Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**«КОНСТРУКТОР КЛАССА. ПЕРЕГРУЗКА МЕТОДОВ КЛАССА»**

**ОТЧЕТ**

**по лабораторной работе №7**

**дисциплины**

**«Технологии программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Мизин Глеб Егорович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

Проработка учебной задачи:



Рисунок 1 – Код учебной задачи



Рисунок 2 – Код учебной задачи

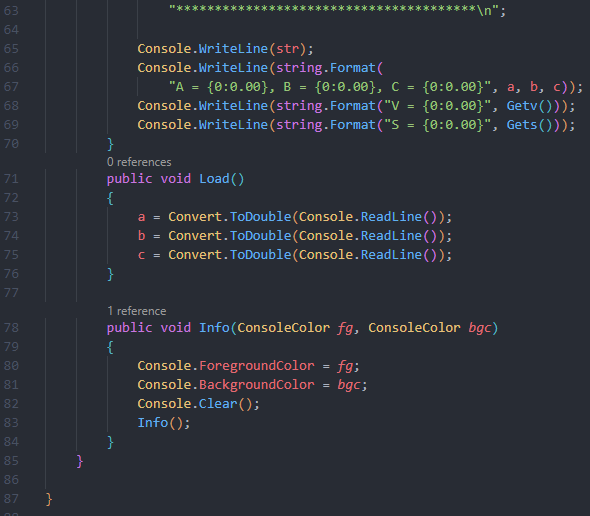


Рисунок 3 – Код учебной задачи

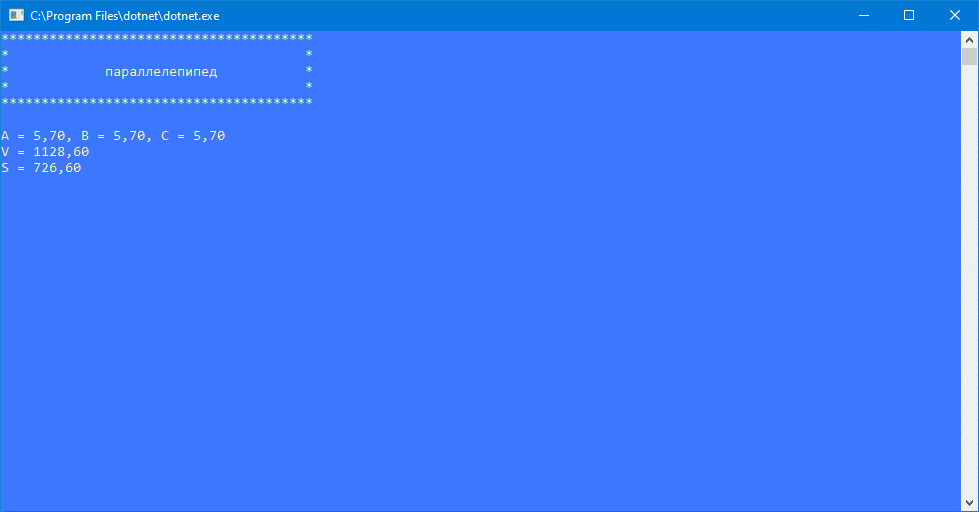


Рисунок 4 – Результат работы учебной задачи в режиме Debug

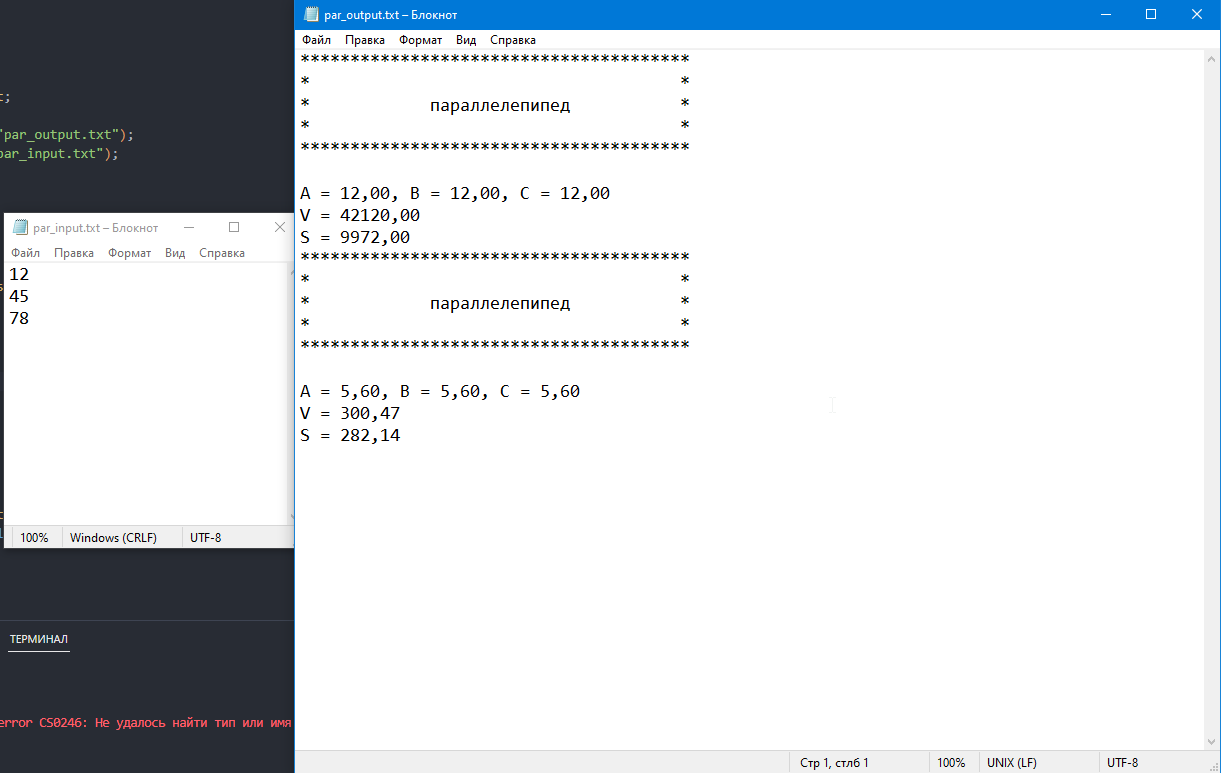
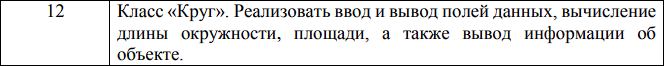


Рисунок 4 – Результат работы учебной задачи в режиме Release

Индивидуальное задание:



Листинг программы индивидуального задания:

namespace Helloworld

{

    class Program

    {

        static void Main(string[] *args*)

        {

#if !DEBUG

        TextWriter save\_out = Console.Out;

        TextReader save\_in = Console.In;

        var new\_out = new StreamWriter(@"par\_output.txt");

        var new\_in = new StreamReader(@"par\_input.txt");

        Console.SetOut(new\_out);

        Console.SetIn(new\_in);

#endif

        Par p1, p2;

#if DEBUG

        p2 = new Par (5.7);

        p2.Info(ConsoleColor.Yellow, ConsoleColor.Blue);

#endif

#if !DEBUG

        p1 = Par.CreatreParFromFile();

        p1.Info();

        p2 = new Par(5.6);

        p2.Info();

#endif

#if !DEBUG

        Console.SetOut(save\_out); new\_out.Close();

        Console.SetIn(save\_in); new\_in.Close();

#endif

#if DEBUG

        Console.ReadKey();

#endif

        }

    }

    class Par

    {

        private const double PI = 3.1415926535897931;

        private double r;

        private Par() { }

        public Par (double *pR*)

        {

            r=*pR*;

