МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по лабораторной работе №3

**«Конструкторы, поля только для чтения, вызов конструкторов»**

Работу выполнил  
студент 41 группы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.C. Онянов

Направление подготовки 01.03.02 Прикладная математика и информатика

Проверил  
канд. техн. наук, доц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Полупанов

Краснодар 2021

**Цель работы:** является изучение структуры проекта на ЯП C#, создания классов, объектов, конструкторов классов, Аксессоров, Индексаторов.

**Задача 1:** Инициализация класса без использования конструктора. Компилятор автоматически инициализировал переменные стандартными значениями.

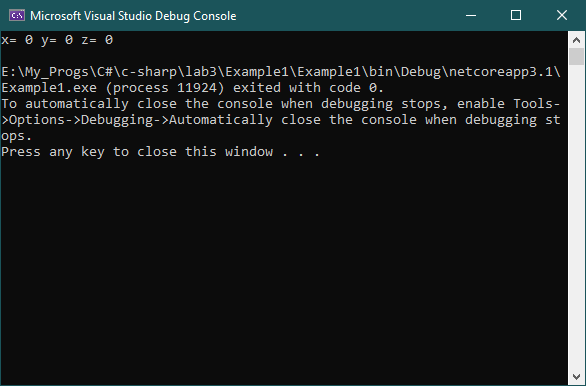


Рисунок 1.

Теперь добавим конструктор

**public CA()**

**{**

**x=3;**

**y=2;**

**z=x+y;**

**}**

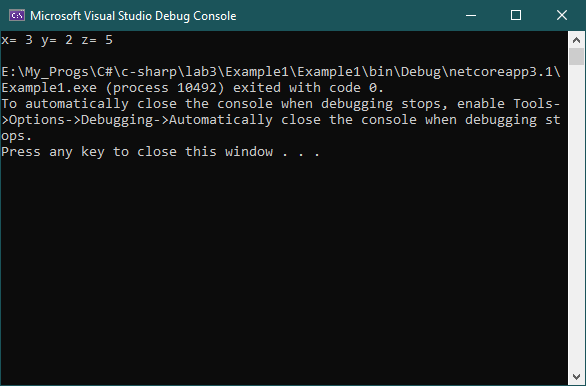


Рисунок 2.

В этом случае данные инициализировались так, как было указано в конструкторе.

Возможно использование конструкторов с параметрами. Тогда при создании экземпляра класса можно вызывать конструкцию

CA obj2 = new CA(5, 7, 25);

Эта конструкция будет обработана конструктором с параметрами.

public CA(int a,int b,int c)

{

x=a;

y=b;

z=c;

}

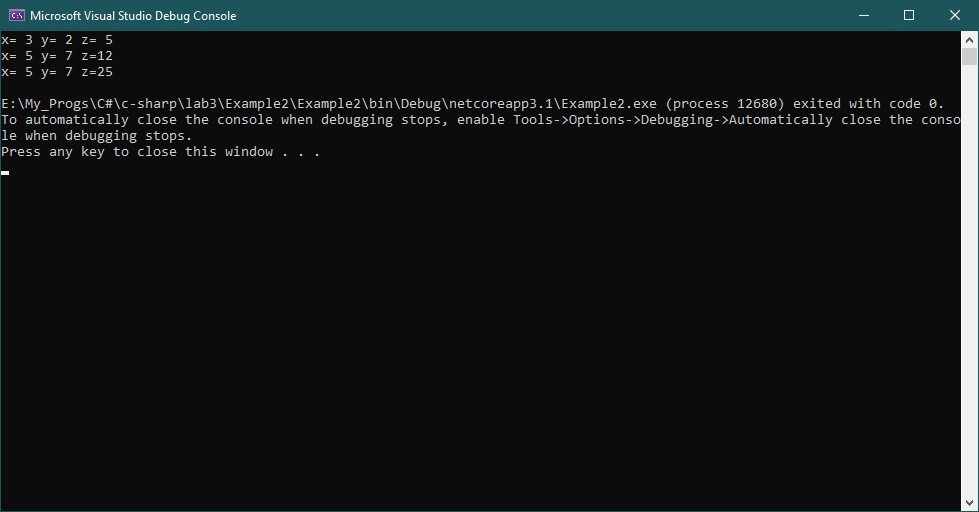


Рисунок 3.

В С# возможно использование полей только для чтения. Они помечаются ключевым словом readonly. Отличие от констант – значение задается только в теле класса. Для этого используется конструктор класса.

public int x,y,z;

public readonly int rd;

public CA()

{

x=3;

y=2;

z=x+y;

rd=x+y+z;

}

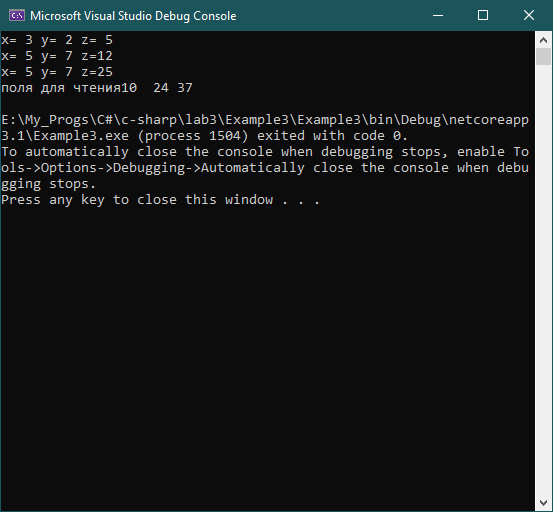


Рисунок 4.

