Tarea 1. Conceptos Sobre Lenguajes de Programación

Layla Tame A01192934

Problema 1

El concepto de un lenguaje de programación ha evolucionado mucho a lo largo de los años. En un principio se conocía simplemente como la notación que se utiliza para decirle a una computadora que queremos que haga. Sin embargo, esta definición fue descartada con la introducción de los lenguajes de bajo nivel, también conocidos como lenguajes ensambladores. Con la creación de estos se dejó de utilizar la práctica de “hardwired” para dar instrucciones a una máquina. Posteriormente surgen los lenguajes de alto nivel, los cuales permitieron mejor la forma de dar instrucciones generando poco cambio en su uso entre diferentes máquinas por medio del uso estandarizado de cierta notación. Estos lenguajes permiten el uso de ciclos, selecciones y asignaciones de variables.

Mediante el avance y los descubrimientos con respecto al hardware de una computadora, el concepto de un lenguaje de programación pasó de ser instrucciones directamente relacionadas a la máquina, a una serie de instrucciones de procesamiento generales. Entonces, un lenguaje de programación se puede definir como un sistema de notación utilizado para describir la computación de forma que sea legible para ambos, una máquina y un ser humano. Para que un lenguaje sea legible por una máquina, es necesario que tenga una estructura simple y que esta pueda ser traducida. Cuando se habla de que un lenguaje pueda ser legible por un humano, se refiere a que mantenga un nivel de abstracción que permita a una persona comprender lo que sucede detrás de la ejecución.

Los tipos y niveles de abstracción ayudan a mejorar la comprensión de un programa para un ser humano. Estos se pueden dividir en dos principales tipos de abstracción: de datos y de control. Cuando se habla de abstracción de datos se encapsula las propiedades relacionadas con números, colecciones y strings. Por otro lado la abstracción de control encapsula las propiedades de loops, if statements y funciones. Existe también la clasificación de abstracciones en base a los niveles, siendo tres principales: básica, estructurada y unitaria. La abstracción básica hace referencia a la capacidad de un lenguaje de interpretar un espacio de memoria asignado como una variable que contiene tanto nombre como un valor. La abstracción estructurada es la manera en que se encapsulan las estructuras de datos, ya sea en forma de listas, arreglos o diccionarios. Finalmente se tiene la abstracción unitaria, la cual trata de modular programas y separarlos en base a su comportamiento. Cuando un lenguaje cumple con los suficientes mecanismos para describir todas las computaciones que puede realizar una máquina de Turing, se dice que el lenguaje es Turing completo.

Existen diferentes paradigmas de programación dentro de los cuales se pueden clasificar los lenguajes. Es importante recalcar que aun que es posible que un lenguaje cumpla con la mayoría o con todas las propiedades de un paradigma en específico, hay muy pocos lenguajes que se adhieren solamente a uno en específico, si no que comúnmente contienen propiedades de diferentes paradigmas. Existen cuatro tipos de paradigmas: imperativo, funcional, orientado a objetos y lógico.

El paradigma imperativo, que en ocasiones también se conoce como procedural, abarca todos aquellos lenguajes que ejecutan las instrucciones dadas paso por paso. En un inicio, los lenguajes de programación estaban basados en el funcionamiento de una máquina con modelo von Neumann. Esto significa que tenían que cumplir con tres propiedades: que la ejecución de las instrucciones fuera de forma secuencial, que el uso de las variables representen espacios de memoria y finalmente que se permita el uso de asignación para cambiar el valor de una variable establecida. Cuando se cumplen los tres requisitos se dice que el lenguaje es imperativo. Aun que muchos lenguajes hoy en día son procedurales, se tiene la desventaja de que no se pueden utilizar para realizar operaciones en paralelo.

Los lenguajes considerados como funcionales suelen tratar las funciones como tipos de datos o variables. Estos lenguajes se basan en su mayoría en la recursividad de las funciones para poder iterar. El paradigma funcional se basa en la valoración de funciones para poder ejecutar el programa de manera exitosa. Esto permite que se pueda cambiar de una función a otra para poder ejecutar correctamente. Los lenguajes funcionales también son conocidos como lenguajes aplicativos y se consideran ser opuestos a los lenguajes orientados a objetos ya que se concentran en valores y funciones en lugar de espacios de memoria.

El paradigma orientado a objetos se basa en la noción de un objeto, que puede ser descrito como la colección de memoria junto con las operaciones que permiten cambiar los valores guardados en estas direcciones de memoria. Estos lenguajes pueden ser considerados como una extensión de los lenguajes imperativos ya que de la misma forma mantienen el estilo de conllevar las operaciones de forma secuencial. Este tipo de lenguajes permiten la creación de diferentes clases para asignar comportamiento individual a los objetos que después permite modificar los datos y las variables de forma sencilla y eficiente.

Finalmente se tiene el paradigma lógico. Este se caracteriza por que su manera de funcionar va enfocada a definir los requerimientos y no precisamente el proceso explícito de como alcanzarlo paso por paso. Tiende a tener una inclinación hacia la programación lógica para alcanzar el resultado establecido por medio del uso de statements que describen lo que se requiere. Este tipo de programación también es conocida como programación declarativa.