

Exámen técnico java developer

Nota : El tiempo de resolución de la prueba es de **2 días** como máximo después de su recepción. Una vez terminada la solución, subir el código fuente a un repositorio público de git (github por ejemplo) y brindar la URL como respuesta.

PARTE 01: Resolver los siguientes casos usando íntegramente Java

Consideraciones preliminares:

Problema 01

Usando Java, crear una clase llamada **ChangeString** que tenga un método llamado **build** el cual tome un parámetro string que debe ser modificado por el siguiente algoritmo. Reemplazar cada letra de la cadena con la letra siguiente en el alfabeto. Por ejemplo, reemplazar **a** por **b** ó **c** por **d**. Finalmente devolver la cadena.

Indicaciones

- Crear el solución en un solo archivo llamado **ChangeString.java**
- El método build devuelve la salida del algoritmo
- Considerar el siguiente abecedario: a, b, c, d, e, f, g, h, i, j, k, l, m, n, ñ, o, p, q, r, s, t, u, v, w, x, y, z.

Ejemplos

- **entrada :** "123 abcd*3" **salida :** "123 **bcde***3"
- **entrada :** "***Casa 52" **salida :** "*****Dbtb** 52"

Problema 02

Usando Java, crear una clase llamada **CompleteRange** que tenga un método llamado **build** el cual tome un parámetro de colección de números enteros positivos (1,2,3, ...n) en cualquier orden. El algoritmo debe completar si faltan números en la colección en el rango dado. Finalmente devolver la colección completa y ordenada de manera ascendente.

Indicaciones

- Crear la solución en un solo archivo llamado **CompleteRange.java**
- El método **build** devuelve la salida del algoritmo

Ejemplos

- **entrada** : [2, 1, 4, 5] **salida** : [1, 2, **3**, 4, 5]
- **entrada** : [4, 2, 9] **salida** : [**1**, 2, **3**, 4, **5, 6, 7, 8**, 9]
- **entrada** : [58, 60, 55] **salida** : [**1, 2**, ...**54**, 55, **56, 57**, 58, **59**, 60]

Problema 03

Usando Java, crear una clase llamada **MoneyParts** que tenga un método llamado **build** que reciba como parámetro una cadena con un monto en soles y devuelva todas las combinaciones posibles en un arreglo.

Indicaciones

- Crear la solución en un solo archivo llamado **MoneyParts.java**
- El método **build** devuelve la salida del algoritmo
- Considerar las siguientes denominaciones (0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200)

Ejemplos

- **entrada :** "0.1" **salida :** [[0.05, 0.05], [0.1]]
- **entrada :** "0.5" **salida :** [[0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05], [0.1, 0.1, 0.1, 0.1, 0.1]]
- **entrada :** "10.50" **salida :** [[0.05, 0.05, 0.05], [10.0, 0.5]]

Los puntos suspensivos representan todas las combinaciones posibles para representar la cantidad monetaria ingresada.

PARTE 02: Desarrollar los siguientes servicios

Consideraciones preliminares:

- Los servicios deben ser realizada usando íntegramente Java 8 con Maven.
- Se debe usar el framework Apache CXF (<https://cxf.apache.org/>) , JAX-RS, JAX-WS, para el desarrollo de los servicios.
- Realizar las pruebas unitarias de los servicios usando Junit4.
- Usar JPA para la capa de datos

Problema práctico

Un aplicativo web que tiene como finalidad manejar bancos, sucursales y sus diversas órdenes de pagos. Para ello considerar los siguientes atributos de cada entidad:

Banco	Sucursales	Órdenes de pago
<ul style="list-style-type: none">• nombre• dirección• fecha de registro	<ul style="list-style-type: none">• nombre• dirección• fecha de registro	<ul style="list-style-type: none">• monto• moneda• estado• fecha de pago

La aplicación web considera las siguientes reglas de negocio. Un banco puede tener muchas sucursales y las órdenes de pago pueden ser pagadas en diferentes sucursales de cada banco en soles ó dólares. Asimismo, el estado de la orden de pago puede ser pagada, declinada, fallida y anulada.

En primer lugar, se desea crear un servicio web en formato JSON para el mantenimiento de todas las entidades de la aplicación, usar JAX-RS.

En segundo lugar, se desea liberar un servicio web en formato JSON que pueda listar todas las órdenes de pago de una sucursal contemplando el filtrado por tipo de moneda, usar JAX-RS.

Finalmente, se debe liberar un servicio web en formato XML que permita buscar todas las sucursales de acuerdo con un banco, usar JAX-WS.