Universidad Autónoma de Ciudad Juárez

División Multidisciplinaria Ciudad Universitaria

Cuadro comparativo de lenguajes de Programación Orientada a Objetos

Programación II

Docente: Alan Ponce

Alumno Lizbeth Aneliz Santillán Tarango 163721

Licenciatura en Ingeniería de Sistemas Computacionales

08 de Marzo de 2019

Introducción

1- Definición de lenguaje procedural.

Se trata de un estilo de programación basado en estructurar el código de un programa en componentes, que reciben el nombre de procedimientos, subrutinas o funciones.

http://www.atc.uniovi.es/telematica/2ac/Transparencias/T02-Programacion-Procedural.pdf

En los lenguajes tradicionales o procedurales, es la aplicación quien controla qué porciones de código se ejecuta, y la secuencia en que este se ejecuta. La ejecución de la aplicación se inicia con la primera línea de código, y sigue una ruta predefinida a través de la aplicación, llamando procedimientos según sea necesario.

Los lenguajes procedurales están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados. Algunos ejemplos son: FORTRAN, PASCAL, C, ADA, ALGOL, …

En general, un lenguaje procedural ofrece al programador conceptos que se traducen de forma natural al modelo de la máquina.

Con un lenguaje procedural el usuario (normalmente será un programador) especifica qué datos se necesitan y cómo obtenerlos. Esto quiere decir que el usuario debe especificar todas las operaciones de acceso a datos llamando a los procedimientos necesarios para obtener la información requerida. Estos lenguajes acceden a un registro, lo procesan y basándose en los resultados obtenidos, acceden a otro registro, que también deben procesar. Así se va accediendo a registros y se van procesando hasta que se obtienen los datos deseados. Las sentencias de un lenguaje procedural deben estar embebidas en un lenguaje de alto

nivel, ya que se necesitan sus estructuras (bucles, condicionales, etc.) para obtener y procesar cada registro individual.
https://www.monografias.com/trabajos38/tipos-lenguajes-programacion/tipos-lenguajes-programacion2.shtml

Indicación de lenguajes de programación y ejemplos de código.

Algunos lenguajes de programación procedurales que se pueden mencionar son:

BASIC

C

COBOL

FORTRAN

Ada

Pascal

https://www.monografias.com/trabajos38/tipos-lenguajes-programacion/tipos-lenguajes-programacion2.shtml

Ejemplos de código.

Lenguaje: C

Ejemplo: Área de un triángulo.

Código fuente del programa:

```
#include <stdio.h>

int main()
{
   float altura, area, base;

   printf( "Introduzca base: " );
   scanf( "%f", &base );
   printf( "Introduzca altura: " );
   scanf( "%f", &altura );

   area = base * altura / 2;

   printf( "El area del triangulo es: %f", area );

   return 0;
}
```

http://www.carlospes.com/ejercicios_de_lenguaje_c/entrada_y_salida_estandar_001_solucion.php

Lenguaje: COBOL

Ejemplo: Un hola mundo.

```
 000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID.     HOLAMUNDO.
000300
000400*
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. RM-COBOL.
000800 OBJECT-COMPUTER. RM-COBOL.
000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100100
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100400     DISPLAY " " LINE 1 POSITION 1 ERASE EOS.
100500     DISPLAY "Hola mundo" LINE 15 POSITION 10.
100600     STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800     EXIT.
```

http://www.holamundo.es/lenguaje/cobol/

Lenguaje: FORTRAN

Ejemplo: Calcular el factorial de un número.

```
IMPLICIT NONE
REAL::J,N
INTEGER::P
WRITE(*,*)"NUMERO AL CUAL DESEA CALCULAR SU FACTORIAL"
READ(*,*)N
P=1
DO J=0,N-1,1
P=P*(N-J)
END DO
WRITE(*,*)"EL FACTORIAL ES:",P
END PROGRAM
```

http://apuntesdeingenieriaquimica.blogspot.com/2015/05/ejemplo-de-codigo-fortran-para-calcular_51.html

Lenguaje: Ada

Ejemplo:

