Unidad I

Fundamentos de Programación Orientada a Objetos

1.1 Evolución de la programación.

Es la primera metodología con la que se trabajo para escribir código, y consistía en capturar cada instrucción línea por línea, mismas que se ejecutaban en forma continua.
La programación lineal no tenia estructuras bien definidas y el control de flujo de datos a través de cientos o miles de líneas de código resultaba casi imposible. Con frecuencia los programadores utilizaban una instrucción llamada **goto** para saltar a otras partes de un programa. Una declaración goto identificaba una línea diferente del programa a la cual saltaba el control. El problema que se presento con la instrucción goto era identificar como procede el flujo del control del programa después del salto. El control continuara después de la línea que salto o regresa al lugar de donde salto. Ejemplo:

Desventajas:

El problema que presento la programación lineal, es que no tenía uncontrol sobre el flujo de los programas.

Programación estructurada.

La **programación estructurada** es una forma de escribir programas de ordenador(programación de computadora) de forma **clara .**

Para ello utiliza únicamente tres estructuras básicas: secuencia, selección e iteración; siendo innecesario el uso dela instrucción o instrucciones de transferencia incondicional (GOTO, EXITFUNCTION, EXIT SUB o múltiples RETURN).Hoy en día las aplicaciones informáticas son mucho más ambiciosas que lasnecesidades de programación existentes en los años 1960, principalmente debido a las aplicaciones gráficas, por lo que las técnicas de programación estructurada no son suficientes. Ello ha llevado al desarrollo de nuevas técnicas, tales como la programación orientada a objetos y el desarrollo de entornos de programación que facilitan la programación de grandes aplicaciones.

1.2 Conceptos fundamentales de la Programación Orientada a Objetos.

La programación orientada a objetos, ha tomado las mejores ideas de la programación estructurada y los ha combinado con varios conceptos nuevos y potentes que incitan a contemplar las tareas de programación desde un nuevo punto de vista. La programación orientada a objetos, permite descomponer mas fácilmente un problema en subgrupos de partes relacionadas del problema. Entonces, utilizando el lenguaje se pueden traducir estos subgrupos a unidades autocontenidas llamadas objetos.

El término Programación Orientada a Objetos (POO), hoy en día ampliamente utilizado, es difícil de definir, ya que no es un concepto nuevo, sino que ha sido el desarrollo de técnicas de programación desde principios de la década de los setenta, aunque sea en la década de los

noventa cuando ha aumentado su difusión, uso y popularidad. No obstante, se puede definir POO como una técnica o estilo de programación que utiliza objetos como bloque esencial de construcción.

Un objeto es una unidad que contiene datos y las funciones que operan sobre esos datos. A los elementos de un objeto se les conoce como miembros; las funciones que operan sobre los objetos se denominan métodos y los datos se denominan miembros datos

http://dis.um.es/~jfernand/0506/dai/poo.pdf

1.3 Lenguajes orientados a objetos

Se le llama así a cualquier lenguaje de programación que implemente los conceptos definidos por la programación orientada a objetos.

Cabe notar que los conceptos definidos en la programación orientada a objetos no son una condición sino que son para definir que un lenguaje es orientado a objetos. Existen conceptos que pueden estar ausentes en un lenguaje dado y sin embargo, no invalidar su definición como lenguaje orientado a objetos.

Quizás las condiciones mínimas necesarias las provee el formalismo que modeliza mejor las propiedades de un sistema orientado a objetos: los tipos de datos abstractos.

Siguiendo esa idea, cualquier lenguaje que permita la definición de tipos de datos, de operaciones nuevas sobre esos tipos de datos, y de instanciar el tipo de datos podría ser considerado orientado a objetos.

Esta definición concuerda incluso con ciertos ejemplos prácticos, que no son considerados dentro de la programación orientada a objetos, pero que podrían serlo. Por ejemplo, la programación de interfaces gráficas de usuario para los sistemas X-Window utilizando infraestructuras de funciones y APIs como Motif, Xview y Xlib, son realizadas usualmente en lenguaje C, pero organizando el código en una manera que "parecen objetos" (los Widgets).

Ejemplos de lenguajes orientados a objeto

- C++
- Objective C
- Java
- Smalltalk
- eC (Ecere C)
- Eiffel

- Lexico (en castellano)
- Ruby
- Python
- OCAML
- Object Pascal
- CLIPS
- Visual .net
- Actionscript
- COBOL
- Perl
- C#
- Visual Basic.NET
- PHP
- Simula
- Delphi
- PowerBuilder
- Maya
- Corel draw

1.4 Relaciones entre clases y objetos.

1.5 Papel de clases y objetos en el análisis y el diseño.

La programación orientada a objetos es una “filosofía”, un modelo de programación, con su teoría y su metodología, que conviene conocer y estudiar antes de nada. Un lenguaje orientado a objetos es un lenguaje de programación que permite el diseño de aplicaciones orientadas a objetos. Dicho esto, lo normal es que toda persona que vaya a desarrollar aplicaciones orientadas a objetos aprenda primero la “filosofía” (o adquiera la forma de pensar) y después el lenguaje, porque “filosofía” sólo hay una y lenguajes muchos. En este documento veremos brevemente los conceptos básicos de la programación orientada a objetos desde un punto de vista global, sin particularizar para ningún lenguaje de programación específico.