        }

        public static Par CreatreParFromFile()

        {

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

            return new Par(RR);

        }

        public double GetL() { return 2 \* PI \* r; }

        public double GetS() { return r \* r \* PI; }

        public void Info()

        {

            String str =

                "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n" +

                "@@@@@@@@@@@@@@@@@@@%#%%%&&&&&&&&&&%%%%#&@@@@@@@@@@@@@@@@@@@\n" +

                "@@@@@@@@@@@@@@&%%&&&&&&&&&&&&&&&&&&&&&&&&%%%%@@@@@@@@@@@@@@\n" +

                "@@@@@@@@@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%%@@@@@@@@@@@\n" +

                "@@@@@@@@%%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%&@@@@@@@@\n" +

                "@@@@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%&@@@@@@\n" +

                "@@@@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@@\n" +

                "@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%&@@@\n" +

                "@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@\n" +

                "@&&&&&&&&&     Это круг, просто поверьте мне    &&&&&&&&&%@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%%@\n" +

                "@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@\n" +

                "@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@\n" +

                "@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@\n" +

                "@@@@&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@@\n" +

                "@@@@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@@@@\n" +

                "@@@@@@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@@@@@@\n" +

                "@@@@@@@@@@@%&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&%#@@@@@@@@@@@\n" +

