Ejercicios Basicos Kotlin

Descargar estos ejercicios

Ejercicio 1

Este ejercicio va a servir para practicar los bucles `for`, el uso de `StringBuilder` y la generación de `números aleatorios` en Kotlin. Para ello vamos a crear un generador de contraseñas siguiendo los siguientes pasos:

1. Crearemos una función **generaNumeros** que devolverá una cadena con los números del 0 al 9 invertidos [987..]. La cadena tendrá una longitud X, siendo X un número aleatorio menor que 10. Investiga el `reverse` de StringBuilder.
2. Otra función será **generaMayusculas** que nos devolverá una cadena con un tamaño indeterminado de letras mayúsculas desordenadas. Para conseguirlo recorreremos los caracteres de la 'A' a la 'Z' (usando el rango A..Z), y decidiremos si añadir la letra o no en la salida a partir del resultado devuelto por `nextBoolean` de Random. Con el `insert` de StringBuilder podremos insertar la letra en un lugar aleatorio, y así conseguir el desorden de la salida (cuidado, no podremos insertar fuera del tamaña del StringBuilder).
3. La función **generaMinusculas** la conseguiremos a partir de la anterior.
4. Además tendremos la función **generaContraseña** a la que le llegará el tamaño de la contraseña a generar y que se encargará a partir de las tres anteriores de generar el string con los caracteres mezclados. Podemos llamar a las anteriores y con todas las cadenas devueltas, crear un nuevo StringBuilder generado con los caracteres seleccionados de ellas de forma aleatoria, no importa que algunos se repitan.

Tres posibles salidas para una contraseña de tamaño 10, serían:

```
xcJi4RY0Ygi
XBfFff0pcyZ
UUnfhHttaYn
6BZ4n0OnoH8
```

Ejercicio 2

Realiza un programa que genere los lados de n triángulos e informe, de cada uno de ellos, qué tipo de triángulo es: equilátero (tres lados iguales), isósceles (dos lados iguales), o escaleno (ningún lado igual). Además al finalizar la muestra deberá indicar la cantidad de triángulos de cada tipo. Una posible solución para una entrada de 10 podría ser:

```
el triangulo número 1 es ESCALENO
el triangulo número 2 es ESCALENO
el triangulo número 3 es ESCALENO
el triangulo número 4 es ISOSCELES
el triangulo número 5 es EQUILATERO
el triangulo número 6 es ISOSCELES
el triangulo número 7 es ESCALENO
el triangulo número 8 es ESCALENO
el triangulo número 9 es ISOSCELES
el triangulo número 10 es EQUILATERO
En total se han generado:
2 triangulos equilateros
3 triangulos isosceles
5 triangulos escalenos
```

Para ello crearemos una función **generaLados** que devolverá una `Triple` (investiga este tipo de datos) con los valores de los lados de un triangulo generados aleatoriamente (el rango de medición podría ser entre 1 y 10). Además tendremos una función **resuelveTriangulo** al que le llega el dato de la medición de los lados y devolverá la cadena con el tipo de triangulo. El programa principal se encargará de realizar las llamadas y el conteo necesario.

> **Aviso**
>
> Los condicionales se deberán de hacer con la sentencia **when** y la repetición con el bucle **for**

Ejercicio 3

Se va a crear una aplicación que permita determinar si una persona puede entrar al **Euskal Encounter**, encuentro al que solo pueden entrar los mayores de edad muy frikis. Para ello se

deberá crear una función **leeEdad** que pedirá una edad mientras la entrada no sea vacía, devolverá la edad como tipo Short siempre que sea posible (numérica), para comprobar que la entrada es solo numérica se tendrá que crear una extensión del tipo String llamada **sonSoloDigitos** que devolverá true en caso de que todos los caracteres de la cadena sean dígitos, false en cualquier otro caso. Si al leer la edad, se ha introducido una entrada no valida se lanzará una excepción con el mensaje *Has introducido una edad no válida*, para ello crearemos una función **fail** a la que le llegará una cadena con el mensaje y lanzará una excepción del tipo `IllegalArgumentException`. Además tendremos la función **permiteEntrada** a la que le llega un Short y devolverá una cadena con el mensaje de si se puede entrar o no, dependiendo de si la edad es mayor de 17 años. En la función principal deberás capturar las excepción y llamar a los métodos necesarios.