

Declaración y Visualización

1. Crear una **Aplicación Java** llamada **ArrayLogicoCompacto** que:

- Declare un array **de tipo lógico** de 5 posiciones.
- Inicialice el array con valores arbitrarios en la misma declaración.
- Muestre por pantalla el contenido del array, pasando posición por posición, en el formato:

```
Estado 1 ...: true
Estado 2 ...: false
Estado 3 ...: false
...
```

2. Crear una **Aplicación Java** llamada **ArrayCaracterCompacto** que:

- Declare un array **de tipo carácter** de 5 posiciones.
- Inicialice el array con valores arbitrarios en la misma declaración.
- Recorra el array utilizando una estructura **while** para mostrar su contenido por pantalla en el formato:

```
Símbolo 1 ...: $
Símbolo 2 ...: /
Símbolo 3 ...: j
...
```

3. Crear una **Aplicación Java** llamada **ArrayStringCompacto** que:

- Declare un array **de tipo String** de 5 posiciones.
- Inicialice el array con nombres arbitrarios en la misma declaración.
- Recorra el array utilizando una estructura **do..while** para mostrar su contenido por pantalla en el formato:

```
Nombre 1 ...: Eustaquio
Nombre 2 ...: MariPuri
Nombre 3 ...: Pakito
...
```

4. Crear una **Aplicación Java** llamada **ArrayRealCompacto** que:

- Declare un array **de tipo real** de 5 posiciones.
- Inicialice el array con valores arbitrarios en la misma declaración.
- Recorra el array utilizando una estructura **for** para mostrar su contenido por pantalla en el formato:

```
Precio 1 ...: 1,23€
Precio 2 ...: 3,21€
Precio 3 ...: 9,75€
...
```

5. Crear una **Aplicación Java** llamada **ArrayEnteroCompacto** que:

- Declare un array **de tipo entero** de 5 posiciones.
- Inicialice el array con valores arbitrarios en la misma declaración.
- Recorra el array utilizando una estructura **for-each** para mostrar su contenido por pantalla en el formato:

```
Aforo actual ...: 123 personas  
Aforo actual ...: 321 personas  
Aforo actual ...: 975 personas  
...
```

Acceso Teclado y Visualización

6. Crear una **Aplicación Java** llamada **ArrayLogicoUsuario** que:

- Declare un array **de tipo lógico** de 5 posiciones.
- Pida al usuario que **introduzca por teclado** un estado lógico para cada posición, pasando una a una por todas y cada una de ellas en el formato:
Estado 1 ...: _
- Muestre por pantalla el contenido del array, pasando posición por posición, en el formato:
Estado 1 ...: true
Estado 2 ...: false
Estado 3 ...: false
...

7. Crear una **Aplicación Java** llamada **ArrayCaracterUsuario** que:

- Declare un array **de tipo carácter** de 5 posiciones.
- Recorra el array utilizando una estructura **while** para pedir al usuario que **introduzca por teclado** un carácter para cada posición en el formato:
Símbolo 1 ...: _
- Recorra el array utilizando una estructura **while** para mostrar su contenido por pantalla en el formato:
Símbolo 1 ...: \$
Símbolo 2 ...: /
Símbolo 3 ...: j
...

8. Crear una **Aplicación Java** llamada **ArrayStringUsuario** que:

- Declare un array **de tipo String** de 5 posiciones.
- Recorra el array utilizando una estructura **do..while** para pedir al usuario que **introduzca por teclado** un nombre arbitrario para cada posición en el formato:
Nombre 1 ...: _
- Recorra el array utilizando una estructura **do..while** para mostrar su contenido por pantalla en el formato:
Nombre 1 ...: Eustaquio
Nombre 2 ...: MariPuri
Nombre 3 ...: Pakito
...

