

RESPUESTAS

Investigar y presentar en un documento de Word las siguientes preguntas:

1. ¿Qué es?, ¿Cómo utilizar? y diferencias entre Scanner y/o BufferedReader en Java. Muestre Un ejemplo de cada una.
2. Como se llama el modificador que se utiliza para declarar constantes en Java, de un ejemplo.
3. Realice un glosario con los siguientes términos: public, private, protected, package, sentencias Break, Continue y Return, bloque Try Catch Finally.

R:/

1. Scanner: Es una clase del paquete *java.util*, que puede capturar por teclado tipos de datos primitivos tales como Entero (byte, short, int, long), Decimal (float, double) y cadenas(String). Además, nos permite en el apartado de persistencia leer archivos o flujos en su constructor.

Ejemplo:

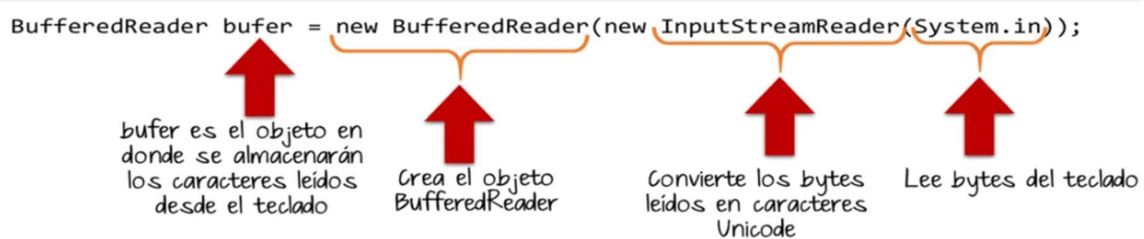
Para captura de tipos primitivos:

```
Scanner keyboard = new Scanner(System.in);  
Int entero = keyboard.nextInt();  
float flotante = keyboard.nextFloat();  
String cadena = keyboard.nextLine();
```

2. ***BufferedReader***: Es una clase del paquete *java.io*, que se apoya en un objeto *InputStreamReader*, que al igual que *Scanner* el constructor recibe un flujo de entrada por teclado (*System.in*), todo esto para capturar de forma más eficiente lo que se digite.

En la siguiente ilustración se observa la instanciación de un objeto tipo *BufferedReader*.

Ilustración 1 Extraída de <https://dcodingames.com/como-usar-la-clase-bufferedreader/>



Ejemplo:

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
String name = reader.readLine();
```

2. Como se llama el modificador que se utiliza para declarar constantes en Java, de un ejemplo.

R: /

Para declarar constantes en Java se usa la palabra reservada *final* que indica que los valores están definidos para un atributo o variable no pueden cambiar.

Ejemplo:

```
final int x=10;
```

Esto nos dice que la variable entera x mantendrá su valor de 10 dentro de su ámbito.

3. Realice un glosario con los siguientes términos: **public**, **private**, **protected**, **package**, sentencias **Break**, **Continue** y **Return**, bloque **Try Catch Finally**.

R=/

public: Modificador de acceso que permite que cualquier método o clase fuera de la clase donde se designa pueda acceder a dicho atributo, método o clase. Siempre y cuando se haga una instancia de dicha clase

private: Modificador de acceso que evita que cualquier método o clase fuera de la clase donde se designa pueda acceder a dicho atributo, método o clase. Solo a través de métodos denominados *setter* y *getter* se debe acceder a dicho atributo, método o clase que lo lleva. Todo esto mientras no sean del mismo paquete.

protected: Modificador de acceso que evita que cualquier método fuera del paquete donde está designado pueda acceder a dicho atributo, método o clase. Se hace por esto indispensable importar la clase con su respectiva dirección de paquete.

package: Palabra reservada para indicar el conjunto de clases que serán accedidas dentro de una carpeta/paquete

break: Funciona como control de bucles. Interrumpe la iteración en la línea donde se coloca.

continue: Funciona como control de bucles. Salta a la siguiente iteración en la línea donde se coloca haciendo que el código siguiente no sea ejecutado.

return: Sirve para retornar el valor estipulado por un método. Es decir si el método no es void devolverá el tipo de dato que se coloque al lado de la palabra reservada return. En caso de ser *void* finalizará el bloque de código siguiente.

try/catch: Sirve para capturar excepciones producidas en tiempo de ejecución. Todo lo que coloque en el bloque *try* se ejecutará salvo se produzca un error, en dado caso se realizará lo que se especifique en el bloque *catch*. La diferencia con un bloque condicional es que junto a la sentencia *catch* se define el tipo de excepción o error que probablemente se produzca, para lo que se define el bloque *catch*.

finally: Se utiliza acompañando un bloque *try/catch*. Sirve para puntualizar aquello que sucederá si se hace el bloque *catch*. Se usa con frecuencia para asegurar el cierre de todos los recursos utilizados.