|  |  |  |
| --- | --- | --- |
| Taller 1 Programación Orientada a Objetos | febrero 13  2017 | |
| Profesor: Carlos Londoño Loaiza | | Elkin Ramírez Jiménez |

Cotecnova

**Índice:**

Contenido

[**1)** **Definir los conceptos fundamentales involucrados en el proceso de resolución de un problema a través de una computadora** 2](#_Toc474755673)

[2) Definir programa, compilador e interprete. 3](#_Toc474755674)

[**3)** **Identificar los antecedentes de la programación orientada a objetos** 3](#_Toc474755675)

[**4)** **Definir los conceptos involucrados en programación orientada a objetos.** 4](#_Toc474755676)

[**5)** **Identificar las etapas para la resolución de un problema bajo un enfoque orientado a objetos.** 6](#_Toc474755677)

[**6)** **Identificar el tipo de datos que se emplean en la creación de un programa.** 6](#_Toc474755678)

[**7)** **Definir los conceptos de variables, expresiones y constantes.** 6](#_Toc474755679)

[**8)** **Describir el procedimiento para declarar los formato de las sentencias de asignación de entrada y salida.** 7](#_Toc474755680)

[**9)** **Que es UML. Y que procesos se puede realizar con este lenguaje.** 7](#_Toc474755681)

[10) Que es java. Porque es importante para la industria del desarrollo del software y en qué proyectos se usan. 8](#_Toc474755682)

[Cibergrafias y bibliografías. 9](#_Toc474755683)

1. **Definir los conceptos fundamentales involucrados en el proceso de resolución de un problema a través de una computadora**

Respuesta.

La creación de un programa de computadora significa más que el simple [trabajo](http://www.monografias.com/trabajos34/el-trabajo/el-trabajo.shtml) mecánico del [hombre](http://www.monografias.com/trabajos15/fundamento-ontologico/fundamento-ontologico.shtml) y la máquina. Constituye la culminación de todo un proceso complejo, que involucra a los dos entes, aportando cada uno su respectiva parte en la realización de la tarea.

Esta tarea ha sido sistematizada por los especialistas del área, buscando optimizar la [eficiencia](http://www.monografias.com/trabajos11/veref/veref.shtml) de los programas producidos. La sistematización comprende un proceso que consta de siete pasos. Estos son:

Definición del problema.

Selección del [método](http://www.monografias.com/trabajos11/metods/metods.shtml) de solución.

Creación del [algoritmo](http://www.monografias.com/trabajos15/algoritmos/algoritmos.shtml).

Programación del algoritmo.

Depuración y [documentación](http://www.monografias.com/trabajos11/ladocont/ladocont.shtml) del programa.

Validación de la solución.

Instalación, [producción](http://www.monografias.com/trabajos16/estrategia-produccion/estrategia-produccion.shtml) y [mantenimiento](http://www.monografias.com/trabajos15/mantenimiento-industrial/mantenimiento-industrial.shtml) del programa.

Debemos enfocar la definición del problema, a través de las siguientes caracterizaciones:

El [modelo](http://www.monografias.com/trabajos/adolmodin/adolmodin.shtml) de situación típica que se estudiará.

Las limitaciones impuestas.

Los [datos](http://www.monografias.com/trabajos11/basda/basda.shtml) con que se cuenta.

La información que se espera obtener.

Las metas u [objetivos](http://www.monografias.com/trabajos16/objetivos-educacion/objetivos-educacion.shtml) a lograr al resolver el problema (este aspecto es sumamente importante).

Las alternativas disponibles para lograrlas.

El balance de los criterios involucrados en el problema ([velocidad](http://www.monografias.com/trabajos13/cinemat/cinemat2.shtml#TEORICO), precisión, confiabilidad de las respuestas, facilidad de uso, ..., entre otros).

# 2) Definir programa, compilador e interprete.

Que es un programa.

Es un conjunto de instrucciones escritas para realizar un trabajo ya bien definido en un computador

Que es un compilador.

Es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje diferente. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (bytecode), o simplemente texto.

