# Переменные и типы данных

Задание 1.

using System;

namespace Operations

{

class Program

{

private static int FirstEvent() { // Calculate value for the first Event

return (1000 - 60) + 90;

}

private static int SecondEvent()// Calculate value for the second Event

{

return (1000 - 60);

}

static void Main(string[] args)

{

Console.WriteLine($"В первом случае {FirstEvent()}\nВо втором случае {SecondEvent()}");

}

}

}



Задание 2.

using System;

namespace Operations

{

class Program

{

private static double CalculateLenghtOfRope(double Width, double Height)

{

double result = (Math.Sqrt((Width \* Width) + (Height \* Height))) + 3; // Calculate Length of the rope and add 3 additional metres

return result;

}

static void Main(string[] args)

{

Console.WriteLine($"Необходимая длина веревки: {CalculateLenghtOfRope(4,3)}");

}

}

}



Задание 3.

using System;

using System.Collections.Generic;

namespace Operations

{

class Program

{

private static List<double> ResultOfCalculaton = new List<double>();

private static double CalculateSquareOfRectangle(double width, double height)

{

double square = width \* height;

return square;

}

private static double CalculateSquareOfCircle(double radius)

{

double square = 2 \* Math.PI \* (radius \* radius);

return square;

}

private static void AddAllValues() {

ResultOfCalculaton.Add(CalculateSquareOfRectangle(7,9));

ResultOfCalculaton.Add(CalculateSquareOfRectangle(8, 9) - CalculateSquareOfRectangle(3, 2));

ResultOfCalculaton.Add(CalculateSquareOfCircle(4.5d));

}

static void Main(string[] args)

{

AddAllValues();

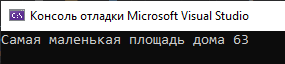
ResultOfCalculaton.Sort();

Console.WriteLine($"Самая маленькая площадь дома {ResultOfCalculaton[0]}");

}

}

}



Задание 4.

using System;

namespace Operations

{

class Program

{

#region Static Methods which returning values for creating universal base sructures

private static float CalculateAllTime(float distance, float speed)// Calculate Time spended on all way

{

float result = distance / speed;

return result;

}

private static float CalculateDistance(float speed, float time)// Calculate Distance with using current speed

{

float result = speed \* time;

return result;

}

private static float ConvertUnitsOfSpeed(float speed) // convert units from metres/sec to killometres/sec

{

float resultofSpeed = speed \* 3.6f;

return resultofSpeed;

}

private static float ConvertUnitsOfTime(float time) // convert units from metres/sec to killometres/sec

{

float resultofTime = time / 60f;

return resultofTime;

}

#endregion

static void Main(string[] args)

{

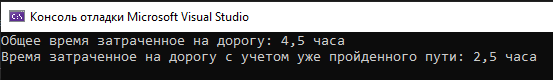
Console.WriteLine("Общее время затраченное на дорогу: " + CalculateAllTime(CalculateDistance(ConvertUnitsOfSpeed(30), ConvertUnitsOfTime(10)), 4f) + " часа");

Console.WriteLine($"Время затраченное на дорогу с учетом уже пройденного пути: { CalculateAllTime(CalculateDistance(ConvertUnitsOfSpeed(30), ConvertUnitsOfTime(10)), 4f) - 2f} часа");

}

}

}



Задание 5.

using System;

using System.Collections.Generic;

namespace SimpleLogic

{

class Program

{

public static void DecideOftheTask() {

float AllCash = 44000;

List<float> spendedMoney = new List<float>();

spendedMoney.Add(11000);

spendedMoney.Add( AllCash \* 0.25f);

spendedMoney.Add(AllCash \* 0.15f);

foreach (float value in spendedMoney) {

AllCash -= value;

}

Console.WriteLine("Осталось " + AllCash + " $");

}

static void Main(string[] args)

{

DecideOftheTask();

}

}

}



Задание 6.

using System;

using System.Collections.Generic;

namespace SimpleLogic

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine(Math.Pow(((60f - (14f + 16f)) / 5f), (22f - 19f)));// \*\* was replacement on classic function Math.Pow()

}

}

}



Задание 7.

using System;

using System.Collections.Generic;

namespace SimpleLogic

{

class Program

{

static void Main(string[] args)

{

List<string> Words = new List<string>() { "fanat", "apple", "abracadabra", "Tarantino", "yahoo", "yandex", "gool", "logo" }; Words.Sort(); foreach (string str in Words) { Console.WriteLine(str); }

}

}

}

