W08 Articulate-Inheritance

CSE210-Cecilia Michalek

Inheritance is a principle of object-oriented programming. It allows you to create a new class or classes based on and existing class. The exciting class is called a base or parent class, and the new classes are called derived or child class. The derived classes inherit the properties and behaviors of the base class. They can access one or more as it is needed.

The biggest benefit if this property is that it allows for code reusability and a structure of hierarchy organization in the program.

Another benefit is maintenance. Whatever changes you make to the base class are automatically inherited to the derived classes.

Here is and example:

using System;

public class Music {

protected string \_singersName = "";

protected string \_song = "";

public Music(string singersName, string song) {

\_singersName = singersName;

\_song = song;

}

public string GetSingersName(){

return \_singersName;

}

public void SetSingersName(string singersName) {

\_singersName = singersName;

}

public string GetSong() {

return \_song;

}

public void SetSong(string song) {

\_song = song;

}

public string GetSummery() {

return $"{\_singersName} - {\_song}";

}

}

public class Genre : Music {

private int \_year;

private string \_producer;

public Genre (string singersName, string song, string producer, int year) : base(singersName, song){

\_year = year;

\_producer = producer;

}

public string GetMusicInfo(){

return $"{\_singersName} - {\_song} - {\_year} - {\_producer}";

}

}

This demonstrates how inheritance in C# allows you to create specialized classes based on more general ones, this improves code organization and reuse.