9. Crear una **Aplicación Java** llamada **ArrayRealUsuario** que:

- Declare un array **de tipo real** de 5 posiciones.
- Recorra el array utilizando una estructura **for** para pedir al usuario que **introduzca por teclado** un valor arbitrario para cada posición en el formato:
Precio 1 ...: _
- Recorra el array utilizando una estructura **for** para mostrar su contenido por pantalla en el formato:
Precio 1 ...: 1,23€
Precio 2 ...: 3,21€
Precio 3 ...: 9,75€
...

10. Crear una **Aplicación Java** llamada **ArrayEnteroUsuario** que:

- Declare un array **de tipo entero** de 5 posiciones.
- Recorra el array utilizando una estructura **for-each** para pedir al usuario que **introduzca por teclado** un valor arbitrario para cada posición en el formato:
Aforo actual ...: _
- Recorra el array utilizando una estructura **for-each** para mostrar su contenido por pantalla en el formato:
Aforo actual ...: 123 personas
Aforo actual ...: 321 personas
Aforo actual ...: 975 personas
...

Acceso Aleatorio y Visualización

11. Crear una **Aplicación Java** llamada **ArrayLogicoRandom** que:

- Declare un array **de tipo lógico** de 5 posiciones.
- Rellene el array con valores lógicos aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Estado 1 ...: true
Estado 2 ...: false
Estado 3 ...: false
...

12. Crear una **Aplicación Java** llamada **ArrayCaracterRandom** que:

- Declare un array **de tipo carácter** de 5 posiciones.
- Rellene el array con caracteres aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Símbolo 1 ...: \$
Símbolo 2 ...: /
Símbolo 3 ...: j
...

13. Crear una **Aplicación Java** llamada **ArrayStringRandom** que:

- Declare un array **de tipo String** de 5 posiciones.
- Rellene el array con nombres aleatorios(!).
- Muestre por pantalla el contenido del array en el formato:
Nombre 1 ...: Eustaquio
Nombre 2 ...: MariPuri
Nombre 3 ...: Pakito
...

14. Crear una **Aplicación Java** llamada **ArrayRealRandom** que:

- Declare un array **de tipo real** de 5 posiciones.
- Rellene el array con valores reales aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Precio 1 ...: 1,23€
Precio 2 ...: 3,21€
Precio 3 ...: 9,75€
...

15. Crear una **Aplicación Java** llamada **ArrayEnteroRandom** que:

- Declare un array **de tipo entero** de 5 posiciones.
- Rellene el array con valores enteros aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Aforo actual ...: 123 personas
Aforo actual ...: 321 personas
Aforo actual ...: 975 personas
...

Busqueda y Visualización

16. Crear una **Aplicación Java** llamada **ArrayCaracterBusqueda** que:

- Declare un array **de tipo carácter** de 5 posiciones.
- Rellene el array con caracteres aleatorios.
- Pida al usuario la entrada de un carácter en el formato:
 Buscar: _
- Muestre por pantalla el contenido del array en el formato:
 Símbolo 1 ...: \$
 Símbolo 2 ...: /
 Símbolo 3 ...: j
 ...
- Si el carácter introducido SI se ha encontrado en el array, expresarlo en el formato:
 Símbolo: / - SI encontrado - Posición 1
- Si el carácter introducido NO se ha encontrado en el array, expresarlo en el formato:
 Símbolo: W - NO encontrado

17. Crear una **Aplicación Java** llamada **ArrayStringBusqueda** que:

- Declare un array **de tipo String** de 5 posiciones con nombres de personas arbitrarios.
- Pida al usuario la entrada de un nombre en el formato:
 Buscar: _
- Muestre por pantalla el contenido del array en el formato:
 Nombre 1 ...: Eustaquio
 Nombre 2 ...: MariPuri
 Nombre 3 ...: Pakito
 ...
- Si el nombre introducido SI se ha encontrado en el array expresarlo en el formato:
 Nombre: MariPuri - SI encontrado - Posición 1
- Si el carácter introducido NO se ha encontrado en el array expresarlo en el formato:
 Nombre: Frasquito - NO encontrado

