

using System;

namespace Shape

{

abstract class Shape

{

public String name;

public Shape(String name)

{

this.name = name;

}

public abstract double CalculateArea();

}

class Circle : Shape

{

private double radius;

public Circle(string name, double radius) : base(name)

{

this.radius = radius;

}

public override double CalculateArea()

{

return 3.14 \* radius \* radius;

}

}

class Rectangle : Shape

{

private double width;

private double length;

public Rectangle(string name, double width, double length) : base(name)

{

this.width = width;

this.length = length;

}

public override double CalculateArea()

{

return width \* length;

}

}

class Program

{

static void Main(string[] args)

{

Circle circle = new Circle("Circle", 5);

Rectangle rectangle = new Rectangle("Rectangle", 10, 5);

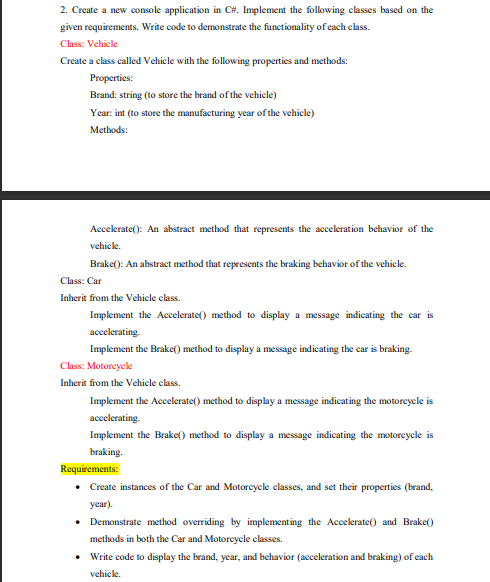
Console.WriteLine(circle.name +" " + circle.CalculateArea());

Console.WriteLine(rectangle.name + " " + rectangle.CalculateArea());

}

}

}



using System;

using System.Data;

namespace question2

{

abstract class Vehicle

{

protected Vehicle(string brand, int year)

{

Brand = brand;

Year = year;

}

public string Brand { set; get; }

public int Year { set; get; }

public abstract void Accelerate();

public abstract void Brake();

}

class car : Vehicle

{

public car(String brand, int year) : base(brand, year)

{

}

public override void Accelerate()

{

Console.WriteLine("The car is accelerating");

}

public override void Brake()

{

Console.WriteLine("The car is braking.");

}

}

class motorcycle : Vehicle

{

public motorcycle(String brand, int year) : base(brand, year) {

}

public override void Accelerate()

{

Console.WriteLine("The motorcycle is accelerating");

}

public override void Brake()

{

Console.WriteLine("The motor is braking.");

}

}

class Program

{

static void Main(string[] args)

{

car car = new car("tata", 2022);

Console.WriteLine("Car");

Console.WriteLine(car.Brand, car.Year);

car.Accelerate();

car.Brake();

Console.WriteLine();

motorcycle mt = new motorcycle("tvs", 2023);

Console.WriteLine("motorcycle");

Console.WriteLine(mt.Brand, mt.Year);

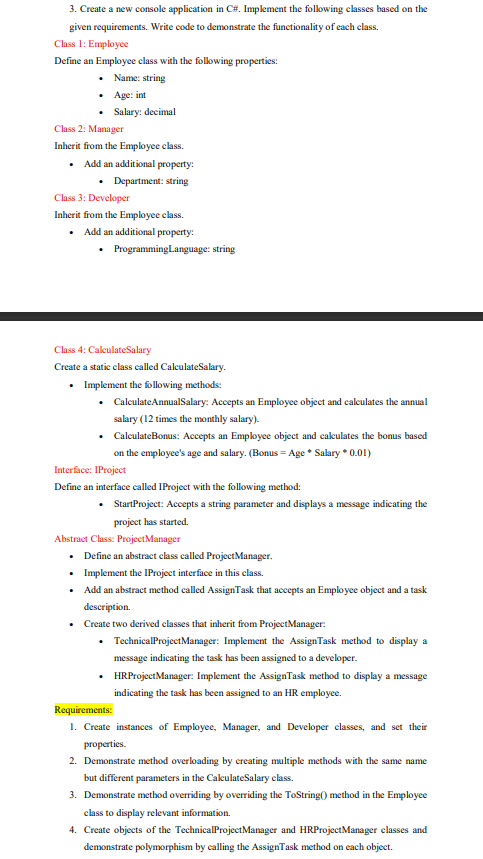
mt.Accelerate();

mt.Brake();

}

}

}



namespace Que3

{

public class Employee

{

public string name { get; set; }

public int age { get; set; }

public decimal salary { get; set; }

public Employee(string name, int age, decimal salary)

{

this.name = name;

this.age = age;

this.salary = salary;

}

public override string ToString()

{

return $"Employee name: {name}, age: {age}, salary: {salary}";

}

}

public class Manager : Employee

{

public string department { get; set; }

public Manager(string name, int age, decimal salary, string department) : base(name, age, salary)

{

this.department = department;

}

}

public class Developer : Employee

{

public string programmingLanguage { get; set; }

public Developer(string name, int age, decimal salary, string programmingLanguage) : base(name, age, salary)

{

this.programmingLanguage = programmingLanguage;

}

}

public static class CalculateSalary

{

public static decimal CalculateAnnualSalary(Employee employee)

{

return employee.salary \* 12;

}

public static decimal CalculateBonus(Employee employee)

{

return employee.age \* employee.salary \* 0.01m;

}

}

public interface IProject

{

void StartProject(string projectName)

{

Console.WriteLine("Project is started." + projectName);

}

}

public abstract class ProjectManager : IProject

{

public abstract void AssignTask(Employee employee, string taskDescription);

}

public class TechnicalProjectManager : ProjectManager

{

public override void AssignTask(Employee employee, string taskDescription)

{

Console.WriteLine("task assigned to developer: " + employee.name + " - " + taskDescription);

}

}

public class HRProjectManager : ProjectManager

{

public override void AssignTask(Employee employee, string taskDescription)

{

Console.WriteLine("task assigned to HR employee: " + employee.name + " - " + taskDescription);

}

}

internal class Program

{

static void Main(string[] args)

{

Employee employee = new Employee("Anand Wanve", 24, 10000m);

Manager manager = new Manager("Vaibhav Wanve", 23, 15000m, "Engineering");

Developer developer = new Developer("Shubhangi Wanve", 31, 17000m, "C#");

// method overloading

Console.WriteLine("Annual salary of employee: {0}", CalculateSalary.CalculateAnnualSalary(employee));

Console.WriteLine("Annual salary of manager: {0}", CalculateSalary.CalculateAnnualSalary(manager));

Console.WriteLine("Annual salary of developer: {0}", CalculateSalary.CalculateAnnualSalary(developer));

Console.WriteLine("Employee info: {0}", employee);

Console.WriteLine("Manager info: {0}", manager);

Console.WriteLine("Dev info: {0}", developer);

TechnicalProjectManager technicalProjectManager = new TechnicalProjectManager();

HRProjectManager hRProjectManager = new HRProjectManager();

technicalProjectManager.AssignTask(developer, "Add new feature");

hRProjectManager.AssignTask(employee, "conduct interviews");

}

}

}