**Не откладывать ошибки**

Если в свойстве Name окажется null, то с ошибкой завершится метод FormatStudent. Чтобы предотвратить отложенную ошибку, сделайте так, чтобы свойству Name нельзя было присвоить null. При попытке это сделать бросайте исключение оператором throw new ArgumentException();

private static void WriteStudent()

{

// ReadName считает неизвестно откуда имя очередного студента

var student = new Student { Name = ReadName() };

Console.WriteLine("student " + FormatStudent(student));

}

private static string FormatStudent(Student student)

{

return student.Name.ToUpper();

}

public class Student

{

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

}

**Программа:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace umop14o6zDontPostponeMistakes

{

class Program

{

static void Main(string[] args)

{

WriteStudent();

Console.ReadKey();

}

private static void WriteStudent()

{

// ReadName считает неизвестно откуда имя очередного студента

var student = new Student { Name = ReadName() };

Console.WriteLine("student " + FormatStudent(student));

}

private static string FormatStudent(Student student)

{

return student.Name.ToUpper();

}

public class Student

{

private string name;

public string Name

{

get { return name; }

set

{

if (value == null) throw new ArgumentException("\"Name\" does not support NULL");

name = value;

}

}

}

public static string ReadName()

{

return "Jhon Daniel";

}

}

}