1. **Классы. Библиотеки FCL. Классы как типы и объекты этих типов**

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.

Листинг программы:

try

{

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

A clas = new A(a, b);

clas.FirstMethod(a, b);

clas.SecondMethod(a, b);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class A

{

public int first;

public int second;

public A(int first, int second)

{

this.first = first;

this.second = second;

}

public void FirstMethod(int a, int b)

{

Console.WriteLine($"Результат первого выражения: {1 / (double)a + 1 / Math.Sqrt(b)}");

}

public void SecondMethod(int a, int b)

{

Console.WriteLine($"Результат первого выражения: {Math.Pow(a, 6)}");

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 16 | Результат первого выражения = 0.75  Результат второго выражения = 64 |

Анализ результатов:

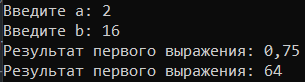


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Построить иерархию классов: двигатель, двигатель внутреннего сгорания, дизель, реактивный двигатель.

Листинг программы:

class Engine

{

public int weight;

public int enginePower;

public int fuelFlow;

public void Start()

{

}

public void Stop()

{

}

}

class ICE : Engine

{

public string combustionChamber;

public string fuelMixture;

public void PutOil()

{

}

public void FuelInjection()

{

}

}

class Diesel : ICE

{

public string typeOfPiston;

public void TurningTheCrankshaft()

{

}

public void AirIntake()

{

}

}

class JetEngine : ICE

{

public string typeOfNozzle;

public string typeOfJetEngine;

public int engineThrust;

public void CreateJetThrust()

{

}

}

Задание 3. Описать класс для работы со строкой, позволяющей хранить только двоичное число и выполнять с ним арифметические операции. Предусмотреть инициализацию с проверкой допустимости значений.

Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

try

{

Console.Write("Введите первое двоичое число: ");

string bitStr = Console.ReadLine();

Console.Write("Введите второе двоичное число: ");

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

bool right = false;

for (int i = 0; i < bitStr.Length; i++)

{

if (bitStr[i] == '1' || bitStr[i] == '0')

{

continue;

}

else

{

Console.WriteLine("Введена неверная строка");

right = true;

break;

}

}

if (!right)

{

BitString clas = new BitString(bitStr);

clas.Sum(b);

clas.Sub(b);

clas.Mul(b);

clas.Div(b);

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class BitString

{

public string bitStr;

public BitString(string bitStr)

{

this.bitStr = bitStr;

}

public static int ToDec(string bin)

{

var result = 0;

for (int i = 0; i < bin.Length; ++i)

{

if (bin[bin.Length - i - 1] == '1')

{

result += (int)Math.Pow(2, i);

}

}

return result;

}

public void Sum(int b)

{

int bits = ToDec(bitStr);

string second = Convert.ToString(b);

b = ToDec(second);

int summ = bits + b;

Console.WriteLine($"Результат сложения: {Convert.ToString(summ, 2)}");

}

public void Sub(int b)

{

int bits = ToDec(bitStr);

string second = Convert.ToString(b);

b = ToDec(second);

int subs = bits - b;

if (bits >= b)

Console.WriteLine($"Результат вычитания: {Convert.ToString(subs, 2)}");

else

Console.WriteLine($"Первое число должно быть меньше второго");

}

public void Mul(int b)

{

int bits = ToDec(bitStr);

string second = Convert.ToString(b);

b = ToDec(second);

int mult = bits \* b;

Console.WriteLine($"Результат умножения: {Convert.ToString(mult, 2)}");

}

public void Div(int b)

{

int bits = ToDec(bitStr);

string second = Convert.ToString(b);

b = ToDec(second);

int div = bits / b;

Console.WriteLine($"Результат деления: {Convert.ToString(div, 2)}");

}

}

Таблица 2.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10, 10 | Результат сложения = 100  Результат вычитания = 0  Результат умножения = 100  Результат деления = 1 |

Анализ результатов:

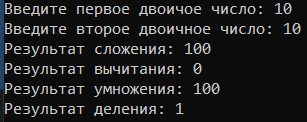


Рисунок 2.2 – Результат работы программы

Источник: собственная разработка