# Animal App

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated

## MainWindow.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace AnimalApp

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Animal animal;

//Create a dictionary to hold animal objects

Dictionary<string, Animal> AnimalDict = new Dictionary<string, Animal>();

Dog dog;

Cat cat;

Bird bird;

public MainWindow()

{

InitializeComponent();

}

//Control for when the Cat button is checked by the user

private void RdBtnCat\_Checked(object sender, RoutedEventArgs e)

{

ImgAnimal.Source = new BitmapImage(new Uri("cat.jpg", UriKind.Relative));

animal = new Cat("Fur","Cat food: dry/wet");

}

//Control for when the Dog button is checked by the user

private void RdBtnDog\_Checked(object sender, RoutedEventArgs e)

{

ImgAnimal.Source = new BitmapImage(new Uri("dog.jpg", UriKind.Relative));

animal = new Dog("long hair, short hair, or floof","kibbles n bits" );

}

//Control for when the Bird button is checked by the user

private void RdBtnBird\_Checked(object sender, RoutedEventArgs e)

{

ImgAnimal.Source = new BitmapImage(new Uri("bird.jpg", UriKind.Relative));

animal = new Bird("feathers","seed and grain");

}

private void BtnSound\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.Sound();

}

//Controls for the six buttons.

//Control for the move button

private void BtnMove\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.Move();

}

//Control for the eat button

private void BtnEat\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.Eat();

}

//Control for the pet button

private void BtnPet\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.Pet();

}

//Control for the sound button

//Control for the food type button

private void BtnFoodType\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.SkinType;

}

//Control for the skin type button

private void BtnSkinType\_Click(object sender, RoutedEventArgs e)

{

LblResults.Content = animal.FoodType;

}

}

}

## Animal.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AnimalApp

{

abstract class Animal

{

private string \_skinType;

private string \_foodType;

//read/write access to skinType and foodType variables

public string SkinType

{

get { return \_skinType; }

set { \_skinType = value; }

}

public string FoodType

{

set { \_foodType = value; }

get { return \_foodType; }

}

//create default output methods

public string Eat()

{

return "Needs food to survive";

}

public string Move()

{

return "Can move around";

}

//abstract methods

public abstract string Pet();

public virtual string Sound()

{

return "Makes a typical animal noise";

}

//parameterized constructor for the animal class

public Animal (string paramFood, string paramSkin)

{

SkinType = paramSkin;

FoodType = paramFood;

}

}

}

## Dog.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AnimalApp

{

class Dog : Animal

{

//overwride the pet() and sound() methods from the animal class for dogs

public override string Pet()

{

return "Dogs love tummy rubs!";

}

public override string Sound()

{

return "Woof! Woof!";

}

//constructor for dog class that implements the base class constructor

public Dog(string paramFood, string paramSkin) : base(paramFood, paramSkin)

{

SkinType = paramSkin;

FoodType = paramFood;

}

}

}

## Cat.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AnimalApp

{

class Cat : Animal

{

//overwride the pet() and sound() methods from the animal class for cats

public override string Pet()

{

return "Pet with caution";

}

public override string Sound()

{

return "Meow! Give me food!";

}

//constructor for dog class that implements the base class constructor

public Cat(string paramFood, string paramSkin) : base(paramFood, paramSkin)

{

SkinType = paramSkin;

FoodType = paramFood;

}

}

}

## Bird.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AnimalApp

{

class Bird : Animal

{

//overwride the pet() and sound() methods from the animal class for birds

public override string Pet()

{

return "Only if you can catch it!";

}

public override string Sound()

{

return "Tweet Tweet";

}

//constructor for dog class that implements the base class constructor

public Bird(string paramFood, string paramSkin) : base(paramFood, paramSkin)

{

SkinType = paramSkin;

FoodType = paramFood;

}

}

}

## MainWindow.xaml

<Window x:Class="AnimalApp.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:AnimalApp"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" WindowStartupLocation="CenterScreen">

<Grid Background="LightBlue">

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="1.5\*"/>

<RowDefinition Height="0.5\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<RadioButton x:Name="RdBtnCat"

Content="Cat"

Grid.Column="1"

HorizontalAlignment="Left"

VerticalAlignment="Center"

FontSize="15"

FontWeight="DemiBold"

Checked="RdBtnCat\_Checked"/>

<RadioButton x:Name="RdBtnDog"

Content="Dog"

Grid.Column="1"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="15"

FontWeight="DemiBold"

Checked="RdBtnDog\_Checked"/>

<RadioButton x:Name="RdBtnBird"

Content="Bird"

Grid.Column="1"

HorizontalAlignment="Right"

VerticalAlignment="Center"

FontSize="15"

FontWeight="DemiBold"

Checked="RdBtnBird\_Checked"/>

<GroupBox Grid.Row="1"

Grid.Column="0"

Header="BaseClass"

FontSize="15"

FontWeight="DemiBold">

<StackPanel>

<Button x:Name="BtnMove"

Content="Move"

FontSize="15"

Margin="10"

Click="BtnMove\_Click"/>

<Button x:Name="BtnEat"

Content="Eat"

FontSize="15"

Margin="10"

Click="BtnEat\_Click"/>

</StackPanel>

</GroupBox>

<GroupBox Grid.Row="1"

Grid.Column="1"

Header="DerivedClass"

FontSize="15"

FontWeight="DemiBold">

<StackPanel>

<Button x:Name="BtnPet"

Content="Pet"

FontSize="15"

Margin="10"

Click="BtnPet\_Click"/>

<Button x:Name="Sound"

Content="Sound"

FontSize="15"

Margin="10"

Click="BtnSound\_Click"/>

</StackPanel>

</GroupBox>

<GroupBox Grid.Row="1"

Grid.Column="2"

Header="Properties"

FontSize="15"

FontWeight="DemiBold">

<StackPanel>

<Button x:Name="BtnFoodType"

Content="Food Type"

FontSize="15"

Margin="10"

Click="BtnFoodType\_Click"/>

<Button x:Name="BtnSkinType"

Content="Skin Type"

FontSize="15"

Margin="10"

Click="BtnSkinType\_Click"/>

</StackPanel>

</GroupBox>

<Image x:Name="ImgAnimal"

Grid.Column="1"

Grid.Row="2"

Source="cat.jpg"

Margin="10"/>

<Label x:Name="LblResults"

Grid.Row="3"

Grid.Column="0"

Grid.ColumnSpan="3"

Content="Results..."

VerticalAlignment="Center"

HorizontalAlignment="Center"

FontSize="15"

FontWeight="DemiBold"/>

</Grid>

</Window>