**Выполнили студент 215 группы**

**Марков К.А.**

**Практическое занятие 17**

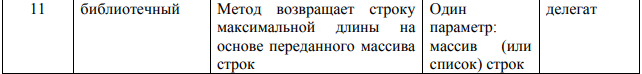
**Тема: «Применение обратных асинхронных делегатов»**

**Цель работы:** научиться использовать обратные асинхронные вызовы для организации ожидания завершения выполнения асинхронного метода.

**Задание:**

1. Создайте консольное приложение C# в MS Visual Studio. 26

2. Реализуйте решение задачи в соответствии с индивидуальным вариантом с использованием асинхронного вызова метода. При реализации получения результата работы асинхронного метода используйте механизм обратного асинхронного вызова.



class Program

{

// Объявление делегата для обратного вызова

delegate void StringResultCallback(string result);

static async Task Main(string[] args)

{

// Пример массива строк

string[] stringArray = { "яблоко", "банан", "виноград", "арбуз" };

// Вызываем асинхронный метод и ожидаем результат

await GetStringWithMaxLengthAsync(stringArray, HandleStringResult);

Console.ReadKey();

}

// Асинхронный метод для получения строки максимальной длины из массива строк

static async Task GetStringWithMaxLengthAsync(string[] stringArray, StringResultCallback callback)

{

await Task.Run(() =>

{

// Ищем строку максимальной длины

string maxLengthString = stringArray[0];

foreach (var str in stringArray)

{

if (str.Length > maxLengthString.Length)

maxLengthString = str;

}

// Вызываем метод обратного вызова с результатом

callback(maxLengthString);

});

}

// Метод обратного вызова для обработки результата

static void HandleStringResult(string result)

{

Console.WriteLine($"Строка максимальной длины: {result}");

}

}

**Вопросы:**

Обратные асинхронные вызовы (reverse asynchronous calls) обычно используются в контексте асинхронного программирования для выполнения длительных операций, которые требуют обратной связи. Обратные асинхронные вызовы позволяют асинхронным методам сообщать вызывающему коду о завершении операции или передаче промежуточных результатов.

Примером обратного асинхронного вызова может быть использование колбэков или событий для уведомления о завершении выполнения операции. Например, асинхронная операция может быть инициирована, а затем вызывающий код ожидает уведомления об успешном завершении или об ошибке.

В C# и .NET такие обратные асинхронные вызовы могут быть реализованы с помощью делегатов, событий, а также путем использования **TaskCompletionSource**, который позволяет создавать задачи, завершение которых управляется вручную.

**Вывод:** научился использовать обратные асинхронные вызовы для организации ожидания завершения выполнения асинхронного метода.