

Ejercicios repaso java

Antes de empezar

Cada uno de los ejercicios se van a realizar en una función diferente para facilitar la realización de los ejercicios, de tal modo que solo se tendrá que llamar a esa función desde la función "onCreate".

Una **función** es un conjunto de instrucciones que ejecutan una rutina y devuelven un resultado. El resultado puede depender de parámetros de entrada.

La estructura de una función es la siguiente:

permisos_acceso: Es la propiedad desde la cual se puede acceder a la función. Para estos ejercicios se puede omitir o poner en 'public'.

tipo_resultado: Es el tipo de datos que se va a devolver (En estos ejercicios usaremos 'void' que es no devolver nada). En caso de querer devolver algún dato se tiene que realizar con la sentencia 'return'.

nombre_función: nombre que se le da a la función (en los ejercicios se le darán el nombre de los ejercicios ej: ej1).

```
permisos_acceso tipo_resultado nombre_función (tipo_parámetro nombre_parámetro, ... ) {  
    sentencia1;  
    sentencia2;  
    ...  
    sentenciaN;  
}
```

Para llamar a la función en el mismo .java simplemente tendremos que escribir el nombre de la función y los parámetros solicitados:

```
nombre_funcion(nombre_parametro, ...);
```

A un string se le pueden encadenar tanto números como otras cadenas de la siguiente forma:

```
System.out.println("hola" + "mundo");
```

```
System.out.println("a: " + valor_variable);
```

Para convertir los números a cadenas de texto se utiliza la función `String.valueOf(numero_a_convertir);`

Una vez visto los conceptos anteriores ya podemos pasar a enunciar los ejercicios.

Ejercicios

Para la realización de los ejercicios se pedirá imprimir por pantalla los resultados, para ello se puede utilizar el `System.out.println()` o `Log.d()`. En el caso de usar este último se recomienda poner como tag el nombre del ejercicio para encontrar fácilmente la información en la pantalla.

El nombre de las funciones tiene que ser el nombre del ejercicio escrito en minúsculas (Ej: `ej1()`)

Ej1

Hacer una función para determinar el valor de cada una de las siguientes expresiones. Imprimir los resultados obtenidos por pantalla.

a) $15 * 14 - 3 * 7$

b) $-4 * 5 * 2$

c) $(24 + 2 * 6) / 4$

d) $3 + 4 * (8 * (4 - (9 + 3) / 6))$

e) $4 - 40 / 5$

f) $(-5) \% (-2)$

Ej2

Escribir una función que permita intercambiar el contenido de dos variables. Dichas variables se inicializarán a un valor fijo (4 y 7 por ejemplo). Imprimir el valor de dichas variables antes y después del intercambio. Ej:

a: 4

b: 7

Intercambio

a: 7

b: 4

No vale cambiar el valor de las variables asignando los nuevos números, se tiene que realizar volcando el valor de las mismas.

Ej3

Escribir una función que lea tres enteros pasados como parámetro de la función y diga si están ordenados de menor a mayor, de mayor a menor o desordenados.

Ej4

Escribir una función que muestre el siguiente dibujo, pudiendo pintar solamente un '*' cada vez y haciendo uso de bucles. **Nota:** para este ejercicio es obligatorio el uso de la sentencia System.out.print y System.out.println puesto que son Log.d no saldría de la manera que se espera.

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

Resultado: Es muy probable que en la consola de Android Studio no se muestre exactamente igual que en lo propuesto al inicio del ejercicio (incluso que se repita el "I/System.out:"):

Ejercicio 4:

```
I/System.out: *
  **
```

```
I/System.out: ***
  ****
```

Esto sería una resolución correcta del ejercicio.

Ej5

Escribir una función que muestre la siguiente figura, pudiendo pintar solamente un '*' cada vez y haciendo uso de bucles. **Nota:** para este ejercicio es obligatorio el uso de la sentencia System.out.print y System.out.println puesto que son Log.d no saldría de la manera que se espera.

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

Resultado: Es muy probable que en la consola de Android Studio no se muestre exactamente igual que en lo propuesto al inicio del ejercicio (incluso que se repita el "I/System.out:"):

Ejercicio 5:

```
I/System.out: ****
  ***
  **
  *
```

Esto sería una resolución correcta del ejercicio.

Ej6

Escribir una función que visualice la siguiente salida:

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

Nota: para este ejercicio es obligatorio el uso de la sentencia `System.out.print` y `System.out.println` puesto que son `Log.d` no saldría de la manera que se espera.

