Лабораторна робота №10

Варіант 4

Завдання 1

|  |  |
| --- | --- |
| 1. **Об’єкт** “Число” | |
| **поля** | * для зберігання натурального числа; |
| **методи** | * введення числа; * виведення числа; * знаходження кількості цифр; * знаходження суми цифр. |

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

try

{

Console.Write("Введiть число: ");

int number = Convert.ToInt32(Console.ReadLine());

Num n = new Num(number);

Console.WriteLine($"Число: {n.ToString()}");

Console.WriteLine($"Кiлькiть цифр: {n.CountOfNums()}");

Console.WriteLine($"Сума цифр: {n.SumOfDigits()}");

}

catch

{

Console.WriteLine("Повинно бути додатнє ціле число");

}

}

}

}

using System;

namespace task1

{

class Num

{

private int num;

public int Number

{

get { return num; }

set

{

if (value < 0)

throw new Exception("Повинно бути додатнє ціле число");

num = value;

}

}

public Num(int userNum)

{

Number = userNum;

}

public override string ToString()

{

return $"{num}";

}

public int CountOfNums()

{

return num.ToString().Length;

}

public int SumOfDigits()

{

int sum = 0;

while (num != 0)

{

sum += num % 10;

num /= 10;

}

return sum;

}

}

}

Завдання 2

|  |  |
| --- | --- |
| 1. **Об’єкт** “Автомобіль” (використати члени-класи) | |
| **поля** | * марка; * відомості про виробника (член-клас) (назва фірми, рік заснування, номер телефону, обсяги виробництва); * відомості про продавця (член-клас) (назва фірми, рік заснування, номер телефону, обсяги продажу); * відомості про власника (член-клас) (прізвище та ініціали, ідентифікаційний код) * колір; * номер; * дата випуску (член-клас); |
| **методи** | * визначення віку автомобіля; * зміни власника. |

class Program

{

static void Main(string[] args)

{

DateTime dateProductionManufacturer = new DateTime(1983, 1, 1);

DateTime dateProductionSeller = new DateTime(2000, 1, 1);

DateTime dateProductionCar = new DateTime(2018, 2, 1);

DateTime currDate = new DateTime(2020, 2, 4);

Manufacturer m = new Manufacturer("Kia Sorento Group", dateProductionManufacturer, 083456772, 40000);

Seller s = new Seller("SellingCars", dateProductionSeller, 123, 95);

Owner o = new Owner("Roman", 111);

Console.WriteLine("------- Iнформацiя про автомобiль -------");

Automobile auto = new Automobile("Kia", "white", 1111, m, s, o, dateProductionCar);

Console.WriteLine(auto);

Console.WriteLine($"Вiк машини - {auto.GetAge(currDate)}");

}

}

}

class Manufacturer

{

private string name;

private DateTime foundation;

private int phoneNumber;

private int productionVolume;

public string Name

{

get { return name; }

set { name = value; }

}

public DateTime Foundation

{

get { return foundation; }

set { foundation = value; }

}

public int PhoneNumber

{

get { return phoneNumber; }

set { phoneNumber = value; }

}

public int ProdutionVolume

{

get { return productionVolume; }

set { productionVolume = value; }

}

public Manufacturer(string name, DateTime foundation, int phoneNumber, int productionVolume)

{

Name = name;

Foundation = foundation;

PhoneNumber = phoneNumber;

ProdutionVolume = productionVolume;

}

public override string ToString()

{

return $"Iм'я: {name}\r\n Foundation: {foundation}\r\n" +

$"Номер телефону: {phoneNumber}\r\n" +

$"Обсяги продажу: {productionVolume}";

}

}

class Seller

{

private string name;

private DateTime foundation;

private int phoneNumber;

private int productionVolume;

public string Name

{

get { return name; }

set { name = value; }

}

public DateTime Foundation

{

get { return foundation; }

set { foundation = value; }

}

public int PhoneNumber

{

get { return phoneNumber; }

set { phoneNumber = value; }

}

public int ProdutionVolume

{

get { return productionVolume; }

set { productionVolume = value; }

}

public Seller(string name, DateTime foundation, int phoneNumber, int productionVolume)

{

Name = name;

Foundation = foundation;

PhoneNumber = phoneNumber;

ProdutionVolume = productionVolume;

}

public override string ToString()

{

return $"{name}-{foundation}-{phoneNumber}-{productionVolume}";

}

class Owner

{

private string name;

private int id;

public string Name

{

get { return name; }

set { name = value; }

}

public int Id

{

get { return id; }

set { id = value; }

}

public Owner(string name, int id)

{

Name = name;

Id = id;

}

public override string ToString()

{

return $"{name} - {id}";

}

}

}

class Automobile

{

private string brand;

private string color;

private int carNum;

private DateTime year;

private Manufacturer manufacturer;

private Seller seller;

private Owner owner;

public string Brand

{

get { return brand; }

set { brand = value; }

}

public string Color

{

get { return color; }

set { color = value; }

}

public int CarNum

{

get { return carNum; }

set { carNum = value; }

}

public Manufacturer Manufacturer

{

get { return manufacturer; }

set { manufacturer = value; }

}

public Seller Seller

{

get { return seller; }

set { seller = value; }

}

public Owner Owner

{

get { return owner; }

set { owner = value; }

}

public DateTime Year

{

get { return year; }

set { year = value; }

}

public Automobile(string brand, string color, int carNum,

Manufacturer manufacturer, Seller seller, Owner owner, DateTime year)

{

Brand = brand;

Color = color;

CarNum = carNum;

Manufacturer = manufacturer;

Seller = seller;

Owner = owner;

Year = year;

}

public override string ToString()

{

return $"Бренд: {brand} \r\nКолiр: {color} \r\nНомер машини: {carNum} " +

$"\r\nВиробник: {manufacturer.Name} " +

$" \r\nПродавець: {seller.Name} \r\nВласник: {owner.Name} " +

$"\r\n Роки виробництва: {year.ToString("MMMM yyyy")}";

}

public int GetAge(DateTime currentDate)

{

return currentDate.Year - year.Year;

}