Возможен вызов конструкторов из других классов. Для этого конструктор необходимо определить с ключевым словом this.

public CA(int a,int b,int c):this(a,b)

{

z=c;

rd=x+y+z;

}

Пример 4. Вызов внешних конструкторов.

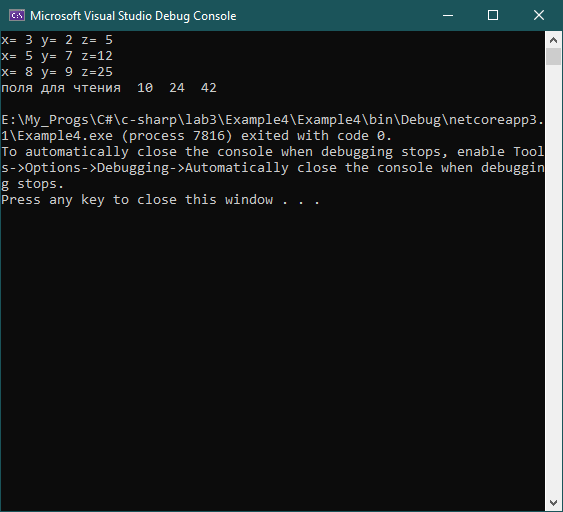


Рисунок 5.

**Задание 2:** Создать приложение, в котором в цикле создается 10 объектов класса. Поля только для чтения каждого экземпляра равны порядковому номеру, отражающего очередность создания объектов.

using System;

namespace Task1

{

class Program

{

class CA

{

public int x, y, z;

public readonly int rd;

public CA(int x, int y, int a) : this(a)

{

z = x + y;

rd = x + y + z;

}

public CA(int a)

{

rd = a;

}

}

static void Main(string[] args)

{

CA[] arr = new CA[10];

for (int i = 0; i < 10; i++)

{

arr[i] = new CA(i);

Console.WriteLine("arr[{0}] = {1}", i, arr[i].rd);

}

}

}

}

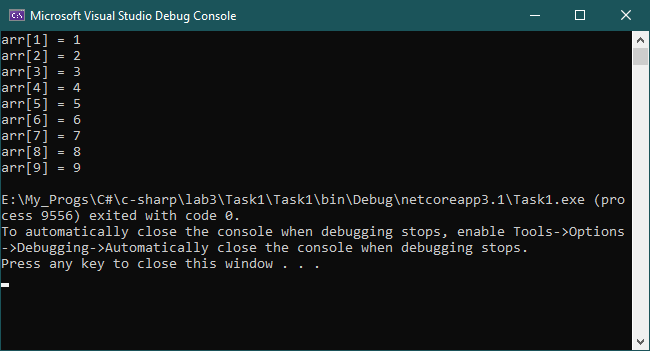


Рисунок 6.

**Задание 3:** Изменить тело конструктора с аргументами таким образом, чтобы производился анализ диапазона значения передаваемого аргумента. При выходе из некоторого диапазона - печаталось соответствующее сообщение, а значение аргумента приравнивалось, в зависимости от ситуации, минимально или максимально допустимому значению.

using System;

using System.Text;

namespace Task2

{

class CA

{

public int x, y, z;

public readonly int rd;

public CA(int x, int y, int a) :this(a)

{

z = x + y;

rd = x + y + z;

}

public CA(int a)

{

if (a >= 0 && a <= 10)

{

rd = a;

}

else if (a < 1)

{

Console.WriteLine("Присваиваемое значение выходит за диапазон значений ");

rd = 0;

}

else

{

Console.WriteLine("Присваиваемое значение выходит за диапазон значений ");

rd = 10;

}

}

}

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

CA test1 = new CA(-1);

CA test2 = new CA(7);

CA test3 = new CA(11);

}

}

}

Результат выполнения:

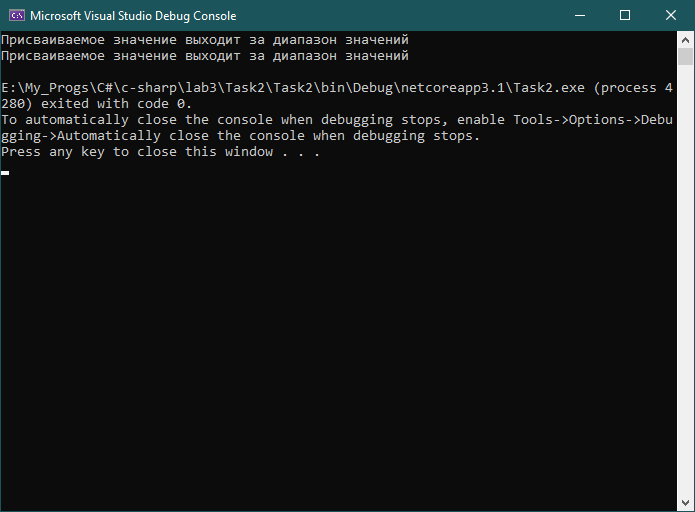


Рисунок 7.

**Вывод:** В ходе лабораторной работы были изучены принципы работы с конструкторами классов. Рассмотрены основные возможности и ограничения.