Resumen: Arrays y Listas (CFGS DAM)

Resumen completo: Arrays y Listas

PARTE 1 ARRAYS (ARREGLOS)

DEFINICIN

Un array es una estructura de datos que guarda varios valores del mismo tipo, con un tamao fijo.

CARACTERSTICAS PARA MEMORIZAR

- El tamao es fijo (no se puede agrandar directamente).
- El ndice empieza en 0.
- Todos los elementos son del mismo tipo.
- Se puede acceder directamente con el ndice: array[i].
- Cuando se crean, ya tienen valores por defecto.

VALORES POR DEFECTO

int -> 0

double -> 0.0

boolean -> false

String -> null

EJEMPLO BSICO

```
int[] numeros = new int[3];
numeros[0] = 10;
numeros[1] = 20;
numeros[2] = 30;
System.out.println(numeros[1]); // Imprime: 20
```

ERRORES FRECUENTES

- Acceder a una posicin que no existe (ArrayIndexOutOfBoundsException).
- Olvidar que el ndice empieza en 0.

Resumen: Arrays y Listas (CFGS DAM)

AGRANDAR UN ARRAY DINMICAMENTE

1. Crear un array nuevo con una posicin ms.
2. Copiar todos los elementos del original.
3. Aadir el nuevo dato.
4. Reemplazar el original.

```
int[] original = {1, 2, 3};
int[] nuevo = new int[original.length + 1];
for (int i = 0; i < original.length; i++) {
    nuevo[i] = original[i];
}
nuevo[nuevo.length - 1] = 4;
original = nuevo;
```

INVERTIR UN ARRAY

MTODO 1: Crear un array nuevo (ms fcil)

```
int[] original = {10, 20, 30, 40};
int[] invertido = new int[original.length];
for (int i = 0; i < original.length; i++) {
    invertido[i] = original[original.length - 1 - i];
}
```

MTODO 2: Invertir el array en el mismo lugar

```
int[] datos = {10, 20, 30, 40};
for (int i = 0; i < datos.length / 2; i++) {
    int temp = datos[i];
    datos[i] = datos[datos.length - 1 - i];
    datos[datos.length - 1 - i] = temp;
}
```

Resumen: Arrays y Listas (CFGS DAM)

PARTE 2 LISTAS (ArrayList)

DEFINICIN

Una lista es como un array, pero dinmico. Se puede agrandar o reducir automticamente y tiene mtodos ms cmodos.

CARACTERSTICAS PARA MEMORIZAR

- Tamao dinmico (se ajusta solo).
- Solo puede contener objetos (String, Integer, etc.).
- Usa mtodos como .add(), .get(), .set(), .remove().
- Necesita importar: import java.util.ArrayList;

EJEMPLO BSICO

```
ArrayList<String> nombres = new ArrayList<>();
nombres.add("Ana");
nombres.add("Luis");
System.out.println(nombres.get(1)); // Imprime: Luis
```

MTODOS TILES

```
.add(valor)      -> Aade al final
.get(i)          -> Obtiene el elemento en posicin i
.set(i, valor)   -> Cambia el valor en i
.remove(i)       -> Elimina el elemento en i
.size()          -> Devuelve el nmero de elementos
```

ERRORES FRECUENTES

- No se puede usar int directamente. Usar Integer.

```
ArrayList<Integer> numeros = new ArrayList<>();
numeros.add(5);
```

Resumen: Arrays y Listas (CFGS DAM)

DIFERENCIAS CLARAS ENTRE ARRAY Y LISTA

ARRAY (Caja fija) vs LISTA (Bolsa flexible)

- Tamao: Fijo vs Dinmico
- Tipo: Primitivos u objetos vs Solo objetos
- Aadir: Requiere copiar vs Solo .add()
- Acceso: array[i] vs lista.get(i)

TRUCO FINAL PARA RECORDAR

- Si ya sabes cuntos datos habr usa un array.
- Si los datos van llegando con el tiempo usa una lista.