

Examen Parcial SOLUCIONARIO

ALUMNO: MERCADO SINCHÉ JULIO ABEL

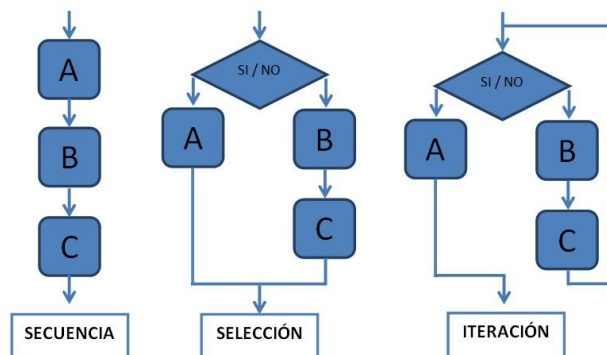
CODIGO: 18190312

GRUPO: G1

1. Describir como mínimo 3 paradigmas de programación y dar sus características.

PARADIGMA ESTRUCTURADO

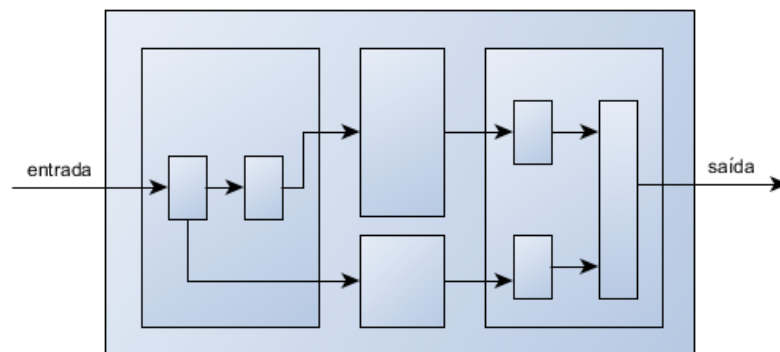
- El inicio del archivo es el inicio de la programación y el final del archivo es el final de la programación; en otras palabras, es secuencial.
- Se pueden utilizar instrucciones de control como las condicionales (if)
- Se pueden utilizar instrucciones de iteración (Bucles)



PARADIGMA PROCEDIMENTAL

El paradigma procedimental se refiere a un conjunto de instrucciones o sentencias. Por lo general pequeños programas, con unas pocas entradas un cálculo específico y una salida esperada. En el caso de pequeños programas, estos principios de organización son eficientes.

- Uso de subprogramación
 - Agrupación de código que permite la creación de acciones complejas
 - Asignación de un nombre para estas acciones complejas
 - Llamada a estas acciones complejas desde cualquier punto del programa
- Estas acciones complejas se denominan procedimientos, subrutinas y funciones

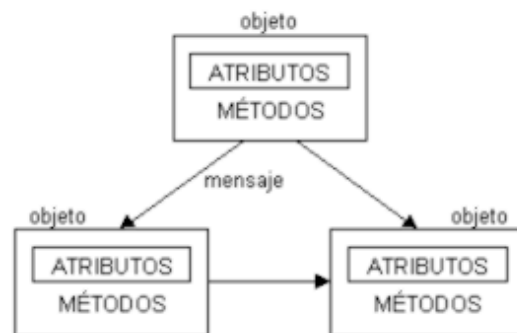


PARADIGMA ORIENTADA A OBJETOS

El paradigma orientado a objetos es de los paradigmas más usados para desarrollo de software en la industria, esto se debe a que los objetos y las clases son la base de muchos proyectos.

Algunas de las cosas que debemos saber de este paradigma son:

- Los objetos se crean y se destruyen
- Las clases son lo que utilizamos para definir los objetos
- Las clases pueden contener atributos y métodos.



2.Cuál es la diferencia entre i++ e ++i, dar un ejemplo en código.

La diferencia está en el valor de retorno de las dos expresiones

POST INCREMENTO

i++, El valor de la expresión es el valor de la variable antes que el nuevo valor sea almacenado.

Ejemplo:

```
int i = 1;
```

```
System.out.println(i++); // imprime 1.
```

```
System.out.println(i); // imprime 2.
```

PRE INCREMENTO

++i, El valor de la expresión es el valor de la variable después que el nuevo valor es almacenado.

Ejemplo:

```
int i = 1;
```

```
System.out.println(++i); // imprime 2.
```

```
System.out.println(i); // imprime 2.
```

Ambas incrementan la variable i en 1. La diferencia es que la expresión ++i devuelve el valor de la variable incrementada en cambio la expresión i++ devuelve el valor original de variable antes de ser incrementada.

3. Completar el cuadro con la información de PRIORIDAD, siendo 1 más prioritario que 5.

OPERADOR	PRIORIDAD	OPERADOR	PRIORIDAD
/	4	(expr)	1
--var	3	+expr	3
*	4	+	5
%	4	-	5
var--	2	&&	5

4. Responder las siguientes preguntas y dar ejemplos.

a) ¿Qué significa casting en programación?

El casting es un procedimiento para transformar una variable primitiva de un tipo a otro. También se utiliza para transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas. En este caso nos centraremos en el primer tipo de casting.

```
float n = 4.5f;
int num;

num = (int) n;
```

b) ¿Qué es una función y un procedimiento?

Funciones

Las funciones son un conjunto de líneas de código (instrucciones), encapsulados en un bloque, usualmente reciben parámetros, cuyos valores utilizan para efectuar operaciones y adicionalmente retornan un valor con la ejecución return.

Ejemplo: función restar

```
private int restar (int numero1, int numero)
{
    Int resta=numero1-numero2
    return resta;
}
```

Procedimientos:

Los procedimientos son básicamente un conjunto de instrucciones que se ejecutan sin retornar ningún valor. En el contexto de Java un procedimiento es básicamente un método cuyo tipo de retorno es void que no nos obliga a utilizar una sentencia return.

Ejemplo: procedimiento limpiar

```
private void limpiar ()
{
    txtNumero1.setText(null);
}
```

c) ¿Qué quiere decir sobrecarga de operadores?

La sobrecarga de operadores es la capacidad para transformar los operadores de un lenguaje como por ejemplo el +, -, etc, cuando se dice transformar se refiere a que los operandos que entran en juego no tienen que ser los que admite el lenguaje por defecto. Mediante esta técnica podemos sumar dos objetos creados por nosotros o un objeto y un entero, en vez de limitarnos a sumar números enteros o reales, por ejemplo.

Java no es compatible con la sobrecarga del operador definida por el usuario. El único aspecto de Java que se acerca a la sobrecarga del operador "personalizado" es el manejo de + para cadenas, que da como resultado la concatenación de constantes en tiempo de compilación o la concatenación en tiempo de ejecución utilizando StringBuilder/StringBuffer. No puedes definir tus propios operadores que actúen de la misma manera.

Hay muchos otros ejemplos de sobrecarga de operador en Java. Por ejemplo, '&', '|' y '^' son sobrecargas para 'boolean' y tipos integrales. Y, de hecho, los operadores aritméticos y relacionales están sobrecargados por varios tipos numéricos.