Práctica 3 - Diseño descendente

Cuando se pide un diseño es hacer el pseudocódigo. En todos los casos resolver haciendo uso del diseño descendente.

1. Escribir un diseño de programa que mientras que el usuario ingrese un número distinto de 0, pida ingresar otro número y lo imprima.
2. Escribir un diseño de programa que mientras que el usuario ingrese un carácter dígito o carácter letra minúscula, imprima dicho carácter, y si es carácter letra minúscula, imprima si es vocal o consonante.
3. Escribir un diseño de programa que mientras que el usuario ingrese un número natural, pida ingresar otro número cualquiera y lo imprima.
4. Escribir un diseño de programa que mientras que el usuario ingrese un número distinto de 0, pida ingresar otros dos números e imprima el resultado de la multiplicación entre los dos últimos números ingresados.
5. Hacer un método que dado un número entero con valor inicial 1, haga una iteración incrementando el número de a uno hasta un valor MAX = 4 (constante). Mientras itera deberá imprimir el número. Luego invocarlo desde el programa principal y cuando termina, imprimir por pantalla “terminó”.
6. Escribir un programa que mientras el usuario cargue desde teclado un carácter letra minúscula, llame a un método que imprime por pantalla la tabla de multiplicar del 9.
7. Escribir un programa que mientras el usuario cargue desde teclado un número entero distinto de 0, imprima por pantalla la suma que se obtiene de invocar un método que calcula la sumatoria de los primeros 200 números naturales (son números enteros entre 1 y 200).
8. Realizar un programa que dado dos números enteros y un carácter (todos ingresados por el usuario) muestre por pantalla el resultado de una operación matemática básica según el valor del carácter ingresado. Si se ingresó el carácter ‘a’ debe realizar la suma, para ‘b’ la resta, para ‘c’ la multiplicación y para ‘d’ la división entre ambos números.
9. Escribir un programa que mientras el usuario ingrese un número entero entre 1 y 10, pida ingresar un carácter, y por cada carácter ingresado imprima:

a. “letra minúscula” si el carácter es una letra del abecedario en minúscula;
b. “letra mayúscula” si el carácter es una letra del abecedario en mayúscula;
c. “dígito” si el carácter corresponde a un número;
d. “otro” para los restantes casos de caracteres.

10. Escribir un programa que mientras el usuario ingrese un número entero entre 1 y 10 realice:
    a. Si el numero ingresado es múltiplo de 3 pida ingresar un caracter y para el caracter ingresado imprima a qué tipo de carácter está asociado:
        i. “letra minúscula” si el carácter es una letra del abecedario en minúscula;
        ii. “letra mayúscula” si el carácter es una letra del abecedario en mayúscula;
        iii. “dígito” si el carácter corresponde a un número;
        iv. “otro” para los restantes casos de caracteres.
    b. Si el número ingresado es múltiplo de 5 imprima la tabla de multiplicar del número ingresado.
11. Escribir un método que retorne el mayor de dos números. Usar ese método para calcular el máximo de una serie de números ingresados por el usuario (20 números en total).
12. Escribir un programa que simule 1000 lanzamientos de un dado y muestre por pantalla cuántas veces salió el valor del dado correspondiente al número entero N ingresado por el usuario.
    Considerar que el valor N ingresado se corresponda a un valor posible para un dado. Usar la sentencia **Math.random()** que devuelve un valor aleatorio real entre 0 y 1.
    Para asignar un posible valor a la variable dado entero:

    dado = (int) (6*Math.random() + 1)

13. Realizar un programa que, dada una opción entera (1,2,3,4), permita realizar operaciones entre tres números reales (r1,r2,r3) ingresados desde teclado. Para la opción:
    1. Calcular la raíz cuadrada de (r1-r3)
    2. Calcular el promedio de r1, r2 y r3
    3. Calcular cociente de la raíz cuadrada de (r3 - r2) dividida por r1
    4. Calcular el cociente del promedio de los tres valores dividido por la raíz cuadrada de r2

    Observación: La raíz cuadrada de un número se calcula con la sentencia: **Math.sqrt(numero)**. Hacer uso del diseño descendente y de la modularización para la solución. Tener en cuenta posibles errores o excepciones como la división por cero.

