**TAREA 1**

* **TÍTULO**

PROGRAMACIÓN ORIENTADA A OBJETOS

* **RESUELVE**

En la actualidad los principales lenguajes de programación de alto nivel son Java y C++. Se pide que realice un cuadro donde especifique las similitudes y diferencias entre ambos lenguajes.

Comparación respecto a operaciones de Entrada y salida.

Las operaciones de Entrada/Salida generalmente no están asociadas al lenguaje en sí por lo que es necesario utilizar bibliotecas.

En Java se utiliza el paquete java.io; mientras que en C++ se utilizará iostream.

Comparación respecto a operaciones de Entrada y salida.

Generalmente la E/S está asociada con flujos de archivos. La pantalla es considerada el archivo de salida estándar, el teclado es el archivo de entrada estándar y la pantalla es la salida de errores estándar.

Comparación respecto a operaciones de Entrada y salida.

La E/S de datos depende en gran medida de la interfaz de usuario de la aplicación. Las interfaces más genéricas son modo texto, modo gráfico.

En Java la Salida se da con System.out y algunos métodos, la Entrada con System.in

Comparación respecto a operaciones de Entrada y salida.

En C++ la salida se da con el objeto cout seguido del operador de indirección de entrada “<<”. Se puede concatenar cualquier tipo de dato. La entrada de datos se hace con el objeto cin y el operador de indirección de salida “>>”.

Comparación respecto a operaciones de Entrada y salida.

Por ejemplo, se desea obtener el área de un triángulo por lo que se tienen que pedir al usuario los datos de la base y la altura:

float area; int altura, base;

cout<<“Dame la base y la altura”;

cin>>base>>altura;

Comparación respecto a operaciones de Entrada y salida.

Se pueden pasar datos E/S por archivos (se verán más adelante) o por argumentos de la función principal.

En Java dichos argumentos se guardan en la variable args del método main.

Comparación respecto a operaciones de Entrada y salida.

En C++ se debe declarar la función main con las variables int argc, char \*argv[]. En donde la primera indica el número de argumentos y la segunda los contiene.

El primer argumento en C++ es el nombre del programa en Java no ocurre esto.

Comparación respecto a operaciones de Entrada y salida.

En Java:

public static void main(String args[]) {

System.out.println(“Argumentos:”+ args.length);

JOptionPane.showMessageDialog(null, “Argumento 1:”+args[0]);

}

Comparación respecto a operaciones de Entrada y salida.

En C++:

int main(int argc, char \*argv[]){

if(argc!=2)

exit(-1);

cout<<“Argumento 1: ”<<argv[0];

return 0;

}

Comparación respecto a la definición, creación y uso de clases y objetos.

En Java todo debe de estar dentro de una clase, en C++ no es necesario.

En Java los archivos se deben llamar como se nombró a la clase, en C++ no.

Comparación respecto a la definición, creación y uso de clases y objetos.

En Java dentro de cada clase se deben incluir la definición de sus métodos y atributos, en C++ no es necesario.

En C++ existe la herencia múltiple, en Java no. En C++ las clases heredan con el operador “:” en Java con “extends”

Comparación respecto a la definición, creación y uso de clases y objetos.

Las clases para poderse utilizar se necesita de la creación de objetos, lo cual se hace a través del operador new en Java, en C++ no es necesario sólo que se pida memoria dinámica para el objeto.

El uso es el mismo, con el operador “.” se hacen a los miembros de la clase.

Comparación respecto a la definición y uso de métodos y parametrización.

Para la definición de atributos y métodos Java dispone de modificadores de alcance para cada elemento individual: public, private y protected. En C++ se utilizan de forma global siendo privado todo lo que está dentro de la clase. A esto en POO se le llama encapsulación.

1. ¿Por qué cree que los paradigmas de programación han evolucionado hasta la orientación a objetos?

Porque es una forma más sencilla de escribir, mantener y reutilizar módulos y programas, aparte de solventar problemas que planteaban otros paradigmas, como el imperativo.

1. Investigue en Internet cómo se indica que una clase hereda de otra en el lenguaje de programación Java.

Se indica con la palabra clave extends que indicara que una clase heredara de otra clase.

1. Realice un cuadro indicando qué lenguajes de alto nivel permiten herencia simple y herencia múltiple.

Herencia multiple: C++, Centura SQL Windows, CLOS, Eiffel, Object REXX, Perl y Python.

Herencia simple: Smalltalk, Java, Ada y C#

Conclusión: aunque sean similares los lenguajes de programación hay bastantes diferencias entre ellos, como en el caso aquí expuesto de java y C++, el paradigma programación orientada a objetos, es bastante importante ya que es algo muy práctico y se puede ver lo que podría hacer un objeto sin necesidad de crearlo, las herencias se transmiten de forma sencilla y clara ya que solamente hay que poner una sola palabra, de las dos herencias, la mejor es la multiple ya que permite más de un objeto, digamos, padre.

Bibliografía

1. [Comparación lenguaje java y C++](https://slideplayer.es/slide/3144650/)
2. [Paradigma programación orientado a objetos](http://aprendeenlinea.udea.edu.co/lms/men_udea/mod/page/view.php?id=19537#:~:text=El%20paradigma%20orientado%20a%20objetos%20(OO)%20define%20los%20programas%20en,t%C3%A9rminos%20de%20comunidades%20de%20objetos.&text=Es%20en%20este%20modo%20de,est%C3%A1n%20separados%20y%20sin%20relaci%C3%B3n.)
3. [Herencia](https://javadesdecero.es/poo/herencia-java-tipos-ejemplos/)
4. [Herencia múltiple](https://es.wikipedia.org/wiki/Herencia_m%C3%BAltiple) y [herencia simple](https://es.wikipedia.org/wiki/Herencia_(inform%C3%A1tica))