1. **Identificar los antecedentes de la programación orientada a objetos**

Los conceptos de la POO tienen origen en [Simula 67](https://es.wikipedia.org/wiki/Simula), un lenguaje diseñado para hacer simulaciones, creado por [Ole-Johan Dahl](https://es.wikipedia.org/wiki/Ole-Johan_Dahl) y [Kristen Nygaard](https://es.wikipedia.org/wiki/Kristen_Nygaard), del Centro de Cómputo Noruego en [Oslo](https://es.wikipedia.org/wiki/Oslo). En este centro se trabajaba en simulaciones de naves, que fueron confundidas por la explosión combinatoria de cómo las diversas cualidades de diferentes naves podían afectar unas a las otras. La idea surgió al agrupar los diversos tipos de naves en diversas clases de objetos, siendo responsable cada clase de objetos de definir sus "propios" datos y comportamientos. Fueron refinados más tarde en [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk" \o "Smalltalk), desarrollado en Simula en [Xerox PARC](https://es.wikipedia.org/wiki/Xerox_PARC) (cuya primera versión fue escrita sobre [Basic](https://es.wikipedia.org/wiki/Basic)) pero diseñado para ser un sistema completamente dinámico en el cual los objetos se podrían crear y modificar "sobre la marcha" (en tiempo de ejecución) en lugar de tener un sistema basado en programas estáticos.

La POO se fue convirtiendo en el estilo de programación dominante a mediados de los [años 1980](https://es.wikipedia.org/wiki/A%C3%B1os_1980), en gran parte debido a la influencia de [C++](https://es.wikipedia.org/wiki/C%2B%2B), una extensión del [lenguaje de programación C](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C). Su dominación fue consolidada gracias al auge de las [interfaces gráficas de usuario](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario), para las cuales la POO está particularmente bien adaptada. En este caso, se habla también de [**programación dirigida por eventos**](https://es.wikipedia.org/wiki/Programaci%C3%B3n_dirigida_por_eventos).

características de orientación a objetos fueron agregadas a muchos lenguajes existentes durante ese tiempo, incluyendo [Ada](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Ada), [BASIC](https://es.wikipedia.org/wiki/BASIC), [Lisp](https://es.wikipedia.org/wiki/Lisp" \o "Lisp) más [Pascal](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Pascal), entre otros. La adición de estas características a los lenguajes que no fueron diseñados inicialmente para ellas condujo a menudo a problemas de compatibilidad y en la capacidad de mantenimiento del código. Los lenguajes orientados a objetos "puros", por su parte, carecían de las características de las cuales muchos programadores habían venido a depender. Para saltar este obstáculo, se hicieron muchas tentativas para crear nuevos lenguajes basados en métodos orientados a objetos, pero permitiendo algunas características imperativas de maneras "seguras". El [lenguaje de programación Eiffel](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Eiffel) de Bertrand Meyer fue un temprano y moderadamente acertado lenguaje con esos objetivos, pero ahora ha sido esencialmente reemplazado por [Java](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)), en gran parte debido a la aparición de [Internet](https://es.wikipedia.org/wiki/Internet) y a la implementación de la [máquina virtual Java](https://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java) en la mayoría de [navegadores web](https://es.wikipedia.org/wiki/Navegadores_web). [PHP](https://es.wikipedia.org/wiki/PHP) en su versión 5 se ha modificado; soporta una orientación completa a objetos, cumpliendo todas las características propias de la orientación a objetos.

1. **Definir los conceptos involucrados en programación orientada a objetos.**

La POO es una forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

[**Clase**](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica))

Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ella.

[**Herencia**](https://es.wikipedia.org/wiki/Herencia_(inform%C3%A1tica))

Por ejemplo, herencia de la clase C a la clase D, es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (*private*) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. En el caso de los componentes registrados como "protegidos" (*protected*) también se heredan, pero solo para esa clase, no para futuras clases heredadas. Esto es así para mantener hegemónico el ideal de POO.

[**Objeto**](https://es.wikipedia.org/wiki/Objetos_(programaci%C3%B3n_orientada_a_objetos))

Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa).

[Método](https://es.wikipedia.org/wiki/M%C3%A9todo_(programaci%C3%B3n_orientada_a_objetos))

Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

**Evento**

Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

**Atributos**

Características que tiene la clase.

**Mensaje**

Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

**Propiedad o atributo**

Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

**Estado interno**

Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.

**Componentes de un objeto**

Atributos, identidad, relaciones y métodos.

**Identificación de un objeto**

Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

En comparación con un lenguaje imperativo, una "variable" no es más que un contenedor interno del atributo del objeto o de un estado interno, así como la "función" es un procedimiento interno del método del objeto.