14. Escribir un programa que mientras el usuario ingresa un carácter distinto del carácter '*', invoque a un método que imprima si es carácter dígito o carácter letra minúscula, y si es letra minúscula imprimir si es vocal o consonante.
15. Escribir un programa que mientras que el usuario ingrese un número entero natural, llame a un método que calcule la sumatoria desde 1 a dicho número (Ej: si n= 5 sumatoria = 1+2+3+4+5=15) y retorne el resultado.
16. Escribir un programa que mientras el usuario ingresa un número de mes (entero) entre 1 y 12 inclusive, muestre por pantalla la cantidad de días del mes ingresado (suponer febrero de 28 días) (Mostrar por pantalla la cantidad de días del mes debería realizarse con un método).
17. Escribir un programa que mientras que el usuario ingrese un carácter letra minúscula, pida ingresar un número entero. Si el número ingresado está entre 1 y 5 inclusive deberá imprimir la tabla de multiplicar de dicho número.
18. Escribir un programa que imprima por pantalla los números perfectos que existen entre 2 y 10000. Un número perfecto es aquel cuya suma de sus divisores (excepto sí mismo) es igual al propio número. Ejemplo: 6 es un número perfecto dado que sus divisores (excepto sí mismo) son 1, 2 y 3 cuya suma da 6. En cambio 12, no es un número perfecto dado que sus divisores (excepto sí mismo) 1, 2, 3, 4 y 6 cuya suma es 16.

Bonus Track 1 - Ejercicio dominio real

1. La cadena de supermercados Carrefive ha pedido la refactorización del componente de software que usa para el control de stock. El código original no estaba modularizado por lo tanto no era legible y no se aprovechaba la reutilización. Dado un artículo (id y nombre), fecha y hora y una cantidad, el componente que maneja el stock genera los siguientes reportes:
    a. Cuando la cantidad de unidades del artículo llega a un límite de seguridad (valor constante) se debe reportar (por consola) los siguientes datos que deben ser enviados al proveedor de dicho artículo: ID_Producto, Nombre, Fecha y Hora (ingresada por el usuario), Cantidad, Prioridad baja.
    b. Cuando la cantidad de unidades está por debajo del límite de seguridad en más de un 30% se debe aumentar la prioridad de baja a media y actualizar la cantidad, la fecha y

la hora en el reporte.

c. Cuando la cantidad de unidades está por debajo del límite de seguridad en más de un 70% la prioridad cambia a alta y se debe actualizar la cantidad, la fecha y la hora en el reporte.
d. Cuando la cantidad es cero se debe imprimir por consola un mensaje “No hay más unidades del producto” + ID_Producto + Nombre. También se debe actualizar la cantidad (0), la fecha y la hora en el reporte.
e. Si la cantidad vendida excede lo que hay en stock (definir un método getStock(id) que le consulte al usuario y retorne la cantidad de stock del producto) se debe informar y no descontar del stock. Informar cuánto stock queda para vender de ese producto.

Escribir un programa con declaración de constantes y variables que haga uso de los métodos definidos y que mientras el usuario no ingrese 0 (en un sistema real lo genera otro componente por ejemplo el que se ejecuta en la caja registradora) pida el ingreso de los datos necesarios (la prioridad se debe obtener automáticamente dependiendo de los valores) para generar el reporte correctamente. Modularizar la solución.