                "@@@@@@@@@@@@@@%%%&&&&&&&&&&&&&&&&&&&&&&&&&&%@@@@@@@@@@@@@@@\n" +

                "@@@@@@@@@@@@@@@@@@@&%&&&&&&&&&&&&&&&&%%&@@@@@@@@@@@@@@@@@@@\n" +

                "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n" ;

            Console.WriteLine(str);

            Console.WriteLine(string.Format("R = {0:0.00}", r));

            Console.WriteLine(string.Format("Длина круга = {0:0.00}", GetL()));

            Console.WriteLine(string.Format("Площадь круга = {0:0.00}", GetS()));

        }

        public void Load()

        {

            r = Convert.ToDouble(Console.ReadLine());

        }

        public void Info(ConsoleColor *fg*, ConsoleColor *bgc*)

        {

            Console.ForegroundColor = *fg*;

            Console.BackgroundColor = *bgc*;

            Console.Clear();

            Info();

        }

    }

}

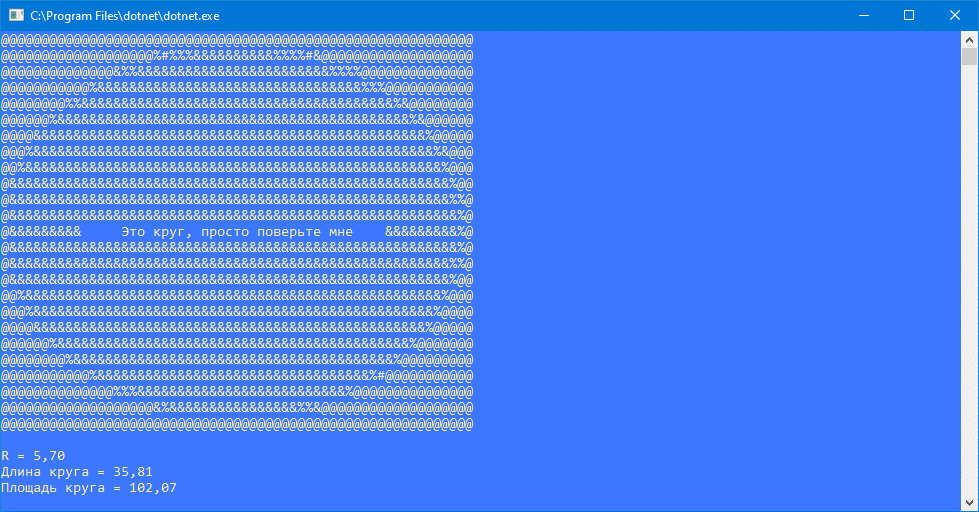


Рисунок 5 – Результат работы программы в режиме Debug

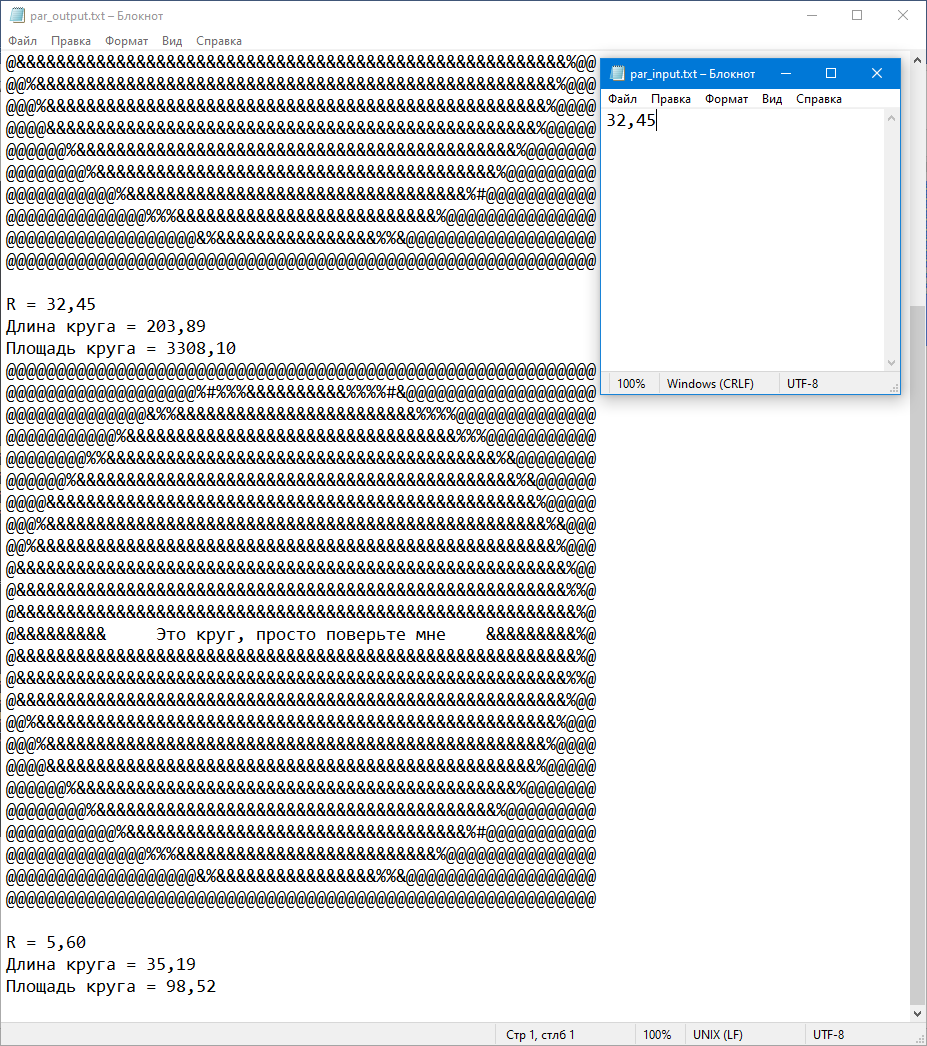


Рисунок 6 – Результат работы программы в режиме Release

Контрольные вопросы

1. Что такое статический класс? Что такое статический метод?

**Статический метод** (static) класса – метод, не имеющий доступа к состоянию (полям) объекта, то есть к переменной this.

2. Что такое конструктор класса?

Конструктор – это метод класса, который не возвращает значения и имеет то же самое имя, что и класс. Если конструктор класса не определен программистом явно, то компилятор создаст конструктор по умолчанию.

3. Что такое перегрузка методов?

