using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Cs\_Lesson10

{

#region Custom Class

// public class PaymentException : ApplicationException

// {

// public PaymentException(string errorMessage, decimal money, DateTime paymentDate)

// {

// ErrorMessage = errorMessage;

// Money = money;

// PaymentDate = paymentDate;

// }

// public string ErrorMessage { get; set; }

// public decimal Money { get; set; }

// public DateTime PaymentDate { get; set; }

// public override string Message => $@"

//Money : {Money}

//Payment Date : {PaymentDate.ToLongDateString()}/{PaymentDate.ToLongTimeString()}

//Error Message : {ErrorMessage}

//";

// }

// public class Program

// {

// static void Pay(decimal money)

// {

// if (money <= 0)

// {

// throw new PaymentException("You do not have enough money", money, DateTime.Now);

// }

// else

// {

// Console.WriteLine("Payment done");

// }

// }

// static void Main(string[] args)

// {

// try

// {

// Pay(-1000);

// }

// catch (Exception ex)

// {

// Console.WriteLine(ex.Message);

// }

// }

// }

#endregion

#region Simple Interface

//abstract class Human

//{

// public string Name { get; set; }

// public string Surname { get; set; }

// public DateTime BirthDate { get; set; }

// public override string ToString()

// {

// return $"Name : {Name } Surname : {Surname} Birth Date {BirthDate.ToLongDateString()}";

// }

//}

//abstract class Employee : Human

//{

// public string Position { get; set; }

// public decimal Salary { get; set; }

// public override string ToString()

// {

// return $"{base.ToString()} Position : {Position} Salary : {Salary}";

// }

//}

//public interface IManager

//{

// void Organize();

// void Control();

// void MakeBudget();

//}

//public interface IWorker

//{

// void Work();

// bool IsWorking { get; set; }

//}

//class Director : Employee, IManager

//{

// public void Control()

// {

// Console.WriteLine("I am boss, I will control you");

// }

// public void MakeBudget()

// {

// Console.WriteLine("Pul yoxdur, maash gelen ay olacaq");

// }

// public void Organize()

// {

// Console.WriteLine("I organise best work");

// }

//}

//class Seller : Employee, IWorker

//{

// public bool IsWorking { get; set; }

// public void Work()

// {

// Console.WriteLine("5 i 1 manat");

// }

//}

//class Cashier : Employee, IWorker

//{

// public bool IsWorking { get; set; }

// public void Work()

// {

// Console.WriteLine("o biri kassa bosdur");

// }

//}

//public class Program

//{

// static void Main(string a)

// {

// //IManager director = new Director()

// //{

// // BirthDate = DateTime.Now,

// // Name = "Mike",

// // Surname = "Mikeli",

// // Position = "Developer",

// // Salary = 1000

// //};

// //director.Control();

// //director.MakeBudget();

// List<IWorker> workers = new List<IWorker>();

// workers.Add(new Seller());

// workers.Add(new Cashier());

// foreach (var worker in workers)

// {

// worker.Work();

// Console.WriteLine(worker.IsWorking);

// if (worker is Seller s)

// {

// Console.WriteLine(s.Salary);

// }

// else if (worker is Cashier c)

// {

// Console.WriteLine(c.Salary);

// }

// }

// }

//}

#endregion

#region Task

abstract public class Car

{

public int MyId { get; set; } = ++staticCarId;

public static int staticCarId { get; set; }

public string Model { get; set; }

public string Vendor { get; set; }

public double Speed { get; set; }

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

";

}

public abstract int ToReachFirstStageSeconds();

public abstract int ToReachSecondStageSeconds();

}

public interface IRetro

{

void PlayRetroMusic();

}

public interface ISport

{

void UseTurbo();

void UserTurboPlus();

}

class Mercedes : Car

{

public override int ToReachFirstStageSeconds()

{

return 5;

}

public override int ToReachSecondStageSeconds()

{

return 12;

}

}

class RetroMercedes : Mercedes, IRetro

{

public void PlayRetroMusic()

