**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. Разработать методы и свойства для каждого из определяемых классов.

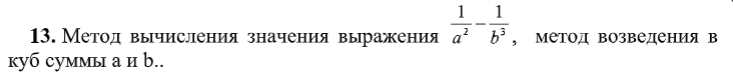


Рисунок 1.1 – Задание №1

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:

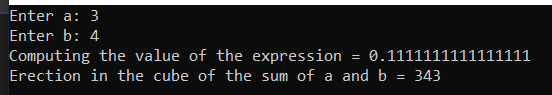


Рисунок 1.2 – Результат работы программы из задания №1

Источник: собственная разработка

Задание №2. Построить иерархию классов в соответствии с вариантом задания: Республика, монархия, королевство, государство.

Листинг программы:

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;

}

}

}

public class State

{

private string typeOfGoverment;

private string name;

private string ruler;

private int yearOfFoundation;

private string capital;

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}");

}

}

namespace z2

{

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");

}

}

}

namespace z2

{

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");

}

}

}

namespace z2

{

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");

}

}

}

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:

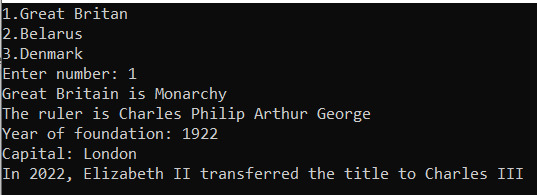


Рисунок 1.3 – Результат работы программы из задания №2

Источник: собственная разработка

Задание №3. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел. Предусмотреть методы сложения, вычитания, умножения и деления дробей. Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

using z3;

class Program

{

static void Main()

{

Console.WriteLine("Enter first Numerator ");

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

Console.WriteLine("Enter first Denominator ");

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

Console.WriteLine("Enter second Numerator ");

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

Console.WriteLine("Enter second Denominator ");

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

Fraction fraction = new Fraction(firstNumerator, firstDenominator, secondNumerator, secondDenominator);

fraction.Multiply();

fraction.Division();

fraction.Sum();

fraction.Subtraction();

}

}

namespace z3

{

class Fraction

{

public int firstNumerator;

public int firstDenominator;

public int secondNumerator;

public int secondDenominator;

public Fraction(int firstNumerator, int firstDenominator, int secondNumeration, int secondDenominator)

{

this.firstNumerator = firstNumerator; //Numerator - числитель

this.firstDenominator = firstDenominator; // Denominator - знаменатель

this.secondNumerator = secondNumeration;

this.secondDenominator = secondDenominator;

}

public void Multiply()

{

int numenator = (firstNumerator \* secondNumerator);

int denominator = (firstDenominator \* secondDenominator);

Console.WriteLine($"Numltiply = {numenator} / {denominator}");

}

public void Division()

{

int numenator = (firstNumerator \* secondDenominator);

int denominator = (firstDenominator \* secondNumerator);

Console.WriteLine($"Division = {numenator} / {denominator}");

}

public void Sum()

{

int commonDenominator = CommonDenominator(firstDenominator, secondDenominator);

int firstFinalNumerator = (commonDenominator / firstDenominator) \* firstNumerator;

int secondFinalNumerator = (commonDenominator / secondDenominator) \* secondNumerator;

Console.WriteLine($"Sum = {firstFinalNumerator + secondFinalNumerator} / {commonDenominator}");

}

public void Subtraction()

{

int commonDenominator = CommonDenominator(firstDenominator, secondDenominator);

int firstFinalNumerator = (commonDenominator / firstDenominator) \* firstNumerator;

int secondFinalNumerator = (commonDenominator / secondDenominator) \* secondNumerator;

Console.WriteLine($"Subtraction = {firstFinalNumerator - secondFinalNumerator} / {commonDenominator}");

}

private int CommonDenominator(int firstDenominator, int secondDenominator)

{

for (int i = firstDenominator; i < secondDenominator \* firstDenominator; i += firstDenominator)

{

if (i % firstDenominator == 0 && i % secondDenominator == 0) return i;

}

return -1;

}

}

}

Таблица 1.3 – Входные и выходные данных задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1  2  3  4 | Numltiply = 3 / 8  Division = 4 / 6  Sum = 5 / 4  Subtraction = -1 / 4 |

Анализ результатов:

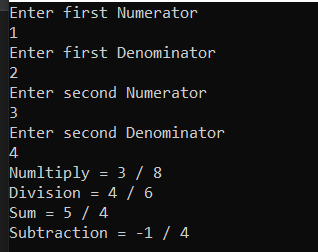


Рисунок 1.4 – Результат работы программы из задания №3

Источник: собственная разработка