PROGRAMA EJEMPLO

```
with Text_IO; -- especificación de contexto
procedure CUENTA is
   NUMERO_DE_ITERACIONES: Integer := 1;    -- definición de una variable
   package Int_IO is new Text_IO.Integer_IO(Integer); -- package local
                                                      --para enteros
begin
   Text_IO.PUT("N° de iteraciones:");
   Int_IO.GET(NUMERO_DE_ITERACIONES);
   Text_IO.NEW_LINE;
   for INDICE in 1..NUMERO_DE_ITERACIONES loop
      Text_IO.PUT("Iteración ");
      Int_IO.PUT(INDICE,4);
      Text_IO.NEW_LINE;
   end loop;
end CUENTA;
```

http://isa.uniovi.es/docencia/TiempoReal/Recursos/temas/Lenguaje_Ada.pdf

Lenguaje: Pascal

Ejemplo: Escribir un programa en Pascal que determine si un número leído desde el teclado es par o impar.

```
PROGRAM EJER34;
        USES CRT;

        VAR num:INTEGER;

BEGIN
        ClrScr;

        WRITE ('Introduzca un numero entero: ');       READLN (num);

        IF num = 0 THEN
           WRITE ('El numero introducido no es par ni impar, es 0')
        ELSE IF ((num mod 2 = 0)) THEN
           WRITE ('El numero introducido es par')
        ELSE
            WRITE ('El numero introducido es impar')

END.


PROGRAM EJER34;
        USES CRT;
        VAR num:INTEGER;
BEGIN

     ClrScr;

     WRITE('Introduzca un numero: ');
     READLN(num);

     IF (num mod 2 = 0) THEN
        WRITE('NUMERO PAR')
     ELSE
        WRITE('NUMERO IMPAR');
END.
```

http://www.cartagena99.com/recursos/programacion/ejercicios/ejercicios_pascal.pdf

1- Definición de lenguaje orientada a objetos

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a cómo expresaremos las cosas en la vida real que otros tipos de programación.

Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación.
https://desarrolloweb.com/articulos/499.php

Lenguajes de programación y reseña de creación y evolución (C++, Java, Python, C#)

C++

En 1980 surge C++ de la mano de Bjarne Stroustrup (también de Laboratorios Bell de AT&T). Diseña este lenguaje con el objetivo de añadir a C nuevas características: clases y funciones virtuales (de SIMULA67), tipos genéricos y expresiones (de ADA), la posibilidad de declarar variables en cualquier punto del programa (de ALGOL68), y sobre todo, un auténtico motor de objetos con herencia múltiple que permite combinar la programación imperativa de C con la programación orientada a objetos.
El siguiente hecho fundamental en la evolución de C++ es sin duda la incorporación de la librería STL años más tarde, obra de Alexander Stepanov y Adrew Koening. Esta librería de clases con contenedores y algoritmos genéricos proporciona a C++ una potencia única entre los lenguajes de alto nivel.

Debido al éxito del lenguaje, en 1990 se reúnen las organizaciones ANSI e ISO para definir un estándar que formalice el lenguaje. El proceso culmina en 1998 con la aprobación del ANSI C++.
https://jorgesaavedra.wordpress.com/2006/12/09/breve-historia-de-c-c-c/

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de los años 90´s.

En Diciembre de 1950 Patrick Naughton, ingeniero de Sun Microsystems, reclutó a varios colegas entre ellos James Gosling y Mike Sheridan para trabajar sobre un nuevo proyecto conocido como "El proyecto verde".
Con la ayuda de otros ingenieros, empezaron a trabajar en una pequeña oficina en Sand Hill Road en Menlo Park, California. Y así interrumpió todas las comunicaciones regulares con Sun y trabajó sin descanso durante 18 meses.

Para 1992, el equipo ya había desarrollado un sistema en un prototipo llamado Star7 (*7), dispositivo parecido a una PDA, cuyo nombre venía de la combinación de teclas del teléfono de la oficina del Proyecto Green que permitía a los usuarios responder al teléfono desde cualquier lugar.

