**Inheritance**

Inheritance is a fundamental concept in object-oriented programming where a new class (called a derived or subclass) is created based on an existing class (called a base or superclass). The derived class inherits the properties and behaviors (methods and attributes) of the base class, allowing it to reuse code and extend functionality.

One significant benefit of inheritance is code reusability, as it enables the derived class to inherit and utilize the functionality of the base class without needing to re-implement it. This promotes a more efficient and organized codebase, reducing redundancy and making maintenance easier.

An application of inheritance in the program I wrote is evident in the activity classes hierarchy. For instance, all activity classes (BreathingActivity, ReflectingActivity, ListingActivity, GratitudeActivity) inherit common functionalities (such as displaying starting and ending messages) from the base Activity class, while each activity class specializes in its unique behaviors.

public class Activity

{

protected string \_name;

protected string \_description;

protected int \_duration;

public Activity(string name, string description, int duration)

{

\_name = name;

\_description = description;

\_duration = duration;

}

public virtual void DisplayStartingMessage()

{

Console.WriteLine($"Starting {\_name} activity...");

Console.WriteLine($"Description: {\_description}");

Console.WriteLine($"Duration: {\_duration} seconds");

}

public virtual void DisplayEndingMessage()

{

Console.WriteLine($"Congratulations! You have completed the {\_name} activity.");

}

// Other shared methods and attributes

}

public class BreathingActivity : Activity

{

public BreathingActivity(string name, string description, int duration) : base(name, description, duration)

{

}

public void Run()

{

DisplayStartingMessage();

// Breathing activity specific implementation

DisplayEndingMessage();

}

}

In this example, the BreathingActivity class inherits the DisplayStartingMessage and DisplayEndingMessage methods from the base Activity class, allowing it to focus on implementing its specific functionality. This demonstrates how inheritance promotes code reuse and facilitates modular design.