{

Console.WriteLine("Retro music is being played in Retro Mercedes Car");

}

public override int ToReachFirstStageSeconds()

{

return 12;

}

public override int ToReachSecondStageSeconds()

{

return 17;

}

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

First Stage : {ToReachFirstStageSeconds()}

First Stage : {ToReachSecondStageSeconds()}

";

}

}

class SportMercedes : Car, ISport

{

public void UserTurboPlus()

{

Speed \*= 1.5;

Console.WriteLine("Turbo was used for Sport Mercedes");

}

public void UseTurbo()

{

Speed \*= 2;

Console.WriteLine("TurboPlus was used for Sport Mercedes");

}

public override int ToReachFirstStageSeconds()

{

return 3;

}

public override int ToReachSecondStageSeconds()

{

return 10;

}

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

First Stage : {ToReachFirstStageSeconds()}

First Stage : {ToReachSecondStageSeconds()}

";

}

}

class BMW : Car

{

public override int ToReachFirstStageSeconds()

{

return 6;

}

public override int ToReachSecondStageSeconds()

{

return 11;

}

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

First Stage : {ToReachFirstStageSeconds()}

First Stage : {ToReachSecondStageSeconds()}

";

}

}

class RetroBMW : BMW, IRetro

{

public void PlayRetroMusic()

{

Console.WriteLine("Retro music is being played in Retro BMW Car");

}

public override int ToReachFirstStageSeconds()

{

return 9;

}

public override int ToReachSecondStageSeconds()

{

return 16;

}

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

First Stage : {ToReachFirstStageSeconds()}

First Stage : {ToReachSecondStageSeconds()}

";

}

}

class SportBMW : BMW, ISport

{

public void UserTurboPlus()

{

Speed \*= 1.5;

Console.WriteLine("Turbo was used for Sport BMW");

}

public void UseTurbo()

{

Speed \*= 2;

Console.WriteLine("TurboPlus was used for Sport BMW");

}

public override int ToReachFirstStageSeconds()

{

return 4;

}

public override int ToReachSecondStageSeconds()

{

return 8;

}

public override string ToString()

{

return $@"

ID : {MyId}

Model : {Model}

Vendor : {Vendor}

Speed : {Speed}

First Stage : {ToReachFirstStageSeconds()}

First Stage : {ToReachSecondStageSeconds()}

";

}

}

public class Program

{

static void Main(string[] args)

{

Mercedes retroMercedes = new RetroMercedes()

{

Model = "E klass",

Speed = 200,

Vendor = "Mercedes"

};

Mercedes sportMercedes = new RetroMercedes()

{

Model = "A klass",

Speed = 270,

Vendor = "Mercedes"

};

BMW retroBMW = new RetroBMW()

{

Model = "X7",

Speed = 210,

Vendor = "BMW"

};

BMW sportBMW = new SportBMW()

{

Model = "X7",

Speed = 270,

Vendor = "BMW"

};

while (true)

{

Console.Clear();

Console.WriteLine(@"

MERCEDES 1

BMW 2");

string choice = Console.ReadLine();

if (choice == "1")

{

Console.Clear();

Console.WriteLine(@"

RETRO MERCEDES 1

SPORT MERCEDES 2");

choice = Console.ReadLine();

Console.Clear();

if (choice == "1")

{

Console.WriteLine(retroMercedes);

}

else if (choice == "2")

{

Console.WriteLine(sportMercedes);

}

else

{

Console.WriteLine("Incorrect Input!");

}

}

else if (choice == "2")

{

Console.Clear();

Console.WriteLine(@"

RETRO BMW 1

SPORT BMW 2");

choice = Console.ReadLine();

Console.Clear();

if (choice == "1")

{

Console.WriteLine(retroBMW);

}

else if (choice == "2")

{

Console.WriteLine(sportBMW);

}

else

{

Console.WriteLine("Incorrect Input!");

}

}

else

{

Console.WriteLine("Incorrect Input!");

}

Console.ReadKey();

Console.WriteLine("\n\n\nPress any key to continue . . .");

}

}

}

#endregion

}