Después de mostrar a Scott McNealy y Bill Joy los prototipos de bajo nivel del sistema, continúan con el desarrollo, incluyendo sistema operativo, Green OS; el lenguaje Oak, las librerías, alguna aplicación básica y el hardware, hasta que el 3 de septiembre de 1992 se termina el desarrollo y con ello el Proyecto Verde.
Patrick Naughton procedió a la construcción del lenguaje de programación Java que se accionaba con un browser prototipo. El 29 de septiembre de 1994 se termina el desarrollo del prototipo de HotJava. Cuando se hace la demostración a los ejecutivos de Sun, esta vez, se reconoce el potencial de Java y se acepta el proyecto.
En enero de 1995 Sun formá la empresa Java Soft para dedicarse al desarrollo de productos basados en la tecnologías Java, y así trabajar con terceras partes para crear aplicaciones, herramientas, sistemas de plataforma y servicios para aumentar las capacidades del lenguaje. Ese mismo mes aparece la versión 1.0 del JDK.
http://www.cad.com.mx/historia_del_lenguaje_java.htm

Python

La historia de Python como lenguaje de programación inicia a finales de los 80s y principios de los 90s con Guido Van Rossum, una historia de 29 años de desarrollo.

A Guido Van Rossum le gustaba mucho el grupo Monty Python, por esta razón escogió el nombre del lenguaje. Actualmente Van Rossum sigue ejerciendo el rol central decidiendo la dirección de Python.
En 1991, Van Rossum publicó el código de la versión 0.9.0 en alt.sources. En esta versión ya teníamos disponibles clases con herencias, manejo de excepciones, funciones y los tipos modulares.
Para 1994 se creó comp.lang.python, un foro de discusión de Python que marcó un hito en su popularidad y multiplicó su cantidad de usuarios.
Para este mismo año, Python llega a la versión 1.0 que incluyó herramientas de la programación funcional como lambda, reduce, filter y map.
Para Python 2.0 se incluyó la generación de listas, una de las características más importantes del lenguaje de programación funcional Haskell. Además incluyó un sistema de recolección de basura capaz de recolectar referencia cíclicas.
La última gran actualización de Python fue en 2008 con Python 3.0 con el PEP 3000, diseñado para rectificar fallas fundamentales en el diseño del lenguaje.
“Reduce feature duplication by removing old ways of doing things”
https://platzi.com/blog/historia-python/

C#

Durante el desarrollo del .NET Framework, sus bibliotecas de clases fueron escritas mediante una compilador de código administrado denominado Simple Managed C (SMC). En enero de 1999, Anders Hejlsberg, creador de lenguajes como Turbo Pascal y Delphi, formó un equipo para diseñar un nuevo lenguaje, cuyo nombre inicial fue Cool (C-like Object Oriented Language). Microsoft quiso mantener dicho nombre como definitivo pero por razones de registro de marca tuvo que echarse para atrás.
En julio de 2000, durante la Professional Developers Conference, el lenguaje había sido renombrado a C# y todas las librerías de clases y el runtime de ASP.NET fueron portadas a C#.

Desde el lanzamiento de C# 2.0 en noviembre de 2005, los lenguajes C# y lenguajes Java han evolucionado en trayectorias cada vez más divergentes, convirtiéndose en algo menos similares el uno del otro.
https://sparraguerra.wordpress.com/2015/05/18/un-poquito-de-historia-de-c/

2- Cuadro comparativo de estos dos paradigmas:

<table>
<tr><th>Paradigma Procedural</th><th>Paradigma Orientada a Objetos</th></tr>
<tr><td>Definición de procedimientos
Definición de tipos de datos
Chequeo de tipos en tiempo de compilación
Cambio de estado de variables
Pasos de ejecución de un proceso
Lenguajes: Fortran, Algol, C, Pascal

Ejemplo: PASCAL

http://www.dccia.ua.es/dccia/inf/asignaturas/LPP/2008-2009/traspas/clase02.pdf
<pre>type
tDimension = 1..100;
eMatriz(f,c: tDimension) = array [1..f,1..c] of real;

tRango = record
f,c: tDimension value 1;
end;

tpMatriz = ^eMatriz;

procedure EscribirMatriz(var m: tpMatriz);
var filas,col : integer;
begin
for filas := 1 to m^.f do begin
for col := 1 to m^.c do
write(m^[filas,col]:7:2);
writeln(resultado);
writeln(resultado)
end;
end;</pre></td><td>El paradigma orientado a objetos (OO) define los programas en términos de comunidades de objetos. Los objetos con características comunes se agrupan en clases (un concepto similar al de tipo abstracto de dato (TAD)).

