**2 КЛАССЫ И ОБЪЕКТЫ. КОНСТРУКТОР И ДЕСТРУКТОР. НАСЛЕДОВАНИЕ**

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. Метод вычисления значения выражения (1/a^2) - (1/b^3), метод возведения в куб суммы a и b.

Листинг программы:

using System;

namespace task2

{

class A

{

int a;

int b;

public A(int A, int B)

{

a = A;

b = B;

}

public void expression()

{

double meaning = 1.0 / (Math.Pow(a, 2)) - 1.0 / (Math.Pow(b, 3));

Console.WriteLine(Math.Round(meaning, 3));

}

public void cube()

{

int sum = a + b;

Console.WriteLine(Math.Pow(sum, 3));

}

}

class Program

{

static void Main(string[] args)

{

A a = new A(10, 3);

a.expression();

a.cube();

}

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10, 3 | -0,027, 2197 |

Анализ результатов:



Рисунок 2.1 – Результат работы программы

Источник – собственная разработка

Задание 2. Построить иерархию классов в соответствии с вариантом задания: республика, монархия, королевство, государство.

Листинг программы:

using System;

namespace task3

{

public class State

{

public int cities;

public int square;

public string rulersName;

public string name;

public State(int cities, int square, string name)

{

this.cities = cities;

this.square = square;

this.name = name;

}

public void Print()

{

Console.WriteLine($"название {name}");

Console.WriteLine($"количество городов {cities}");

Console.WriteLine($"площадь {square} кв.км");

}

}

public class Republic : State

{

public string president;

public Republic(int cities, int square, string name, string president) : base(cities,square,name)

{

this.cities = cities;

this.square = square;

this.name = name;

this.president = president;

}

public new void Print()

{

Console.WriteLine($"имя президента {president}");

base.Print();

}

}

public class Monarchy : State

{

public string titleMonarch;

public string nameMonarch;

public Monarchy(int cities, int square, string name, string titleMonarch, string nameMonarch) : base(cities, square, name)

{

this.cities = cities;

this.square = square;

this.name = name;

this.titleMonarch = titleMonarch;

this.nameMonarch = nameMonarch;

}

public new void Print()

{

Console.WriteLine($"имя монарха {nameMonarch}");

Console.WriteLine($"титул монарха {titleMonarch}");

base.Print();

}

}

public class kingdom : State

{

public string queen;

public string king;

public kingdom(int cities, int square, string name, string queen, string king) : base(cities, square, name)

{

this.cities = cities;

this.square = square;

this.name = name;

this.queen = queen;

this.king = king;

}

public new void Print()

{

Console.WriteLine($"имя короля {king}");

Console.WriteLine($"имя королевы {queen}");

base.Print();

}

}

class Program

{

static void Main(string[] args)

{

State s1 = new State(12, 263, "Беларусь");

Console.WriteLine($"Государство");

s1.Print();

Republic s2 = new Republic(4, 108, "Россия", "Вовка");

Console.WriteLine($"\nРеспублика");

s2.Print();

Monarchy s3 = new Monarchy(22, 442, "США", "Король","Саймон");

Console.WriteLine($"\nМонархия");

s3.Print();

kingdom s4 = new kingdom(37, 681, "Чечня", "Елизавета", "Абдула");

Console.WriteLine($"\nКоролевство");

s4.Print();

}

}

}

Таблица 2.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Государство  название Беларусь  количество городов 12  площадь 263 кв.км  Республика  имя президента Вовка  название Россия  количество городов 4  площадь 108 кв.км  Монархия  имя монарха Саймон  титул монарха Король  название США  количество городов 22  площадь 442 кв.км  Королевство  имя короля Абдула  имя королевы Елизавета  название Чечня  количество городов 37  площадь 681 кв.км |

Анализ результатов:

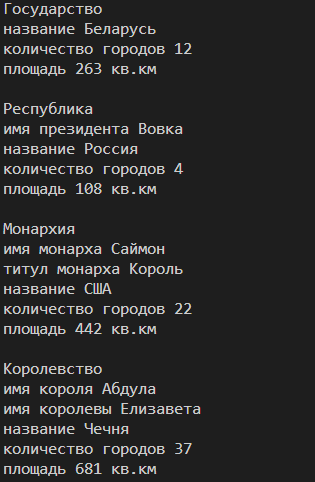


Рисунок 2.2 – Результат работы программы

Источник – собственная разработка