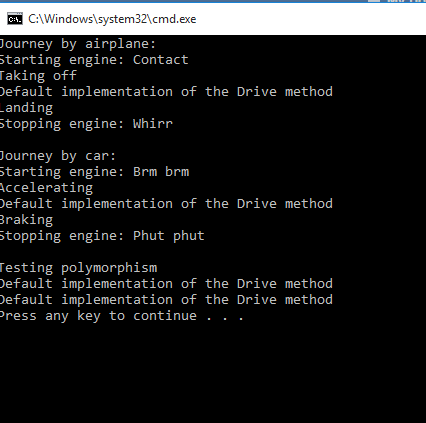
Student: Brian Johnston

Class: COP2362

ASssignment 3

**Vehicles:**

**Step 27**



**Program.cs:**

using System;

namespace Vehicles

{

class Program

{

static void doWork()

{

Console.WriteLine("Journey by airplane:");

Airplane myPlane = new Airplane();

myPlane.StartEngine("Contact");

myPlane.TakeOff();

myPlane.Drive();

myPlane.Land();

myPlane.StopEngine("Whirr");

Console.WriteLine("\nJourney by car:");

Car myCar = new Car();

myCar.StartEngine("Brm brm");

myCar.Accelerate();

myCar.Drive();

myCar.Brake();

myCar.StopEngine("Phut phut");

Console.WriteLine("\nTesting polymorphism");

Vehicle v = myCar;

v.Drive();

v = myPlane;

v.Drive();

}

static void Main()

{

try

{

doWork();

}

catch (Exception ex)

{

Console.WriteLine("Exception: {0}", ex.Message);

}

}

}

}

**Vehicle.Cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vehicles

{

class Vehicle

{

public void StartEngine(string noiseToMakeWhenStarting)

{

Console.WriteLine("Starting engine: {0}", noiseToMakeWhenStarting);

}

public void StopEngine(string noiseToMakeWhenStopping)

{

Console.WriteLine("Stopping engine: {0}", noiseToMakeWhenStopping);

}

public virtual void Drive()

{

Console.WriteLine("Default implementation of the Drive method");

}

}

}

**Airplane.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vehicles

{

class Airplane : Vehicle

{

public void TakeOff()

{

Console.WriteLine("Taking off");

}

public void Land()

{

Console.WriteLine("Landing");

}

}

}

**Car.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Vehicles

{

class Car : Vehicle

{

public void Accelerate()

{

Console.WriteLine("Accelerating");

}

public void Brake()

{

Console.WriteLine("Braking");

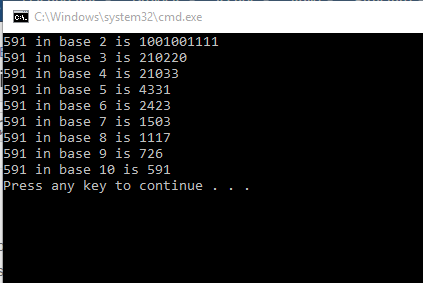
}

}

}

**ExpensionMethod**

**Screenshot**



**Program.CS**

using System;

using Extensions;

namespace ExtensionMethod

{

class Program

{

static void doWork()

{

int x = 591;

for (int i = 2; i <= 10; i++)

{

Console.WriteLine("{0} in base {1} is {2}",

x, i, x.ConvertToBase(i));

}

}

static void Main()

{

try

{

doWork();

}

catch (Exception ex)

{

Console.WriteLine("Exception: {0}", ex.Message);

}

}

}

}

**Util.CS**

using System;

namespace Extensions

{

static class Util

{

public static int ConvertToBase(this int i, int baseToConvertTo)

{

if (baseToConvertTo < 2 || baseToConvertTo > 10)

{

throw new ArgumentException("Value cannot be converted to base " + baseToConvertTo.ToString());

}

int result = 0;

int iterations = 0;

do

{

int nextDigit = i % baseToConvertTo;

i /= baseToConvertTo;

result += nextDigit \* (int)Math.Pow(10, iterations);

iterations++;

}

while (i != 0);

return result;

}

}

}