**17 Сборки, библиотеки, атрибуты, директивы**

Задание 1. Создать библиотеку с двумя классами:

Треугольник (методы ввода сторон, проверки на существование, вычисления периметра, вычисления площади, определения вида треугольника (разносторонний, равнобедренный, равносторонний));

Прямоугольник (методы ввода сторон, вычисления периметра, вычисления площади).

Листинг программы:

Треугольник:

namespace ShapeLib

{

public class Triangle

{

public double a, b, c; double P;

public enum type

{

Isosceles,

Equilateral,

Versatile

}

public Triangle(double a, double b, double c)

{

if (!IsExistTriangle(a, b, c))

{

throw new Exception("Треугольник не существует!");

}

else

{

this.a = a;

this.b = b;

this.c = c;

}

}

public bool IsExistTriangle(double a, double b, double c)

{

bool prov = ((a > 0) && (b > 0) && (c > 0) && ((a+b>c)&&(b+c>a)&&(a+c>b)));

return prov;

}

public double Perimetr()

{

P = a + b + c;

return P;

}

public double Square()

{

double S = Math.Sqrt(P \* (P - a) \* (P - c) \* (P - b));

return S;

}

public type TriangleType()

{

if (a == b && b == c)

{

return type.Equilateral;

}

else if ((a == b && a != c) || (b == c && b != a) || (a == c && a != b))

{

return type.Isosceles;

}

else

{

return type.Versatile;

}

}

}

}

Прямоугольник:

namespace ShapeLib

{

public class Rectanglee

{

public double a, b; double P;

public Rectanglee(double a, double b)

{

if (!ExistTriangle(a, b))

{

throw new Exception("Прямоугольник не существует!");

}

else

{

this.a = a;

this.b = b;

}

}

public bool ExistTriangle(double a, double b)

{

bool prov = ((a > 0) && (b > 0));

return prov;

}

public double Perimetr()

{

P = 2 \* a + 2 \* b;

return P;

}

public double Square()

{

double S = a \* b;

return S;

}

}

}

Вывод:

namespace MyNameSpace

{

public class Program

{

public static void Main()

{

Console.Write("Введите первую сторону: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите вторую сторону: ");

double b = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите третью сторону: ");

double c = Convert.ToDouble(Console.ReadLine());

Console.WriteLine();

try

{

Triangle Tr = new(a, b, c);

Console.WriteLine("Пириметр: " + Tr.Perimetr());

Console.WriteLine("Площадь: " + Tr.Square());

Console.WriteLine("Тип: " + Tr.TriangleType());

Console.WriteLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message,ex.StackTrace);

}

Console.Write("Введите высоту: ");

double d = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите длинну: ");

double e = Convert.ToDouble(Console.ReadLine());

Console.WriteLine();

try

{

Rectanglee Re = new(d, e);

Console.WriteLine("Пириметр: " + Re.Perimetr());

Console.WriteLine("Площадь: " + Re.Square());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

}

}

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4  8  5  4  6 | 17  154,4927182750048  Versatile  20  24 |

Анализ результатов:

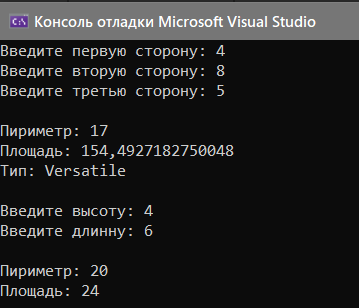


Рисунок 17.1 – Результат работы программы

Задание 2. Создать библиотеку на тему «Хлеб» и продемонстрировать ее функциональность.

Листинг программы:

namespace BreadLib

{

public class Bread

{

double weight;

public enum BreadtType

{

Black,

White

}

BreadtType type;

public Bread(BreadtType type, double weight)

{

if (!BreadChek(weight))

{

throw new Exception("У вас нет хлеба :(");

}

else

{

this.type = type;

this.weight = weight;

}

}

public bool BreadChek(double weight)

{

bool HaveBread = weight > 0;

return HaveBread;

}

public double Kcal()

{

double kcalIn100, kcal;

if (type == BreadtType.Black)

{

kcalIn100 = 222;

kcal = kcalIn100 \* (weight / 100);

return kcal;

}

else if (type == BreadtType.White)

{

kcalIn100 = 265;

kcal = kcalIn100 \* (weight / 100);

return kcal;

}

else

{

throw new Exception("Странный хлеб!");

}

}

}

}

Вывод:

using BreadLib;

namespace MyNameSpace

{

public class Program

{

public static void Main()

{

Console.Write("Введите вес хлеба: ");

double weight = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите тип хлеба\n1. Белый\n2. Черный");

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

Bread.BreadtType type;

if (NumType == 1)

{

type = Bread.BreadtType.Black;

}

else

{

type = Bread.BreadtType.White;

}

try

{

Bread Joe = new(type, weight);

Console.WriteLine("Кол-во ккал: " + Joe.Kcal());

Console.WriteLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message, ex.StackTrace);

}

}

}

}

Таблица 10.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 200  2 | 530 |

Анализ результатов:

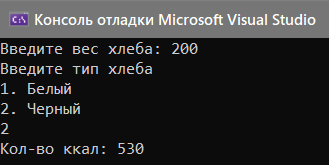


Рисунок 17.2 – Результат работы программы