

Práctica 5. Fundamentos de Programación

Trim.16-I

Profra. Graciela Román Alonso

1. Abre Eclipse.

-Selecciona un espacio de trabajo (workspace), el Escritorio.

-Construye un nuevo proyecto: Selecciona File -> New -> Java project, da el nombre de proyecto: Pract5_FP -> Finish

-Crearemos una clase. Posiciona el mouse sobre el nombre del proyecto y con el botón derecho selecciona -> New -> Class. Luego en la ventana emergente escribe el nombre de la clase en el campo **name**: Prog1_java, marca el cuadrito que indica: **public static void main**. Luego haz click en Finish. Observa el código generado automáticamente:

```
public class Prog1_java {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
    }  
}
```

Si no ves el código, abre el directorio del proyecto, luego abre el directorio src el cual contiene a la clase Prog1_java. Haz doble click sobre el nombre de la Clase y observa el código generado automáticamente.

-El texto que se encuentra delimitado por **/**** y ***/** se considera información para documentar un programa Java, el compilador no lo toma en cuenta. Por ahora vamos a empezar a documentar nuestros programas indicando la versión del programa (usando **@version**) y su(s) autor(es) (usando **@author**), por ejemplo de la siguiente manera:

```
public class Prog1_java {  
    /**  
     * @version  
     * aquí va la fecha  
     * @author  
     * aquí va tu nombre y apellidos  
     */  
  
    /**  
     * Aquí va un comentario sobre lo que hace el programa  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
    }  
}
```

Modifica el método main para que se calcule el promedio de un conjunto de n números dados por el usuario.

Paso 1.1.- Primero pide al usuario el valor de n, y valida que n>0, si el número es inválido pedir al usuario que vuelva a dar el número. Usa **do-while** en este paso, sintaxis de **do-while**:

```
do {  
    instrucciones;  
}while(condición);
```

Paso 1.2.- Haz que el usuario ingrese los n números y calcula el promedio. Usa un ciclo **para** en este paso, sintaxis del ciclo **para**:

```
for ( i= val_inicial; i<= val_final ; i++)  
    acción;
```

- Al final muestra al usuario el promedio de los números dados
- Modifica los comentarios de Java con tus datos y la descripción del programa

2. Crearemos otra clase. Posiciona el mouse sobre el nombre del proyecto Pract5_FP y con el botón derecho selecciona -> New -> Class. Luego en la ventana emergente escribe el nombre de la clase en el campo name: Prog2_java y selecciona la opción : public static void main(String [] args) -> Finish.

Abre el directorio del proyecto, luego abre el directorio **src** el cual contiene a la clase Prog2_java.

Haz doble click sobre el nombre de la Clase Prog2_java para empezar a editar el programa.

Los ciclos **para** o **for** también pueden ir anidados, uno dentro de otro. Por ejemplo, introduce el siguiente código en el método main y corre el programa. Qué desplegado se genera?

```
int i,j;  
    for(i=0; i <=2; i++)  
    {  
        System.out.println(" i:" + i);  
        for(j=0; j<=4; j++)  
            System.out.println(" >>> j:" + j);  
    }
```

Observa que el primer ciclo for depende de la variable **i** y el segundo for depende de la variable **j**. El ciclo for más interno es el que cambia más rápido. Cuando **i** vale 0, el valor de **j** varía desde 0 a 4, luego cuando **i** vale 1, el valor de **j** otra vez varía de 0 a 4, y en general, para cada valor de **i** el valor de **j** va a variar de 0 a 4, hasta que **i** es mayor que 2.

Dentro del segundo ciclo for, después de desplegar el valor de **j**, anida un tercer ciclo for que dependa de la variable **k**, $0 \leq k \leq 3$. Haz que el valor de **k** sea desplegado un poco mas a la derecha que el de **j**. Observa la ejecución.

Documenta tu programa.

3.- Crea una nueva clase Prog3_java y escribe un programa para que se genere una figura geométrica triangular, siguiendo los siguientes pasos.

Paso 3.1.

Con un ciclo **para**, haz que se despliegue una línea formada por **n** asteriscos (*), considerando que **n** es un número dado por el usuario, **n**>0 (validarlo). Utiliza System.out.print en vez de System.out.println para que el desplegado no salte de línea.

Por ejemplo,
si $n = 5$, se desplegará la siguiente línea:

si $n = 12$, se desplegará la siguiente línea:

Paso 3.2.

Pon el ciclo **para** anterior dentro de otro ciclo **para**, de tal modo que se puedan desplegar varias líneas de asteriscos, como lo vemos en la siguiente figura geométrica, cuando **n** tiene un valor de 6:

```
*
**
***
****
*****
*****
```

Paso 3.3.- Modifica el programa para que el desplegado se vea así:

```
  *
  **
 ***
****
*****
*****
```

Para lograr este segundo desplegado es necesario que antes de desplegar los asteriscos se despliegan varios espacios en blanco. Para el primer renglón se despliegan n espacios en blanco seguidos de 1 asterisco, en el segundo renglón se despliegan $n-1$ espacios en blanco seguidos de 2 asteriscos, en el tercero hay $n-2$ espacios en blanco seguidos de 3 asteriscos, etc. En el último renglón solo hay 1 espacio en blanco desplegado antes de los n asteriscos.

Documenta tu programa.

4.- Crea una nueva clase Prog4_java. Escribe un programa para decir si un número $n > 0$ es perfecto o no. Un número perfecto es aquel donde sus factores, incluyendo el 1 pero excluyendo el número mismo, suman igual que el número. Por ejemplo, el 6 es un número perfecto ya que sus factores suman al mismo número, es decir, $1+2+3=6$.

Paso 4.1.- Primero calcula los factores de un número $n > 0$, dado por el usuario. Un factor o divisor de un número entero n , es un número también entero menor que n que lo divide exactamente.

Por ejemplo, si $n == 6$, los factores de n son: 1, 2, 3 y 6
si $n == 12$, los factores de n son: 1, 2, 3, 4, 6 y 12
si $n == 25$, los factores de n son: 1, 5 y 25

Paso 4.2.- Suma los factores del número n , excluyendo el 1 y n , y decide si n es perfecto o no lo es.

Documenta el programa.