**2 Классы**

Задание №1. Создать класс А с целочисленными полями а и b и двумя методами согласно варианту. Внутри класса реализовать конструктор для инициализации a и b. Создать объект класса и продемонстрировать работу со всеми элементами класса.

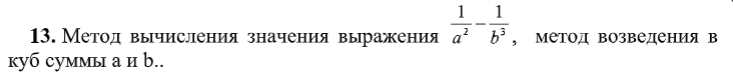


Рисунок 1.1 – Задание №1

Источник: задание 1

Листинг программы:

class A

{

public int a;

public int b;

public A (int a, int b)

{

this.a = a;

this.b = b;

}

public double GetArithmeticMean() => 1 / Math.Pow(a, 2);

public double CubSumOfNumbers() => Math.Pow((a+b),3);

}

class Program

{

static void Main()

{

Console.Write("Enter a: ");

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

Console.Write("Enter b: ");

int numB = Convert.ToInt32(Console.ReadLine())

A a = new A(numA, numB);

Console.WriteLine($"Computing the value of the expression = {a.GetArithmeticMean()}");

Console.WriteLine($"Erection in the cube of the sum of a and b = {a.CubSumOfNumbers()}");

}

}

Таблица 1.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3  4 | 0.1111111111111111  343 |

Анализ результатов:

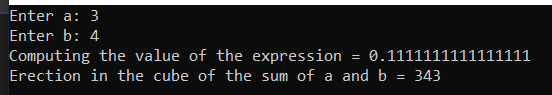


Рисунок 1.2 – Результат работы программы из задания №1

Источник: собственная разработка

Задание №2. Построить иерархию классов в соответствии с вариантом задания:

Листинг программы:

public class State

{

private string typeOfGoverment;

private string name;

private string ruler;

private int yearOfFoundation;

private string capital;

public string Name { get { return name; } set { name = value; } }

public string Ruler { get { return ruler; } set { ruler = value; } }

public decimal YearOfFoundation { get { return yearOfFoundation; } set { yearOfFoundation = (int)value; } }

public string Capital { get { return capital; } set { capital = value; } }

public string TypeOfGoverment { get { return typeOfGoverment; } set { typeOfGoverment = value; } }

protected State(string name, string ruler, int yearOfFoundation, string capital, string typeOfGoverment)

{

this. name = name;

this.ruler = ruler;

this.yearOfFoundation = yearOfFoundation;

this.capital = capital;

this.typeOfGoverment = typeOfGoverment;

}

public void giveInformation()

{

Console.WriteLine($"{name} is {typeOfGoverment}\nThe ruler is {ruler} \nYear of foundation: {yearOfFoundation} \nCapital: {capital}");

}

}

public class Republic : State

{

public Republic(string name, string ruler, int yearOfFoundation, string capital, string typeOfGoverment) : base(name, ruler, yearOfFoundation, capital, typeOfGoverment)

{

}

public void callReferendum()

{

Console.WriteLine("Referendum scheduled for February 27, 2022");

}

}

public class Monarchy : State

{

public Monarchy(string name, string ruler, int yearOfFoundation, string capital, string typeOfGoverment) : base(name, ruler, yearOfFoundation,capital,typeOfGoverment)

{

}

public void titleTransfer()

{

Console.WriteLine("In 2022, Elizabeth II transferred the title to Charles III");

}

}

public class Kingdom : State

{

public Kingdom(string name, string ruler, int yearOfFoundation, string capital, string typeOfGoverment) : base(name, ruler, yearOfFoundation,capital,typeOfGoverment)

{

}

public void titleTransfer()

{

Console.WriteLine("Margrethe II is reigning queen of Denmark from the Glücksburg dynasty since 14 January 1972, after the death of her father King Frederick IX");

}

}

class Program

{

static void Main()

{

Console.WriteLine("1.Great Britan \n2.Belarus \n3.Denmark ");

Console.Write("Enter number: ");

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

switch (num)

{

case 1:

Monarchy monarchy = new Monarchy("Great Britain", "Charles Philip Arthur George", 1922, "London", "Monarchy");

monarchy.giveInformation();

monarchy.titleTransfer();

break;

case 2:

Republic republic = new Republic("Belarus", "Alexander Lukashenko", 1991, "Minsk", "Republic");

republic.giveInformation();

republic.callReferendum();

break;

case 3:

Kingdom kingdom = new Kingdom("Denmark", "Margrethe Alexandrina Thorhildur Ingrid", 1523, "Copenhagen", "Kingdom");

kingdom.giveInformation();

kingdom.titleTransfer();

break;

}

}

}

Таблица 1.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 | Great Britain is Monarchy  The ruler is Charles Philip Arthur George  Year of foundation: 1922  Capital: London  In 2022, Elizabeth II transferred the title to Charles III |

Анализ результатов:

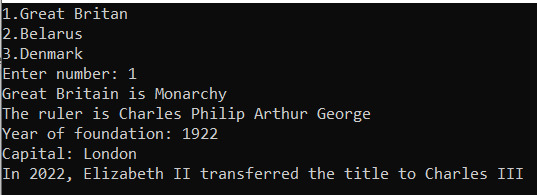


Рисунок 1.3 – Результат работы программы из задания №2

Источник: собственная разработка

Задание №3. Составить программу (при решении данных задач использовать оператор switch или вложенные операторы if). Во время сильного дождя на остановке стояло 12 человек. Подкативший автобус забрызгал грязью 4-го, 5-го, 6, 7 и 8-го человека. Остальные попрыгали в колючие кусты, причем 3-ий, 9-ый и 12-ый так и не смогли выбраться из них. Задать номер пассажира и определить грязный он или исцарапанный.

Листинг программы:

Console.Write("Enter passenger number from 1 to 12: ");

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

switch (number)

{

case (4):

Console.Write("The passenger is dirty");

break;

case (5):

Console.Write("The passenger is dirty");

break;

case (6):

Console.Write("The passenger is dirty");

break;

case (7):

Console.Write("The passenger is dirty");

break;

case (3):

Console.Write("The passenger is scratched");

break;

case (9):

Console.Write("The passenger is scratched");

break ;

case (12):

Console.Write("The passenger is scratched");

break;

default:

Console.Write("Passenger is fine");

break;

}

Таблица 1.3 – Входные и выходные данных задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 12 | The passenger is scratched |

Анализ результатов:



Рисунок 1.4 – Результат работы программы из задания №3

Источник: собственная разработка