

EXAMEN PARCIAL DE P.O.O.-GRUPO 01(20/05/2019)

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

Paradigma orientado a objetos:

- Paradigma de programación que surgió en 70's.
- Basado en Objetos que son entidades con estado, método e identidad.
- Usa en técnicas como herencia, cohesión abstracción, polimorfismo, acoplamiento y encapsulamiento.
- Usa objetos como elementos fundamentales para la solución. Fomenta la reutilización de código.

Paradigma declarativo:

- Es un paradigma de programación que se basa en un conjunto de condiciones proposiciones, ecuaciones que describen el problema y detallan su solución.
- Los lenguajes declarativos son razonados matemáticamente.
- Existe varios tipos de lenguajes declarativos Lógicos, funcionales.

Paradigma imperativo:

- Está basado en términos del estado del programa y sentencias que cambian el estado de programa.
- Es un paradigma más cercano a la arquitectura física de la máquina.
- Almacena los datos en variables.
- Se basa en la ejecución de pasos elementales.
- Trabaja con sentencias de selección (if, match), repetición (for, while) y de secuencia

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

POST INCREMENTO:

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

Ejemplo:

```
int x = 1;  
System.out.println(x++); // imprime 1.  
System.out.println(x); // imprime 2.
```

PRE INCREMENTO:

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

Ejemplo:

```
int x = 1;  
System.out.println(++x); // imprime 2.  
System.out.println(x); // imprime 2.
```

- 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 un casting en programación?

Conversión entre tipos primitivos (**casting**)

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.

Dentro de este casting de variables primitivas se distinguen dos clases: **implícito y explícito**.

b. ¿Qué es una función y un procedimiento?

Función:

El concepto de función en programación se fundamenta en el concepto de función matemática.

Una función, desde el punto de vista de la programación, se define como un proceso que recibe valores de entrada (llamados parámetros) y el cual retorna un valor resultado. Adicionalmente, las funciones son subprogramas dentro de un programa, que se pueden invocar (ejecutar) desde cualquier parte del programa, es decir, desde otra función, desde la misma función o desde el programa principal, cuantas veces sea necesario.

Las funciones se usan cuando existen dos o más porciones de algoritmo dentro de un programa que son iguales o muy similares, por ejemplo, en un algoritmo se puede emplear varias veces una porción de algoritmo que eleva a una potencia dada un número real. De esta manera conviene definir una función que al ser invocada ejecute dicho código, y en el lugar donde estaba la porción de algoritmo original, se hace un llamado (ejecución) de la función creada.

En el pseudocódigo una función se declara de la siguiente manera:

```
funcion <nombre> ( param1 : tipo1 , ..., paramn : tipon ) : tipo
variables
    <declaraciones>
inicio
```

Donde,

- <nombre>: representa el nombre de la función
- param_i: representa el parámetro i-ésimo de la función.
- tipo_i: representa el tipo del i-ésimo parámetro de la función.

Procedimientos:

En muchos casos existen porciones de código similares que no calculan un valor si no que por ejemplo, presentan información al usuario, leen una colección de datos o calculan más de un valor. Como una función debe retornar un único valor² este tipo de porciones de código no se podrían codificar como funciones. Para superar este inconveniente se creó el concepto de procedimiento. Un procedimiento se puede asimilar a una función que puede retornar más de un valor mediante el uso de parámetros por referencia³.

Los procedimientos se usan para evitar duplicación de código y conseguir programas más cortos. Son también una herramienta conceptual para dividir un problema en subproblemas logrando de esta forma escribir más fácilmente programas grandes y complejos.

En el pseudocódigo un procedimiento se define de la siguiente manera

```
procedimiento <nombre> ( param1: tipo1, ..., paramn : tipon)  
variables  
    <declaraciones>  
inicio
```

Donde:

- <nombre>: representa el nombre del procedimiento.
- param_i: representa el parámetro i-ésimo del procedimiento.
- tipo_i: representa el tipo del i-ésimo parámetro del procedimiento.
- <declaraciones>: representa el conjunto de variables definidas para el procedimiento (diferentes a los parámetros).
- <instrucciones>: representa el conjunto de instrucciones que realiza el procedimiento.

c. ¿Qué quiere decir sobrecarga de métodos?

La **sobrecarga de Métodos** se apoya de métodos y constructores, los destructores no se pueden sobrecargar ya que sólo debe haber uno por clase.

La sobrecarga de métodos hace que un mismo nombre pueda representar distintos métodos con distinto tipo y número de parámetros, manejados dentro de la misma clase. En el ámbito de la POO, la sobrecarga de métodos se refiere a la posibilidad de tener dos o más métodos con el mismo nombre pero distinta funcionalidad. Es decir, dos o más métodos con el mismo nombre realizan acciones diferentes y el compilador usará una u otra dependiendo de los parámetros usados. Esto también se aplica a los constructores (de hecho, es la aplicación más habitual de la sobrecarga).

Se pueden diferenciar varios métodos sobrecargados a través de sus parámetros, ya sea por la cantidad, el tipo o el orden de los mismos.

Gracias a la sobrecarga de métodos, una clase puede tener distinto comportamiento dependiendo de cual método sobrecargado se use,

a esta característica se le conoce como Polimorfismo por sobrecarga. La palabra Polimorfismo viene de la capacidad que tiene una clase de cambiar su comportamiento. En el caso del polimorfismo por sobrecarga, el cambio del comportamiento de una clase se define sobrecargando los métodos necesarios para lograr el polimorfismo.