Universidad de San Carlos de Guatemala
Escuela de Ciencias Físicas y Matemáticas
Laboratorio de Simulación
Diego Sarceño 201900109
7 de mayo de 2022

Tarea 2

1. Programación Orientada a Objetos

En programación se tienen varios paradigmas, los cuales son útiles en diferentes circunstancias y para diferentes niveles de conocimiento. El más famoso es el de programación secuencial, uno "contrario" a este es el de programación en paralelo, el cual busca utilizar todo el poder del procesador. Normalmente, cuando se enseña a programar, luego de estudiar la programación estructurada, se introduce un concepto muy útil en muchas ramas del saber, función. La programación funcional es muy útil al estar trabajando con problemas divididles en subproblemas. Luego, con esta idea en mente, se puede tener paquetes/librerías de funciones para poder reutilizar en varios programas y con mayor facilidad de depuración, a esto se le conoce como programación modular.

Partiendo de la idea de la programación modular y funcional, se introduce el paradigma de programación orientada a objetos (POO), el cual mezcla la idea de programación modular en un solo objeto de programación, llamado "objeto", valga la redundancia. Un ojeto esta combuesto de atributos (datos) y métodos (funciones) y un cojunto de objetos es una clase. En POO el programa no se divide en tareas sino en modelos. Lenguajes de programación que utilizan este paradigma como base: C++, C#, Java y JavaScript; también es posible utilizar este paradigma en otros lenguajes como Python o R.

1.1. Características

Abstracción: Es la base de cualquier programa a realizar, no solo en POO. Es la forma en la que el programador resuelve el problema en cuestión, de modo que el lenguaje sea capaz de entenderlo y de realizar los respectivos procedimientos. Es basicamente es traducir la solución general a un algoritmo entendible por la máquina, obviamente siguiendo la POO.

Encapsulamiento: Es la organización de todos los elementos del programa, juntando aquellos que pertenezcan a la misma entidad (mismo nivel de abstracción).

Herencia: Es la relación que se tiene entre diferentes clases, esto es para facilitar y optimizar el código; también ayuda al polimorfismo, dos métodos que se nombran igual pero con características heredadas de distintas clases, realizarán susrespectivos procesos independientemente de su nombre.

Polimorfismo: Son comportamientos diferentes de objetos distintos, con entradas distintas que poseen el mismo nombre. Esto es una característica de muchos lenguajes de programación, el único punto bajo es que esta característica aumenta el tiempo de ejecución del programa.

Modularidad: Con lo explicado en los primeros párrafos referido a los diferentes paradigmas de programación, los módulos son subdiviciones de una aplicación, las cuales deben ser lo más independientemente posible de la aplicación en sí, así como de sus partes. La idea es facilitar la depuración y tener un programa más ordenado.

Al estudiar y trabajar con este paradigma de programación, es importante tener dominio de sus respectivos conceptos, es decir, entender el paradigma en sí. Para ello se introducen dichos conceptos

Clase: Es una plantilla en donde se definen los atributos y métodos de cierto objeto.

Objeto: Es la instancia de una clase provista de atributos y métodos. Estos corresponden a objetos reales.

Método: Es un algoritmo asociado a un objeto, el cual se ejecuta luego de la recepción de un mensaje. Estos también pueden conectarse con algún otro objeto del sistema.

Atributos: Característica, datos y variables de la clase.

1.2. Lenguaje Unificado de Modelado

En programación orientada a objetos se tiene una especie de "análogo" al diagrama de flujo, el cual se denomina como Lenguaje Unificado de Modelado (UML). Este se estableció como un modelo estandarizado para describir un enfoque de POO. Los diagramas de clases, son una tabla con 3 filas en las cuales se colocan el nombre de la clase, sus atributos y los métodos (en los cuales, preferiblemente, se colocan sus respectivas entradas y tipo de salidas).

1.3. Ventajas y Desventajas de la POO

Como en cualquier cosa, se tienen sus respectivas ventajas y desventajas, acontinuación se mencionan algunas de ellas.

1.3.1. Ventajas

1. Reusabilidad.
2. Mantenibilidadad.
3. Modificabiliad.
4. Fiabilidad.

Las cuales, parecen ser basntante claras en su concepción.

1.3.2. Desventajas

1. Es un cambio bastante drástico entre la idea común de programación y creación de algoritmos (secuencial, funcional, etc.), por lo que suele ser complicado adaptarse a esta nueva forma de abstracción.
2. La velocidad de ejecución es más lenta.