

**ОТЧЕТ О ВЫПОЛНЕНИИ**

**практической работы №10**

**Тема: « Методы в C#».**

Выполнил: ст.гр. Алирзаева З.Н

Специальность: 09.02.07

Информационные системы и программирование

Проверил: преподаватель Боклач Б. И.

Москва

2022

**Цель работы**:

Получение практических навыков по разработке методов.

**Вариант 1**.

**Задание:** Разработать консольное приложение, осуществляющее работу со строковыми данными. В программе в качестве метода оформить обработку данных.

Дана символьная строка. Заменить все символы '!' точками, кроме первого и вывести полученную строку.

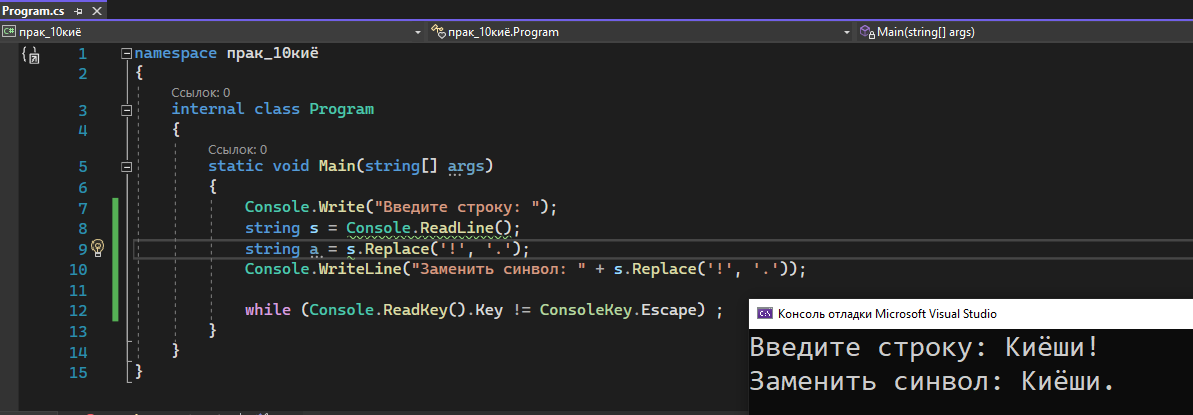


Рис 1.

**Вывод:**

В данной практической работе я познакомилась с задачей, который имел свой способ решения, а также овладела практическими навыками по разработками методов.

**Контрольные вопросы:**

1. **Для каких целей предназначены спецификаторы в структуре описания метода?**

Спецификатор **public** разрешает доступ к данному члену классу из любой части программы, т.е. другими словами любой метод или член другого класса может получить доступ к члену класса, у которого спецификатор стоит **public**. В свою очередь, **private** запрещает открытый доступ к члену класса, т.е. доступ к нему имеет только члены и методы данного класса. Из других классов получить доступ к данному члену класса (у которого стоит спецификатор доступа **private**) нельзя.

1. **Какие требования предъявляются к имени метода?**

К названиям методов предъявляются в принципе те же требования, что и к названиям переменных. Однако, как правило, названия методов начинаются с большой буквы. Перед названием метода идет возвращаемый тип данных. Здесь это тип void , который указыает, что фактически ничего не возвращает, он просто производит некоторые действия. После названия метода в скобках идет перечисление параметров. Но в данном случае скобки пустые, что означает, что метод не принимает никаких параметров. После списка параметров в круглых скобках идет блок кода, который представляет набор выполняемых методом инструкций.

1. **Какое отличие между параметрами-переменными и параметрами-значениями?**

Отличие параметра-переменной от обычного параметра (т. е. параметра-значения) состоит в том, что инструкции процедуры или функции могут изменить значение переменной, являющейся фактическим параметром. Использование параметров-переменных позволяет использовать процедуры для изменения значений переменных основной программы.

1. **Назначение оператора Return.**

Оператор **return** завершает выполнение функции и возвращает управление вызывающей функции. Выполнение возобновляется в вызывающей функции в точке сразу после вызова. Оператор **return** может возвращать значение, передавая его вызывающей функции. Дополнительные сведения см. в статье Тип возвращаемого значения.

1. **Формальные параметры и фактические параметры: назначение и различия.**

Формальные параметры – это параметры, которые принимают значение, переменные функции от функционального вызова. Формальные параметры объявляются в списке параметров в начале описания функции. Список параметров определяет имена параметров и порядок, в котором они принимают значение при вызове функции. Формальные параметры – входные и выходные. Входные – те, которые должны быть известны для вычисления в функции. Выходные – результат.