Jordan Peterson

Inheritance Explanation

Inheritance is a fundamental concept in object-oriented programming that allows one class to inherit properties and methods from another class. This mechanism enables code reuse and establishes hierarchical relationships between classes. The child class has the ability to extend or override the functionality of the parent class.

There are several major benefits to the use of inheritance. The first major benefit of inheritance is code reusability. By inheriting from a parent class, a subclass can reuse existing code without rewriting it. This reduces duplication, simplifies code maintenance, and promotes modularity in software design.

Second, it aids in abstraction by allowing the parent class to focus on high levels of functionality, while the child classes focus on low level functionality. Changes can be easily made to child classes without worry that it will effect other child classes.

An example of code reusability from my project is the use of a method in the parent class Activity, which calculates the amount of time that was spent by the user in an activity. At first I had this code in each child class and was sending the calculated trueDuration back to the parent class. But by adding this method to the parent class I’m able to just send the start and end times of the activity and it can handle the calculation.

*public* *void* SetTrueDuration(DateTime *startTime*, DateTime *trueEndTime*)

{

TimeSpan timeDifference = *trueEndTime* - *startTime*;

*int* totalSeconds = (*int*)timeDifference.TotalSeconds;

\_trueDuration = totalSeconds;

}