Министерство образования республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Институт информационных технологий

Специальность Программное обеспечение информационных технологий

**КОНТРОЛЬНАЯ РАБОТА**

По предмету «ООТПиСП»

Вариант №10

Студент-заочник 2 курса

Группы: №681072

ФИО: Иванов Кирилл Евгеньевич

Тел.: +375(29) 319 43 00

Минск, 2017

**Содержание**

Содержание 2

Задание 1 3

1.1. Формулировка задания 3

1.2. Диаграмма классов 3

1.3. Текст программы 3

1.4. Результат выполнения программы 5

Задание 2 6

2.1. Формулировка задания 6

2.2. Диаграмма классов 6

2.3. Текст программы 6

2.4. Результат работы программы 8

Задание 3 9

3.1. Формулировка задания 9

3.2. Диаграмма классов 9

3.3. Текст программы 9

3.4. Результат работы программы 11

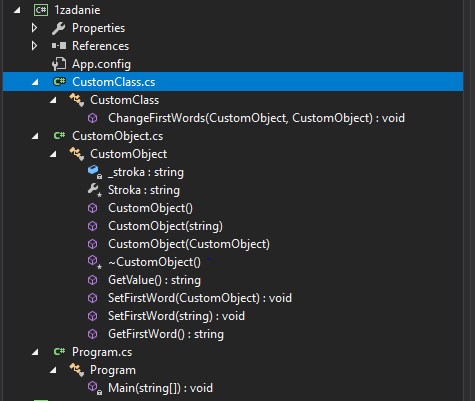
Список использованных источников 12

**Задание 1**

**1.1. Формулировка задания**

Создать 2 объекта разработанного класса. Одной из компонент класса является символьная строка. В результате выполнения программы в обоих объектах поменять первые слова строки. Содержимое объектов (их строки) до и после обмена вывести на экран.

**1.2. Диаграмма классов**



**1.3. Текст программы**

CustomClass.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace proj

{

class CustomClass // тут выводится результат //При реализации задания 1 функцию вывода содержимого объекта реали-зовать как внешнюю по отношению к классу

{

public static void ChangeFirstWords(CustomObject a, CustomObject b) // принимает два обьекта, выводит их начальное значение, вызывает ф-ции замены первых слов и выводит окончательный вариант

{

Console.WriteLine("In start strings:");

Console.WriteLine(" ");

Console.WriteLine(" A string: {0}", a.GetValue());

Console.WriteLine(" B string: {0}", b.GetValue());

Console.WriteLine(" ");

Console.WriteLine("Change first words in strings");

Console.WriteLine(" ");

//делаем копии обьектов, используя конструктор копирования

CustomObject modA = new CustomObject(a);

CustomObject modB = new CustomObject(b);

modA.SetFirstWord(b);

modB.SetFirstWord(a);

Console.WriteLine("After change strings:");

Console.WriteLine(" ");

Console.WriteLine(" A string: {0}", modA.GetValue());

Console.WriteLine(" B string: {0}", modB.GetValue());

}

}

}

CustomObject.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace proj

{

class CustomClass // тут выводится результат //При реализации задания 1 функцию вывода содержимого объекта реали-зовать как внешнюю по отношению к классу

{

public static void ChangeFirstWords(CustomObject a, CustomObject b) // принимает два обьекта, выводит их начальное значение, вызывает ф-ции замены первых слов и выводит окончательный вариант

{

Console.WriteLine("In start strings:");

Console.WriteLine(" ");

Console.WriteLine(" A string: {0}", a.GetValue());

Console.WriteLine(" B string: {0}", b.GetValue());

Console.WriteLine(" ");

Console.WriteLine("Change first words in strings");

Console.WriteLine(" ");

//делаем копии обьектов, используя конструктор копирования

CustomObject modA = new CustomObject(a);

CustomObject modB = new CustomObject(b);

modA.SetFirstWord(b);

modB.SetFirstWord(a);

Console.WriteLine("After change strings:");

Console.WriteLine(" ");

Console.WriteLine(" A string: {0}", modA.GetValue());

Console.WriteLine(" B string: {0}", modB.GetValue());

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace proj

{

class Program

{

static void Main(string[] args)

{

// создаем два экземпляра обьекта с параметром стринг

CustomObject a = new CustomObject("THIS IS SPARTA!");

CustomObject b = new CustomObject("It is UIStackView!");

// вызывает ф-цию, которая вывведет результат

CustomClass.ChangeFirstWords(a, b);

// что бы консоль не закрылась

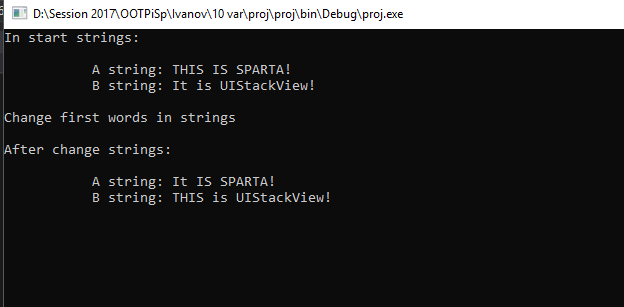
Console.ReadKey();

}

}

}

**1.4. Результат выполнения программы**

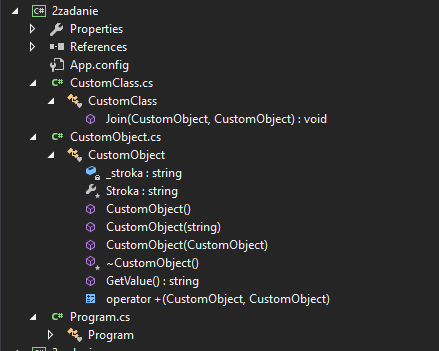


**Задание 2**

**2.1. Формулировка задания**

Создать несколько объектов (например, a и b) разработанного класса. Класс – символьная строка. Реализовать для объектов данного класса перегрузку операции + (c=a+b). Содержимое объектов (a,b,c, их строк), до и после выполнения опера-ции, вывести на экран.