2. Los días lunes se recibe la mercadería que envían los proveedores y se debe actualizar el stock. Suponga que se tiene un método que modifica el stock con la cantidad recibida y que invoca a otro método que elimina la línea del pedido para el producto recibido (se conoce el ID_Producto). Programar un método al cual se le pase el día de la semana y si es lunes pida al usuario el ID_Producto y la cantidad recibida. Con estos datos invoque al método que modifica el stock (no implementar) y al que elimina la línea del pedido (no implementar). Hacer un programa que llame a los diferentes métodos, establezca variables y constantes. Finalmente imprima cuanto stock quedó.
3. El proyecto SETI analiza información proveniente de radiotelescopios e intenta determinar si corresponde a una señal inteligente de vida extraterrestre. Para este proyecto, se pide un programa que permita a un operador ingresar valores enteros positivos (si ingresa 0 termina el turno del operador) que representan diferentes intensidades captadas por el radiotelescopio. En el inicio del turno (antes de ingresar los valores del radiotelescopio), el operador indica el tipo de patrón a buscar (1 o 2) y la cantidad de repeticiones de dicho patrón. Hay dos tipos de patrones buscados:
    - Patrón primo (1): la intensidad captada es un número

primo.
- Patrón divisor (2): la intensidad captada es múltiplo de la unidad de la propia intensidad (Ej: el 357 cumple porque 357 es múltiplo de 7, mientras que 837 no)

El programa debe indicar que se halló el patrón buscado cuando haya habido tantas repeticiones del patrón como lo indicó el operador (luego de hallar un patrón las repeticiones comienzan nuevamente desde 0). Al finalizar el turno, se debe indicar la cantidad de patrones hallados durante el mismo.

4. Revisar los ejercicios 3 y 4 de la práctica 2 y analizar la posibilidad de modularizar el código.

Bonus Track 2 - Ejercicio análisis de código

1. Dados los siguientes códigos, analizar y detectar los errores. Justificar en cada caso.

a.

```
public class Practica_3_Bonus_1 {
      public static void main (String [] args){
            int resultado=0;
            System.out.println("Tabla de multiplicar del 7");
            resultado=imprimirTabla7(resultado);
      }

      public static void imprimirTabla5(){
             for (int i = 1 ; i <= 10; i++) {
                 System.out.println(5*i);
             }
      }
}
```

b.

```
public class Practica_3_Bonus_2 {
  public static void main(String []args){
        char letra='a';
        letra=Utils.leerChar();
        if (letra>='a' && letra<='z'){
            System.out.println("Es una letra minuscula");
        }
  }
}
```

c.

```
public class Practica_3_Bonus_3 {
  public static int numero = 2;
  public static void main(String[] args) {
      int b = 3;
      {
```

```
            System.out.println (a + ", " + numero);
            int b = 2;
            int a = 5;
            System.out.println (a + ", " + b);
            {
                  int c = 7;
                  System.out.println (a + ", " + b + ", " + c);
            }
            System.out.println (a + ", " + b + ", " + c);
      }
   }
}
```

2. Dados los siguientes enunciados y códigos, analizar y detectar los errores. Justificar en cada caso.
   a. Escribir un programa que llame un método que calcule el promedio de la suma de valores enteros entre 1 y 1000. Finalmente, el promedio debe mostrarse por pantalla

```
public class Practica_3_Bonus_4 {
  public static final int MAX = 1, MIN = 1000;
  public static void main(String[] args) {
    System.out.println("El promedio de la suma entre " MIN " y
" MAX " es "+ calcular_promedio());
  }

  public static char calcular_promedio(int MAX, int MIN) {
      int suma = 0;
      int numero = MIN;
      for (; numero <= MAX; numero++) {
            suma += numero;
      }
      return suma/(MAX-MIN+1);
  }
}
```

   b. Dados dos números positivos y un carácter opción cuyo valor es una letra minúscula ‘s’ o ‘r’ (únicamente puede tomar estos dos valores), obtener la suma de ambos números si la opción es ‘s’ o si es ‘r’ se debe restar el primero al segundo.

```
public class Practica_3_Bonus_5 {
      public static void main(String[] args) {
            int numero1=235; // valor de ejemplo
            int numero2=-5; // valor de ejemplo
            char opcion='s';
            resolver(opcion,numero1,numero2);
      }

      public static void resolver(int num1, int num2, char op){
```

```
            if (op==’s’){
                  System.out.println("La suma es: "+num1+num2);
            }
            else if (op==’r’){
                  System.out.println("La resta es:
      "+num1-num2);
            }
            else
                  System.out.println("Opción no válida”);
      }
}
```