1. **Identificar las etapas para la resolución de un problema bajo un enfoque orientado a objetos.**

Las etapas para la resolución de un problema bajo con un enfoque poo podría definirse de la siguiente manera

* Definición del problema o dominio del problema
* Análisis orientado a objetos
* diseño orientado a objetos.
* Programación.

1. **Identificar el tipo de datos que se emplean en la creación de un programa.**

Los datos que se crean para la creación y desarrollo de un programa son los siguientes.

Definición del problema, esto con el fin de poder identificar las posibles soluciones.

Las variables. Lugar donde se recolectan los datos del usuario para procesarlos debidamente

Entidades atributos y métodos.

En los atributos describiríamos las características de los objetos.

En los métodos podríamos realizar en trabajo en específico.

Como por ejemplo tipo de acceso tipo de retorno nombre del método o parámetros.

También encontramos los tipos de acceso como lo pueden ser de tipo público o privado.

1. **Definir los conceptos de variables, expresiones y constantes.**

Constantes.

Una constante es un dato cuyo valor no se puede cambiar durante la ejecución del programa, ella recibe un valor en el momento de la compilación del programa y permanecerá inalterado durante la ejecución del programa.

Variables.

Es el nombre asociado a un elemento de datos que está situado en posición contiguas a la memoria principal, y su valor puede cambiar durante la ejecución de un programa y toda variable pertenece un tipo de dato.

Expresión.

Podemos definirlo como la combinación de variables y constantes o funciones.

1. **Describir el procedimiento para declarar los formato de las sentencias de asignación de entrada y salida.**

La sintaxis para la entrada o salida son los siguientes.

En java la entrada o salida se lee y se escribe en flujos (streams). La fuente básica de entrada y salida es el teclado y la fuente de salida aes la pantalla.

La clase system define dos referencias a objetos estatic para la gestión de entrad ay salida por consola.

Systen.inc para la entrada por teclado.

System.out para la salida pro pantalla.

1. **Que es UML. Y que procesos se puede realizar con este lenguaje.**

El lenguaje del que estoy hablando se llama UML unifed modeling lenguaje (lenguaje de modelado unificado de modelado) . es un lenguaje como pscal c++ o c sharp.

Como sucede con otros lenguajes el uml fue inventado por necesidad en uml se usan símbolos para trasmitir significados.

Uml es una definición oficial del lenguaje pictórico con símbolos y relación comunes que tienen un significado común. Si todos los participantes hablan UML..

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el [Proceso Unificado Racional](https://es.wikipedia.org/wiki/Proceso_Unificado_Racional), *Rational Unified Process* o [RUP](https://es.wikipedia.org/wiki/RUP)), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la [programación estructurada](https://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada), pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que programación estructurada es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML solo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

# Que es java. Porque es importante para la industria del desarrollo del software y en qué proyectos se usan.

**Que es java.**

Se conoce con o un lenguaje de programación y un entorno para la ejecución de programas escritos en lenguaje java, al contrario que los compiladores tradicionales, que convierten el código fuente en instrucciones a n nivel de máquina, el compiladora de java traduce el código fuente java a instrucciones que son interpretadas por la máquina virtual de java. (Jvm) . a diferencia de otros lenguajes como C y C++ en los que está inspirado java. Java es un lenguaje interpretado.

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes

**Porque es importante java para la industria del desarrollo.**

La importancia de Java radica en que se trata de un lenguaje multiplataforma y que por lo tanto se pueden desarrollar aplicaciones para todos los sistemas operativos ya que solo es necesario instalar la máquina virtual de Java (JVM) para poder ejecutar aplicaciones.

## Cibergrafias y bibliografías.

**Pregunta 1**

Sito de interés

<http://www.monografias.com/trabajos105/uso-computadoras-solucion-problemas/uso-computadoras-solucion-problemas.shtml#ixzz4YVghRzLv>

**Pregunta 2**

Cibergrafia

<https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos>

**Pregunta 3**

www.wikipedia.com

**Pregunta 9**

BIBLIOGRAFIA.

Manual de UML de Paul Kimmel.

Editorial. Mc Graw Gil

**Pregunta 10**

Bibliografía.

Programación en java 2

Luis joyanes Aguilar

Ignacio zahonero Martínez

Editorial

MC Graw Gil.