ФГАОУ ВО «Омский государственный технический университет» Кафедра «Автоматизированные системы обработки информации и управления»

### ОТЧЁТ

по дисциплине «Разработка программного кода» студента Чернуха Ильи Евгеньевича группы ИВТМ-221

на тему «Интерфейсы»

### Пояснительная записка

Шифр работы РГР–02068999–43–Ивтм-221 ПЗ

Направление 09.04.01 Информатика и вычислительная техника Направленность (профиль) «Автоматизированные системы обработки информации и управления»

Доцент П.Н. Надточий

Студент И.Е. Чернуха

Омск 2023

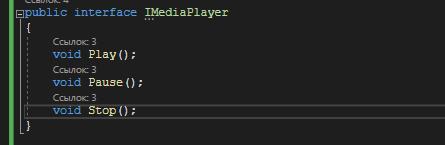
**Задание:**

В рамках выполняемого курсового проекта по дисциплине «Разработка программного кода» спроектируйте интерфейс и опишите классы, его реализующие. При этом необходимо описать ту часть проекта, для реализации которой создавался интерфейс и классы, его реализующие.

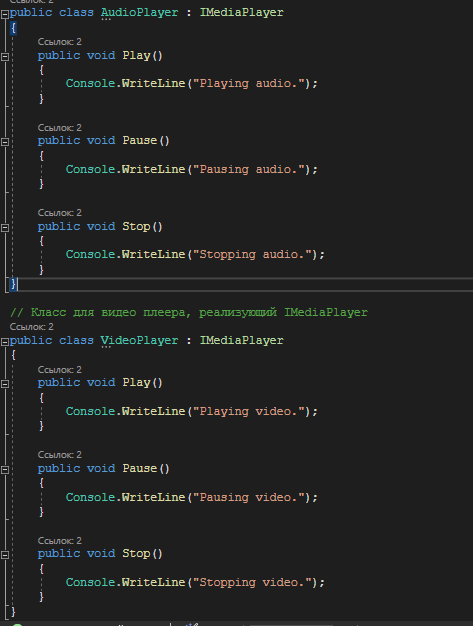
Интерфейсы будут спроектированы с помощью языка C#. C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык.

Интерфейс (interface) представляет собой не более чем просто именованный набор абстрактных членов. Абстрактные методы являются чистым протоколом, поскольку не имеют никакой стандартной реализации. Конкретные члены, определяемые интерфейсом, зависят от того, какое поведение моделируется с его помощью.

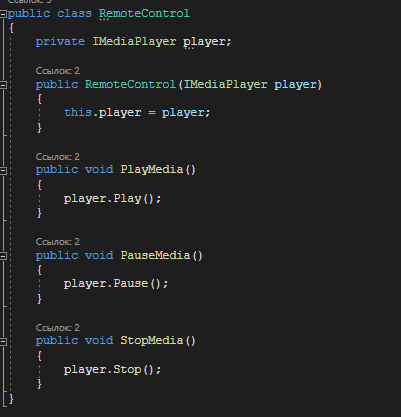
**Решение:**



IMediaPlayer - интерфейс определяет операции для воспроизведения медиафайлов, такие как Play, Pause и Stop.

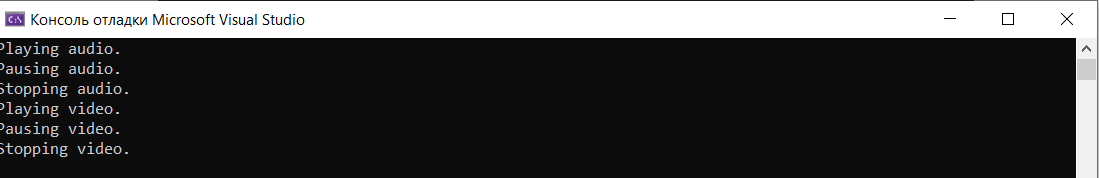


AudioPlayer и VideoPlayer - классы реализуют интерфейс IMediaPlayer и предоставляют конкретную реализацию операций для аудио и видео воспроизведения.



RemoteControl - класс, представляющий главное управляющее устройство. Конструктор принимает объект плеера, реализующего IMediaPlayer. Класс RemoteControl использует интерфейс IMediaPlayer для управления воспроизвдения.

**Демонстрация работы:**



Создали объекты плееров и два управляющих устройства для аудио и видео плеера, после чего воспроизвели аудио и видео.

**Листинг программы:**

// Интерфейс для устройства, поддерживающего воспроизведение медиа

public interface IMediaPlayer

{

void Play();

void Pause();

void Stop();

}

// Класс для аудио плеера, реализующий IMediaPlayer

public class AudioPlayer : IMediaPlayer

{

public void Play()

{

Console.WriteLine("Playing audio.");

}

public void Pause()

{

Console.WriteLine("Pausing audio.");

}

public void Stop()

{

Console.WriteLine("Stopping audio.");

}

}

// Класс для видео плеера, реализующий IMediaPlayer

public class VideoPlayer : IMediaPlayer

{

public void Play()

{

Console.WriteLine("Playing video.");

}

public void Pause()

{

Console.WriteLine("Pausing video.");

}

public void Stop()

{

Console.WriteLine("Stopping video.");

}

}

// Класс для главного управляющего устройства

public class RemoteControl

{

private IMediaPlayer player;

public RemoteControl(IMediaPlayer player)

{

this.player = player;

}

public void PlayMedia()

{

player.Play();

}

public void PauseMedia()

{

player.Pause();

}

public void StopMedia()

{

player.Stop();

}

}

public class Program

{

public static void Main(string[] args)

{

// Создаем объекты плееров

AudioPlayer audioPlayer = new AudioPlayer();

VideoPlayer videoPlayer = new VideoPlayer();

// Создаем главное управляющее устройство для аудио плеера

RemoteControl audioRemoteControl = new RemoteControl(audioPlayer);

// Воспроизведение аудио

audioRemoteControl.PlayMedia();

audioRemoteControl.PauseMedia();

audioRemoteControl.StopMedia();

// Создаем главное управляющее устройство для видео плеера

RemoteControl videoRemoteControl = new RemoteControl(videoPlayer);

// Воспроизведение видео

videoRemoteControl.PlayMedia();

videoRemoteControl.PauseMedia();

videoRemoteControl.StopMedia();

}

}