PROGRAMACIÓN

INTRODUCCIÓN

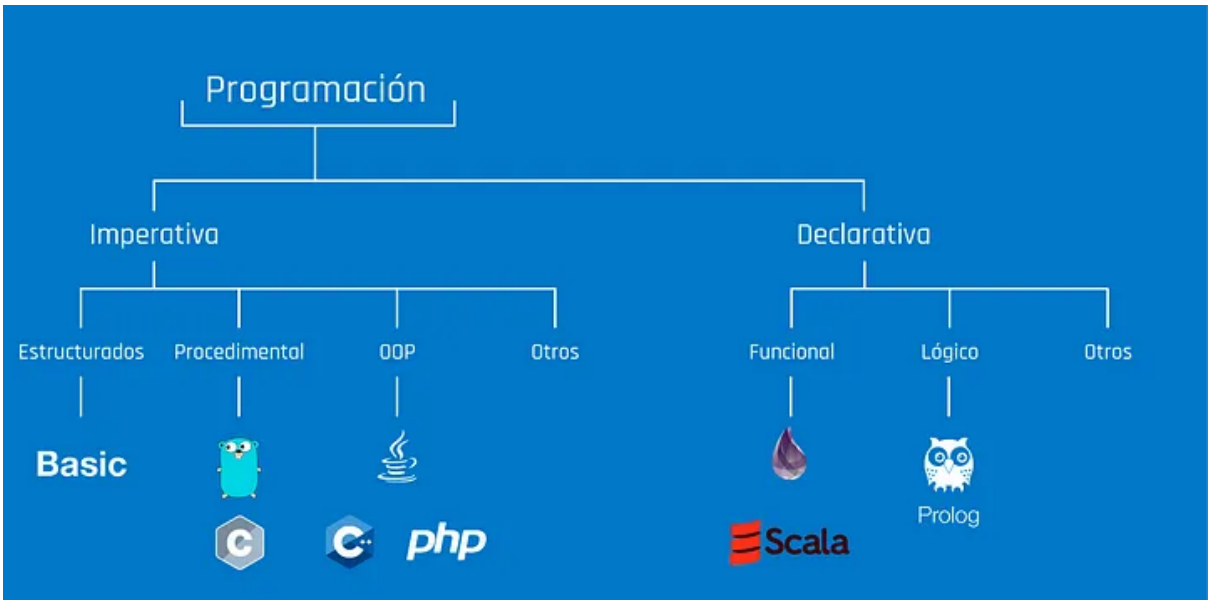
A continuación voy a hacer un breve inciso sobre los lenguajes de programación, paradigmas y sus estándares, ¿Que es la programación?. La programación es la manera que tenemos de poder comunicarnos con una máquina la cuál le pedimos que haga una cierta cosa, ya sea para facilitarnos un trabajo, para divertirnos y jugar a un videojuego etc.
Luego están los paradigmas que nos sirven para estructurar nuestro código de una manera legible y eficaz, que nos permite poder reutilizar código y ahorrarnos muchas líneas de código.
Los estándares son reglas que han sido creadas por programadores para escribir código con unas ciertas pautas para así poder entender el código de otra persona y que tú lenguaje sea entendido por otra también.

Tipos de lenguaje de programación

Generalmente los lenguajes de programación se dividen en alto, medio y bajo nivel, ¿Y porque se dividen?. Bueno realmente la máquina habla un lenguaje binario (1 y 0, donde 1 = True y 0 = False), pero el ser humano no sería capaz de hablar semejante lenguaje por lo que se creó el lenguaje de bajo nivel Ensamblador, lo cuál tampoco era tarea fácil, entonces se crearon lenguajes de medio nivel como C, C es un lenguaje muy efectivo que nos permite tener un acceso directo a la memoria lo cual agiliza su compilación respecto a otros lenguajes (Compilar significa traducir un lenguaje al lenguaje a otro lenguaje, en este caso sería al lenguaje máquina) que se utiliza por ejemplo para crear sistemas operativo. Los lenguajes de medio nivel están muy bien pero para crear cosas más simples no son necesarios usarlos, por eso crearon los lenguajes de alto nivel que se parecen más a nuestro lenguaje cotidiano como Python, que nos sirve por ejemplo para desarrollo web o multiplataforma.

Paradigmas de programación

Los lenguajes de programación se dividen en dos estilos, programación imperativa y declarativa. La programación imperativa le estamos dando a la máquina una serie de pautas para que realice algo, ejemplos de lenguajes son por ejemplo Python, Java. En la declarativa le estamos diciendo a la máquina que es lo que queremos obtener, es por así decirlo como un filtro de búsqueda que le damos a la máquina para que nos muestre un resultado y nos da igual el “cómo” lo hace, un ejemplo de lenguaje sería SQL.
Dentro de estos 2 paradigmas de programación tendríamos varios subparadigmas de programación, que son los siguientes



Como bien había dicho en la introducción los paradigmas de programación nos permiten estructurar nuestro código en bloques, de manera que nos permite ahorrarnos reutilizar y ahorrar muchas líneas de código.
Existen muchos paradigmas de programación, pero uno de los más conocidos es el POO (Programación orientada a objetos) que es un paradigma imperativo, ¿Que es?
Bueno es un paradigma que se utiliza en los videojuegos que estructura el código en clases. Hay dos tipos de clases la clase padre y la clase hijo, generalmente la clase hijo va a heredar ciertas características de la clase padre, de esta manera nos ahorramos tener que escribir el mismo código muchas veces.
A continuación voy a mostrar parte de mi código orientado a POO creado en Python

```
from sys import exit
from random import randint

class Scene(object):

    def enter(self):
        print "This scene is not yet configured. Subclass it and implement enter ()."
        exit(1)

class Engine(object):

    def __init__(self, scene_map):

        self.scene_map = scene_map


    def play(self):
        current_scene = self.scene_map.opening_scene()

        while True:
            print "\n- - - - - - - - - "
        next_scene_name = current_scene.enter()
        current_scene = self.scene_map.next_scene(next_scene_name)


class Death(Scene):

    quips = [
        "You died. You Kinda suck at this",
        "Such a luser."
    ]

    def enter(self):
        print Death.quips[randint(0, len(self.quios)- 1)]
        exit(1)

class CentralCorridor(Scene):

 def enter(self):
        print "The Gothons of Planet Percal #25 have invaded your ship and destroyed"
```

También tenemos otro paradigma imperativo como lo es la programación procedimental que divide el código en partes llamadas procedimientos o funciones. De esta manera se consigue que el código sea más claro y que no sean necesarias las repeticiones de código gracias a las llamadas a las funciones y procedimientos. Ejemplos de este tipo de lenguajes son el Python.
Un ejemplo sería el siguiente código:

```
def  add5(x):
    return x+5

def  dotwrite(ast):
    nodename  =  getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print  '     %s  [label="%s'  %  (nodename,  label),
    if  isinstance(ast[1],  str):
        if  ast[1].strip():
            print  '= %s"];'  %  ast[1]
        else:
            print  '"]'
     else:
        print  '"];'
        children  =  []
        for  in n,  childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print  ,'     %s  ->  {'  %  nodename
        for  in :namechildren
            print  '%s'  %  name,
```

Luego tenemos la programación funcional, que es un paradigma declarativo, donde nos enfocaremos en "qué" estamos haciendo y no en "cómo" se está haciendo que sería el enfoque imperativo. Esto quiere decir que nosotros expresaremos nuestra lógica sin describir controles de flujo, es decir, no usaremos ciclos o condicionales.

Un ejemplos de programación funcional sería lo siguiente:

```
Long result = numeros.stream().filter(num -> num > 10).count();
System.out.println(result);
```

Podemos hacer lo mismo, pero de manera imperativa:

```
List<Integer> numeros = List.of(18, 6, 4, 15, 55, 78, 12, 9, 8);

int contador = 0;
for(int numero : numeros) {
    if(numero > 10) {
        contador ++;
    }
}
System.out.println(contador);
```

Aquí se ve como en el paradigma funcional ahorramos muchas líneas de código, lo cual nuestro código va a ser más eficiente.

Estándares de programación

Los estándares son normas que han sido creadas por programadores que facilitan la comprensión del código donde se promueve la uniformidad en la estructura y el estilo del código, facilitando la colaboración entre equipos y evitando errores causados por diferencias en la interpretación del código.
Unos ejemplos serían los siguientes:

- Google Java Style Guide:
  Características Principales: Reglas para el formato, estructura y mejores prácticas en Java.

- PEP 8 (Python Enhancement Proposal 8):
  Características Principales: Líneas de código que no superen los 72 caracteres, usar bien los espacios en blanco, un código mal indentado.

Seguir este tipo de reglas te va ser muy beneficioso a la hora de colaborar con otros programadores, ya que tu código podrá ser entendido por otras personas, además

el buen código no es aquel que nadie entiende es aquello que todos seríamos capaces de entender.

Conclusión

En resumen la comprensión de la diferencia que existe entre los diferentes lenguajes de programación, sus paradigmas, estilos y estándares nos permite tener un buen conocimiento de lo que es el desarrollo de software.
Conocer las diferencias entre los diferentes lenguajes de programación y sus paradigmas nos permite saber cuál sería la mejor opción para desarrollar un software.
Conocer sus estilos y estándares nos permite tener un buen hábito para que podamos entender el código de los demás programadores y que puedan entender el nuestro.

Referencias

Entre los libros que tengo leídos y conocimientos adquiridos a lo largo de mi etapa como programador, también he utilizado estas páginas para recopilar información.

https://www.hackaboss.com/blog/tipos-lenguajes-programacion
https://www.epitech-it.es/lenguaje-bajo-nivel/
https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/
https://learn.microsoft.com/es-es/dotnet/standard/linq/functional-vs-imperative-programming
https://medium.com/@Loopa/paradigmas-de-programaci%C3%B3n-programaci%C3%B3n-imperativa-y-programaci%C3%B3n-declarativa-4c4a4182fd87
https://codigofacilito.com/articulos/programacion-funcional
https://www.genbeta.com/desarrollo/programacion-imperativa-vs-declarativa-i