**2.2. Диаграмма классов**



**2.3. Текст программы**

CustomClass.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_2zadanie

{

class CustomClass // тут выводится результат //При реализации задания 1 функцию вывода содержимого объекта реали-зовать как внешнюю по отношению к классу

{

public static void Join(CustomObject a, CustomObject b) // принимает два обьекта, выводит их начальное значение, вызывает перегруженную ф-цию + и выводит окончательный вариант

{

Console.WriteLine("In start strings:");

Console.WriteLine(" ");

Console.WriteLine(" A string: {0}", a.GetValue());

Console.WriteLine(" B string: {0}", b.GetValue());

Console.WriteLine(" ");

Console.WriteLine("C = A + B = {0}", a + b);

}

}

}

CustomObject.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_2zadanie

{

class CustomObject

{

private string \_stroka;

protected string Stroka

{

get { return \_stroka; }

set { \_stroka = value; }

}

//дефолтный конструктор

public CustomObject()

{

Stroka = "This is My default string.";

}

//конструктор с параметром

public CustomObject(string stroka)

{

Stroka = stroka;

}

//конструктор копирования

public CustomObject(CustomObject Stroka)

{

this.Stroka = Stroka.GetValue();

}

~CustomObject()

{

this.Stroka = null;

Console.WriteLine("was destroy");

}

//получение значение обьекта с доступностью public

public string GetValue()

{

return this.Stroka;

}

public static string operator +(CustomObject a, CustomObject b)

{

return String.Join(" ", a.GetValue(), b.GetValue());

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_2zadanie

{

class Program

{

static void Main(string[] args)

{

// создаем два экземпляра обьекта с параметром стринг

CustomObject a = new CustomObject("THIS IS SPARTA?");

CustomObject b = new CustomObject("NO! THIS IS PATRIK!");

// вызывает ф-цию, которая вывведет результат

CustomClass.Join(a, b);

// что бы консоль не закрылась

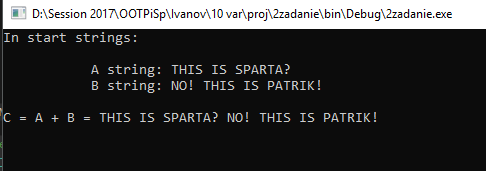
Console.ReadKey();

}

}

}

**2.4. Результат работы программы**

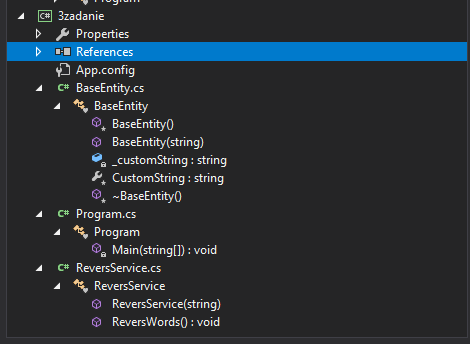


**Задание 3**

**3.1. Формулировка задания**

Создать иерархию классов представляющих простое наследование. Базовый класс – строка символов (char \*). Производный класс – методы, работающие с дан-ными базового класса. Реализовать в производном классе метод – переворота каж-дого слова (по отдельности) в строке базового класса.

**3.2. Диаграмма классов**



**3.5. Текст программы**

BaseEntity.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3zadanie

{

class BaseEntity

{

protected BaseEntity()

{

}

public BaseEntity(string customString)

{

CustomString = customString;

}

private string \_customString;

protected string CustomString

{

get { return \_customString; }

set { \_customString = value; }

}

~BaseEntity()

{

Console.WriteLine("was destroy");

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3zadanie

{

class Program

{

static void Main(string[] args)

{

var stroka = "Argentina manit negra O\_o";

ReversService service = new ReversService(stroka);

service.ReversWords();

Console.Read();

}

}

}

RevercecService.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_3zadanie

{

class ReversService: BaseEntity

{

public ReversService(string customString)

{

CustomString = customString;

}

public void ReversWords()

{

Console.WriteLine("Start string:");

Console.WriteLine(" {0}", CustomString);

Console.WriteLine(" ");

Console.WriteLine("Revers words in string.");

var splitWord = CustomString.Split(' ');

Console.Write(" ");

foreach (var word in CustomString.Split(' '))

{

char[] reversArray = word.ToCharArray();

Array.Reverse(reversArray);

Console.Write(" {0}", new String(reversArray));

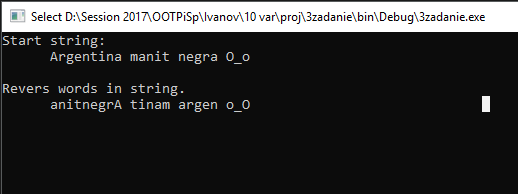
}

}

}

}

**3.6. Результат работы программы**



**Список использованных источников**

* 1. Пол А. Объектно-ориентированное программирование на С# – Изд-во Бином, Невский Диалект, 2001. – 464 с.
  2. Буч Г. Объектно-ориентированное программирование с примерами применения Издательство: M.: Конкорд, раниц; 1992 г. – 519 с
  3. Форум программистов и сисадминов CyberForum.ru (Электронный ресурс). URL: http://www.cyberforum.ru/.