Entre los lenguajes que soportan el paradigma OO están Smalltalk, C++, Delphi (Object Pascal), Java y C#.

http://aprendeenlinea.udea.edu.co/lms/men_udea/mod/page/view.php?id=19537

Definición de clases y herencia
Objetos como abstracción de datos y procedimientos
Polimorfismo y chequeo de tipos en tiempo de ejecución
Lenguajes: Modula, Smalltalk, Java, C#

Ejemplo: Java
http://www.dccia.ua.es/dccia/inf/asignaturas/LPP/2008-2009/traspas/clase02.pdf
<pre>public class Bicicleta {
  public int marcha;
  public int velocidad;

  public Bicicleta(int velocidadInicial, int marchaInicial) {
    marcha = marchaInicial;
    velocidad = velocidadInicial;
  }

  public void setMarcha(int nuevoValor) {
   marcha = nuevoValor;
  }

  public void frenar(int decremento) {
   velocidad -= decremento;
  }

  public void acelerar(int incremento) {
   velocidad += incremento;
  }
}</pre></td></tr>
</table>

Lenguajes de Programación Orientada a Objetos

En esta sección se deberá focalizar en las diferencias entre los diversos lenguajes de programación que soportan el paradigma de la programación orientada a objetos.

C++

Incluir código de cómo implementar:

Una clase (Ejemplo de clase automóvil)

```
class Automobile {
public:
 Automobile();
 void Input();
 void set_NumDoors( int doors );
 void Display();
 int get_NumDoors();
private:
string Make;
int NumDoors;
int NumCylinders;
int EngineSize;
};
```

http://profesores.elo.utfsm.cl/~agv/elo329/1s09/lectures/C++/ClasesC++.pdf

Herencia (Herencia con clase, personas)

```
// Clase base Persona:
class Persona {
  public:
   Persona(char *n, int e);
   const char *LeerNombre(char *n) const;
```

```
   int LeerEdad() const;
   void CambiarNombre(const char *n);
   void CambiarEdad(int e);

  protected:
   char nombre[40];
   int edad;
};

// Clase derivada Empleado:
class Empleado : public Persona {
  public:
   Empleado(char *n, int e, float s);
   float LeerSalario() const;
   void CambiarSalario(const float s);

  protected:
   float salarioAnual;
};
```

http://c.conclase.net/curso/index.php?cap=036

Polimorfismo

```
// funcion principal
void main(){
      clrscr();
      triangulo t(5.2,6.5,7.1);
      virtualViaPointer (& t);
      virtualViaReference (t);

      cuadrado c(8.7);
      virtualViaPointer (&c);
      virtualViaReference (c);
```

```
    getch();
    clrscr();

    cubo cub(8.3);
    virtualViaPointer (&cub);
    virtualViaReference (cub);

    paralelepipedo p(4.5,6.7,9.2);
    virtualViaPointer (&p);
    virtualViaReference (p);

    getch();
}
```

http://www.programacionenc.net/index.php?option=com_content&view=article&id=70:polimorfismo&catid=37:programacion-cc&Itemid=55

Encapsulamiento

```
class TObjGraf {

public:
  int       X;        // Propiedades
  int       Y;
  TColor    Color;
  TPaintBox *PaintBox;

  void Mostrar (void); // Métodos
};
```

https://elvex.ugr.es/decsai/builder/intro/5.html

Java

Incluir código de cómo implementar:

- **Una clase**
- **Herencia**
- **Polimorfismo**
- **Encapsulamiento**

Python

Incluir código de cómo implementar:

- **Una clase**
- **Herencia**
- **Polimorfismo**
- **Encapsulamiento**

C#

Incluir código de cómo implementar:

- **Una clase**
- **Herencia**
- **Polimorfismo**
- **Encapsulamiento**

Conclusiones.

Dentro de un análisis comparativo, el lenguaje procedural al final son las funciones y procedimientos, una forma de escribir un programa en manera clara. Cualquiera puede hacer un programa con escasos conocimientos en PP.
En POO se trata más a objetos que se pueden incluir en clases, etc.