Jordan Peterson

Polymorphism Explanation

Meaning of Polymorphism

Polymorphism is a principle of object orientated programming where child classes can inherit attributes and methods from a master class and then alter them as needed to suit their more specialized use case. Polymorphism simplifies adding new classes or behaviors without modifying existing code. It allows you have have generic, or even blank, methods set on the base class which set an expectation for a set of common behaviors in the derived classes. This can greatly reduce code duplication, enhance maintainability, and support scalability as more features are added.

And example of polymorphism from my Eternal Quest project is the following.

From the Goal class:

From the Goal class  
public abstract string GetStringRepresentation();

From the ChecklistGoal class which inherits the Goal class:

public override string GetStringRepresentation()

{

*return* $"|{GetDetailsString()}|{\_target}|{\_amountComplete}|{\_bonus}";

}

In this case the base class creates an empty method called GetStringRepresentation. This sets the expectation that all of the derived class will have this behavior. But then it’s up to each of the derived classes to set the specific behavior based on their needs. Here each derived class needs to be able to create a unique string based on what attributes are important for them. Some have more, and some have less.