События

Задание №1. Создать 3 метода, подсчитывающих периметр, площадь квадрата, и выводящий сторону треугольника. Создать делегат. В программе вызывать все три метода при помощи делегата.

Листинг программы:

namespace Geometry

{

class Program

{

delegate void GeometryDelegate();

static void Main(string[] args)

{

// Создаётся экземпляр делегата

GeometryDelegate geometryDelegate = new GeometryDelegate(CalculateGeometry);

// Вызываются все методы с помощью делегата

geometryDelegate.Invoke();

Console.ReadKey();

}

static void CalculateGeometry()

{

try

{

// Вычисление периметра и площади квадрата

int side = 5;

int perimeter = 4 \* side;

int area = side \* side;

Console.WriteLine($"Квадрат со стороной {side} имеет периметр {perimeter} и площадь {area}");

// Вычисляется сторона треугольника

int hypotenuse = 10;

int leg = 6;

int sideTriangle = (int)Math.Sqrt(hypotenuse \* hypotenuse - leg \* leg);

Console.WriteLine($"Треугольник с гипотенузой {hypotenuse} и катет {leg} его сторона {sideTriangle}");

}

catch (Exception e)

{

// Регистрируется и обрабатывается исключение

Console.WriteLine($"Ошибка: {e.Message}");

}

}

}

}

Анализ программы:



Рисунок 1.1 - Анализ программы

Источник: собственная разработка

Задание №2. На основе своей программы создать метод, в котором делегат будет параметром.

Листинг программы:

using System;

namespace Geometry

{

class Program

{

// Определение делегата

delegate void GeometryDelegate(int a);

static void Main(string[] args)

{

// Создание экземпляра делегата

GeometryDelegate del = new GeometryDelegate(CalculateArea);

// Вызов делегата с разными аргументами

del(5);

del(10);

del(15);

// Вызов метода с делегатом в качестве параметра

UseDelegate(del);

}

// Метод вычисления периметра квадрата

static void CalculatePerimeter(int side)

{

int perimeter = 4 \* side;

Console.WriteLine($"Периметр квадрата со стороной {side} равна {perimeter}");

}

// Метод вычисления площади квадрата

static void CalculateArea(int side)

{

int area = side \* side;

Console.WriteLine($"Площадь квадрата со стороной {side} равна {area}");

}

// Метод чтобы напечатать длину стороны треугольника

static void PrintTriangleSide(int side)

{

Console.WriteLine($"Длина стороны треугольника равна {side}");

}

// Метод использования делегата в качестве параметра

static void UseDelegate(GeometryDelegate del)

{

del(20);

}

}

}

Анализ программы:

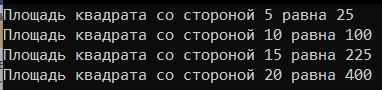


Рисунок 1.2 - Анализ программы

Источник: собственная разработка

Задание №3. Создать класс, в нем делегат и соответствующее ему событие. Создать два класса-наблюдателя, в методах которых будет описана их реакция на событие. Добавить 2 обработчика к событию из первого класса и один из второго. Вывести результат на экран. Удалить один обработчик события и вывести результат на экран.

Листинг программы;

using System;

namespace EventHandling

{

public delegate void EventHandler(object sender, EventArgs e);

public class MyClass

{

public event EventHandler MyEvent;

public void RaiseEvent()

{

Console.WriteLine("Event raised");

OnMyEvent(EventArgs.Empty);

}

protected virtual void OnMyEvent(EventArgs e)

{

MyEvent?.Invoke(this, e);

}

}

public class Observer1

{

public void HandleEvent(object sender, EventArgs e)

{

Console.WriteLine("Observer1: Event handled");

}

}

public class Observer2

{

public void HandleEvent(object sender, EventArgs e)

{

Console.WriteLine("Observer2: Event handled");

}

}

class Program

{

static void Main(string[] args)

{

MyClass myClass = new MyClass();

Observer1 observer1 = new Observer1();

Observer2 observer2 = new Observer2();

myClass.MyEvent += observer1.HandleEvent;

myClass.MyEvent += observer2.HandleEvent;

myClass.MyEvent += observer1.HandleEvent;

myClass.RaiseEvent();

myClass.MyEvent -= observer1.HandleEvent;

myClass.RaiseEvent();

Console.ReadLine();

}

}

}

Анализ программы:

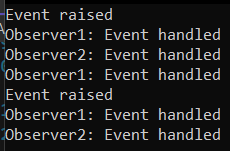


Рисунок 1.3 - Анализ программы

Источник: собственная разработка