В C# поддерживается перегрузка методов – то есть может существовать несколько версий одного метода, но с разными сигнатурами (методы отличаются количеством и / или типом параметров). Чтобы перегрузить метод, просто объявляются методы с одинаковыми именами, но разными сигнатурами.

4. Может ли один конструктор класса вызывать другой конструктор? Прежде чем отвечать попробуйте реализовать такой вызов в своем разработанном классе.

Нет, нельзя. Конструктор класса может вызваться только из другого конструктора этого же класса. Главное назначение конструктора класса – создание экземпляров (объектов) класса

5. Сколько перегрузок может иметь метод класса?

В классе перегруженный метод может иметь любое количество реализаций.

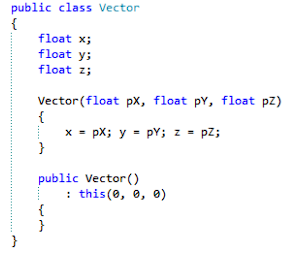
6. Для каких целей следует перегружать конструктор класса?

Например: один из конструкторов не принимает параметры, второй принимает два параметра, третий принимает три параметра. Позже, во время создания объекта, параметры передаются в качестве аргументов. Так компилятор сможет определить, какой из объявленных конструкторов применить при создании объекта

7. Что такое сигнатура метода?

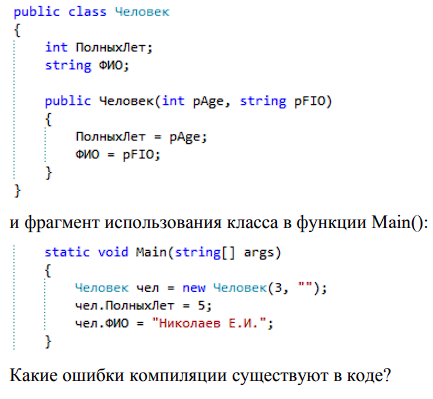
Сигнатура метода — это имя метода плюс параметры (причем порядок параметров имеет значение). В сигнатуру метода не входит возвращаемое значение, а также бросаемые им исключения.

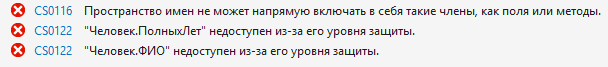
8. Какие ошибки присутствуют в объявлении класса?



Ошибок нет

9. Дано определение класса:





10. Укажите ошибки (если есть) в объявлении класса:

