

Gerardo Omar Guerrero Garza.
322324217

Practica 2: Palabras Reservadas e Identificadores

1. Hacer una lista con los identificadores encontrados en el siguiente programa y expresar que es lo que están representando. No es necesario escribir más de una vez cada identificador.

- **Identificadores:** Está expresando el nombre de la clase del programa.
- **main:** Aunque es una palabra reservada, main aquí está expresando el nombre del método principal, el es el punto de partida de ejecución del programa.
- **pps:** Este identificador representa el arreglo de cadenas que van a llevar los argumentos de la línea de comandos.
- **pi:** Es el nombre de la variable tipo float y representa el número pi: 3.1416
- **radio:** Variable tipo int que representa el radio de un círculo
- **area:** Variable tipo double que representa el área calculada de un círculo
- **mensaje:** Variable tipo int que acompañara al resultado del área del circulo obtenida.

2. Dadas las siguientes declaraciones:

```
int x, y;  
float z = 3.1313f;  
double w = 3.1212;  
boolean verdad = true;
```

Evaluar las siguientes expresiones siempre y cuando sea posible. En caso de no ser posible la evaluación justificar la respuesta.

- **x = 6;** Está expresión es verdad pues se le está asignando el valor de 6 a una variable x que es entera.
- **y = 1000;** También es válida porque se le está asignando a una variable declarada como entera, un valor 1000 que es entero.
- **y = 2.33333;** Esta variable no es válida porque se le está asignado un valor que es no entero, ya que tiene decimales.
- **25 ++;** No es válida pues el operador incremento solamente se le aplica a variables y no a valores numéricos como 25.
- **w = 175, 000;** No es válida pues la sintaxis de Java no admite comas para separar números.
- **verdad = 1;** No es válida, pues la variable verdad es boolean y por lo tanto sólo admite valores true y false.
- **z = 3.1416;** No es válida pues al asignarle valores a un float siempre debe llevar la f al final.
- **(x + y) ++;** No es válida porque como dije antes, el operador incremento solamente se aplica a variables y no a expresiones como (x+y)

- **y = 1;** Es válida porque se le está aplicando el valor entero de 1 a una variable que es entera.

4. Investigar, ¿Por qué las String (Cadenas) no son un tipo primitivo en Java?.

Las cadenas no son tipos primitivos en Java porque son mucho más complejas que los tipos simples, requieren una gestión de memoria y funcionalidades avanzadas, y están diseñadas para integrarse completamente con la orientación a objetos del lenguaje.

Más que datos primitivos se consideran más como objetos y para manipularlas se ocupan métodos más avanzados como los substrings, la concatenación o el `indexOf()`, etc.. Además de que no es tan sencillo editar el contenido de un string, pues al nosotros modificar o cambiar algo de un string específico, lo que en realidad estamos haciendo, es crear otro string diferente.