Object-Oriented

Paradigm

Object Oriented Paradigm

Object Oriented programming relies on having multiple “Objects”, including classes, variables and methods. These objects can be referred to from inside other objects, and have a few key features. This allows for scalability thanks you multiple instances of each object or class, as well as simple library integration and simplification and abstraction for code blocks and private variables.

These features afford OOP a much simpler design philosophy, focussing on abstraction as said before, with complex code blocks being easily referred to as objects and instances of objects.

Object Oriented Characteristics

Encapsulation, where classes and methods contain private variables that are encapsulated entirely within the parent object.

Abstraction, where classes can be simplified into instances of objects rather than large amounts of variables and methods.

Inheritance, where classes can inherit properties, methods and attributes from parent classes. Inheritance also features polymorphism, which is where objects that inherit from a parent can be clearly differentiated and used for different tasks, while still being derived from the same parent.

(Petkov, n.d.)

Procedural

Paradigm

Procedural Paradigm

The Procedural paradigm features multiple elements, including functions and methods, local variables and global variables. Functions are code blocks that can utilise local variables and global variables to return data or to perform other functions (i.e. console.write and console.read differ in that console.write returns nothing). Local variables are instanced variables that only exist in the instance of each function call, usually being unable to be accessed outside of the function they're contained within. Global variables are variables that are non constrained by the instance of a function call they're contained within, and can be accessed anywhere and any time.

The key feature that procedural programming results in is modularity, allowing for large blocks of text to be reduced to a single function that can be repeated and used as a module of a larger code block or implementation.

(Bhatia, n.d.)

Event-Driven

Paradigm

Event Driven Paradigm

The Event Driven paradigm simply involves having code that runs because of an event, such as a GUI interaction like pressing a button. These events are similar to functions in the Procedural paradigm, however in this case they rely on an event listener, which is a specific language-level implementation of code designed to check for a condition, in this case an event.

This improves on procedural loops that check every code iteration as there’s a single, low complexity (in terms of code complexity as well as time complexity) solution for listening to an event.

(Computerhope.com, 201

Paradigm Similarities

Object Oriented programming links very well with Event Driven programming, as it allows for objects to have event interactions as method calls within the main classes and objects.

OOP and Procedural programming work well together, less so because they have an interesting synergy, but more so because having abstraction and simplicity that comes from OOP links well with the simplicity that if statements and for/while loops afford thanks to the Procedural paradigm. The abstraction and simplicity of these two elements make it much easier to program in languages that feature both due to the simplicity and the fact that they’re more intuitive than most alternatives.

Procedural and Event Driven programming work very well together, as they allow for if/then decisions on event calls, speeding up software production and simplifying the code path of a given set of variables.

Bibliography

Petkov, A. (n.d.). *How to explain object-oriented programming concepts to a 6-year-old*. [online] freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/ [Accessed 12 Nov. 2019].

Bhatia, S. (n.d.). What is Procedural Programming? Key Features of Procedural Programming. [online] Hackr.io. Available at: https://hackr.io/blog/procedural-programming [Accessed 13 Nov. 2019].

Computerhope.com. (2019). What is Event-driven Programming?. [online] Available at: https://www.computerhope.com/jargon/e/event-driven-prog.htm [Accessed 13 Nov. 2019].

Computerhope.com. (2019). What is an Event Listener?. [online] Available at: https://www.computerhope.com/jargon/e/event-listener.htm [Accessed 13 Nov. 2019].