

**Cat Class**

namespace Ana\_AnimalLib

{

public abstract class Cat : Animal

{

//constructor

public Cat(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override abstract void Voice();

public override bool IsCarnivore()

{

return true;

}

public override bool IsHerbivore()

{

return false;

}

public override bool IsOmnivore()

{

return false;

}

public abstract override Image GetPicture();

}

}

**Lion Class**

namespace Ana\_AnimalLib

{

public class Lion : Cat

{

//constructor

public Lion(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override void Voice()

{

System.Media.SoundPlayer player = new System.Media.SoundPlayer(SoundFile);

player.Play();

}

public override Image GetPicture()

{

return Image.FromFile(ImageFileName);

}

}

}

**Tiger Class**

namespace Ana\_AnimalLib

{

public class Tiger : Cat

{

//constructor

public Tiger(double weight, string imageFileName, string soundFile, string id) :

base(weight, imageFileName, soundFile, id) { }

//methods

public override void Voice()

{

System.Media.SoundPlayer player = new System.Media.SoundPlayer(SoundFile);

player.Play();

}

public override Image GetPicture()

{

return Image.FromFile(ImageFileName);

}

}

}

**Form1.cs**

/\*Ana Carolina de Souza Mendes

\* Midterm - Spring 2018

\*

\* May 10, 2018

\* \*/

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Ana\_AnimalLib;

namespace Ana\_Midterm

{

public partial class Form1 : Form

{

//List to hold IAnimals

List<IAnimal> animals = new List<IAnimal>();

//string to hold objects names

string[] names = new string[10];

public Form1()

{

InitializeComponent();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FORM LOAD\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void Form1\_Load(object sender, EventArgs e)

{

Random rand = new Random();

//Instanciate Animal objects

Eagle vulture = new Eagle(15, "Animal\_Pictures/Vulture.jpg", "Animal\_Sounds/vulture.wav", rand.Next(1000, 10000).ToString());

Lion youngerLion = new Lion(300, "Animal\_Pictures/Lion.jpg", "Animal\_Sounds/hippo2.wav", rand.Next(1000, 10000).ToString());

Elephant zebra = new Elephant(1000, "Animal\_Pictures/Zebra.jpg", "Animal\_Sounds/zebra5.wav", rand.Next(1000, 10000).ToString());

Tiger tiger = new Tiger(670, "Animal\_Pictures/Tiger.jpg", "Animal\_Sounds/tiger.wav", rand.Next(1000, 10000).ToString());

Eagle canari = new Eagle(0.05d, "Animal\_Pictures/Canari.jpg", "Animal\_Sounds/canary.wav", rand.Next(1000, 10000).ToString());

Whale whale = new Whale(9000, "Animal\_Pictures/Whale.jpg", "Animal\_Sounds/whale.wav", rand.Next(1000, 10000).ToString());

Eagle mockingbird = new Eagle(0.1d, "Animal\_Pictures/Canari.jpg", "Animal\_Sounds/mockingbird.wav", rand.Next(1000, 10000).ToString());

Elephant elephant = new Elephant(12000, "Animal\_Pictures/Elephant.jpg", "Animal\_Sounds/elephant.wav", rand.Next(1000, 10000).ToString());

Lion lion = new Lion(420, "Animal\_Pictures/Lion.jpg", "Animal\_Sounds/dolphin3.wav", rand.Next(1000, 10000).ToString());

Tiger youngerTiger = new Tiger(550, "Animal\_Pictures/Tiger.jpg", "Animal\_Sounds/tiger.wav", rand.Next(1000, 10000).ToString());

Whale youngerWhale = new Whale(10000, "Animal\_Pictures/Whale.jpg", "Animal\_Sounds/whale.wav", rand.Next(1000, 10000).ToString());

//add to list

animals.Add(vulture);

animals.Add(youngerLion);

animals.Add(zebra);

animals.Add(tiger);

animals.Add(canari);

animals.Add(whale);

animals.Add(mockingbird);

animals.Add(elephant);

animals.Add(lion);

animals.Add(youngerTiger);

animals.Add(youngerWhale);

//put names in array

names = new string[11] { "vulture", "younger lion", "zebra", "tiger", "canari","whale","mockingbird","elephant","lion","younger tiger", "younger whale"};

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*METHODS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private Animal GetAnimalById(string id)

{

foreach (Animal animal in animals)

{

if (animal.ID == id)

return animal;

}

//if animal is not found, return null

return null;

}

private void DisplayAnimals(List<IAnimal> a)

{

//clear listview

listView1.Items.Clear();

//string to hold type of consumption for animal

string consumption;

//lvi to hold rows

ListViewItem lvi = new ListViewItem();

//index for names array

int i = 0;

foreach (Animal animal in a)

{

//conditions for consumption type

if (animal.IsCarnivore())

consumption = "Carnivore";

else if (animal.IsHerbivore())

consumption = "Herbivore";

else

consumption = "Omnivore";

//string to hold columns in row

string[] item = { animal.GetType().Name, names[i], animal.Weight.ToString("f2"), animal.ID, consumption, animal.SoundFile };

//crete row with string[] item

lvi = new ListViewItem(item);

//add to listview

listView1.Items.Add(lvi);

//control names index

i++;

if (i > names.Length)

i = 0;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EVENTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void btnDisplay\_Click(object sender, EventArgs e)

{

DisplayAnimals(animals);

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

if(listView1.SelectedItems.Count > 0 )

{

//get index from selecteditem in listview

int index = listView1.SelectedIndices[0];

//get animal from list using index

Animal selectedAnimal = (Animal)animals[index];

//get file path to display image in picturebox

pictureBox1.Image = selectedAnimal.GetPicture();

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

//call method to play sound

selectedAnimal.Voice();

}

}

}

}