Resultado: Es muy probable que en la consola de Android Studio no se muestre exactamente igual que en lo propuesto al inicio del ejercicio (incluso que se repita el “`I/System.out:`”):

Ejercicio 6:

```
1
I/System.out: 12
123
1234
I/System.out: 123
12
1
```

Esto sería una resolución correcta del ejercicio.

Ej7

. Escribir un programa `TablasMultiplicar` que muestre las tablas de multiplicar de los números del 1 al 10, con el siguiente formato:

```
1 x 0 = 0
1 x 1 = 1
...
1 x 10 = 10
-----
2 x 0 = 0
2 x 1 = 2
...
2 x 10 = 20
-----
...
```

Ej8

Escribir una función que calcule la suma de los pares existentes entre 0 y 100 por un lado y la de los impares por el otro

Ej9

Escribir una función que muestre las permutaciones con repetición que pueden hacerse con los números 1,2,3 y 4. También que muestre el número de estas.

PERMUTACIONES CON REPETICIÓN

1 1 1 1	1 3 3 2	2 2 1 3
1 1 1 2	1 3 3 3	2 2 1 4
1 1 1 3	1 3 3 4	2 2 2 1
1 1 1 4	1 3 4 1	2 2 2 2
1 1 2 1	1 3 4 2	2 2 2 3
1 1 2 2	1 3 4 3	2 2 2 4
1 1 2 3	1 3 4 4	2 2 3 1
1 1 2 4	1 4 1 1	2 2 3 2
1 1 3 1	1 4 1 2	2 2 3 3
1 1 3 2	1 4 1 3	2 2 3 4
1 1 3 3	1 4 1 4	2 2 4 1
1 1 3 4	1 4 2 1	2 2 4 2
1 1 4 1	1 4 2 2	2 2 4 3
1 1 4 2	1 4 2 3	2 2 4 4
1 1 4 3	1 4 2 4	2 3 1 1
1 1 4 4	1 4 3 1	2 3 1 2
1 2 1 1	1 4 3 2	2 3 1 3
1 2 1 2	1 4 3 3	2 3 1 4
1 2 1 3	1 4 3 4	2 3 2 1
1 2 1 4	1 4 4 1	2 3 2 2
1 2 2 1	1 4 4 2	2 3 2 3
1 2 2 2	1 4 4 3	2 3 2 4
1 2 2 3	1 4 4 4	2 3 3 1
1 2 2 4	2 1 1 1	2 3 3 2
1 2 3 1	2 1 1 2	2 3 3 3
1 2 3 2	2 1 1 3	2 3 3 4
1 2 3 3	2 1 1 4	2 3 4 1
1 2 3 4	2 1 2 1	2 3 4 2
1 2 4 1	2 1 2 2	2 3 4 3
1 2 4 2	2 1 2 3	2 3 4 4
1 2 4 3	2 1 2 4	2 4 1 1
1 2 4 4	2 1 3 1	2 4 1 2
1 3 1 1	2 1 3 2	2 4 1 3
1 3 1 2	2 1 3 3	2 4 1 4
1 3 1 3	2 1 3 4	2 4 2 1
1 3 1 4	2 1 4 1	2 4 2 2
1 3 2 1	2 1 4 2	2 4 2 3
1 3 2 2	2 1 4 3	2 4 2 4
1 3 2 3	2 1 4 4	2 4 3 1
1 3 2 4	2 2 1 1	2 4 3 2
1 3 3 1	2 2 1 2	2 4 3 3

2434	3331	4222
2441	3332	4223
2442	3333	4224
2443	3334	4231
2444	3341	4232
3111	3342	4233
3112	3343	4234
3113	3344	4241
3114	3411	4242
3121	3412	4243
3122	3413	4244
3123	3414	4311
3124	3421	4312
3131	3422	4313
3132	3423	4314
3133	3424	4321
3134	3431	4322
3141	3432	4323
3142	3433	4324
3143	3434	4331
3144	3441	4332
3211	3442	4333
3212	3443	4334
3213	3444	4341
3214	4111	4342
3221	4112	4343
3222	4113	4344
3223	4114	4411
3224	4121	4412
3231	4122	4413
3232	4123	4414
3233	4124	4421
3234	4131	4422
3241	4132	4423
3242	4133	4424
3243	4134	4431
3244	4141	4432
3311	4142	4433
3312	4143	4434
3313	4144	4441
3314	4211	4442
3321	4212	4443
3322	4213	4444
3323	4214	
3324	4221	

Hay 256 permutaciones con repetición