18. Crear una **Aplicación Java** llamada **ArrayRealBusqueda** que:

- Declare un array **de tipo real** de 5 posiciones.
- Rellene el array con valores aleatorios.
- Pida al usuario la entrada de un valor en el formato:
 Buscar: _
- Muestre por pantalla el contenido del array en el formato:
 Precio 1 ...: 1,23€
 Precio 2 ...: 3,21€
 Precio 3 ...: 9,75€
 ...
- Si el valor introducido SI se ha encontrado en el array expresarlo en el formato:
 Precio: 3,21 - SI encontrado - Posición 1
- Si el valor introducido NO se ha encontrado en el array expresarlo en el formato:
 Precio: 4,56 - NO encontrado

19. Crear una **Aplicación Java** llamada **ArrayEnteroBusqueda** que:

- Declare un array **de tipo entero** de 5 posiciones.
- Rellene el array con valores aleatorios.
- Pida al usuario la entrada de un valor en el formato:
 Buscar: _
- Muestre por pantalla el contenido del array en el formato:
 Aforo actual ...: 123 personas
 Aforo actual ...: 321 personas
 Aforo actual ...: 975 personas
 ...
- Si el valor introducido SI se ha encontrado en el array expresarlo en el formato:
 Aforo: 321 - SI encontrado - Posición 1
- Si el valor introducido NO se ha encontrado en el array expresarlo en el formato:
 Aforo: 456 - NO encontrado

Operaciones con Arrays

20. Crear una **Aplicación Java** llamada **ArrayAleatorioAcumulado** que:

- Declare un array **de tipo real** de 5 posiciones.
- Rellene el array con valores reales aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Dato 1 ...: 1.23
Dato 2 ...: 3.21
Dato 3 ...: 9.87
...
• Muestre por pantalla la **suma** de los valores del array en el formato:
Total: 45.67

21. Crear una **Aplicación Java** llamada **ArrayUsuarioMedia** que:

- Declare un array **de tipo real** de 5 posiciones.
- Recorra el array para pedir al usuario que **introduzca por teclado** un valor arbitrario para cada posición en el formato:
Dato 1 ...: _
- Muestre por pantalla el contenido del array en el formato:
Dato 1 ...: 1.23
Dato 2 ...: 3.21
Dato 3 ...: 9.87
...
• Muestre por pantalla la **media** de los valores del array en el formato:
Media: 2.46

22. Crear una **Aplicación Java** llamada **ArrayAleatorioMaximo** que:

- Declare un array **de tipo real** de 5 posiciones.
- Rellene el array con valores reales aleatorios.
- Muestre por pantalla el contenido del array en el formato:
Dato 1 ...: 1.23
Dato 2 ...: 3.21
Dato 3 ...: 9.87
...
• Muestre por pantalla el **valor máximo** de los valores del array en el formato:
Máximo ...: 9.87 – Posición: 2

23. Crear una **Aplicación Java** llamada **ArrayUsuarioMinimo** que:

- Declare un array **de tipo real** de 5 posiciones.
- Recorra el array para pedir al usuario que **introduzca por teclado** un valor arbitrario para cada posición en el formato:
Dato 1 ...: _
- Muestre por pantalla el contenido del array en el formato:
Dato 1 ...: 1.23
Dato 2 ...: 3.21
Dato 3 ...: 9.87
...
• Muestre por pantalla el **valor mínimo** de los valores del array en el formato:
Mínimo ...: 1.23 – Posición: 0

Supuestos Prácticos

24. Crear una **Aplicación Java** llamada **TestSemana** que:

- **Defina** una clase de librería llamada **UtilesSemana** que:
 - **Defina** las constantes enteras:
 - **CODIGO_ESP** – 34
 - **CODIGO_ING** – 44
 - **CODIGO_ALE** – 49
 - **Defina** las constantes de texto:
 - **NOMBRE_ESP** – "Español"
 - **NOMBRE_ING** – "Inglés"
 - **NOMBRE_ALE** – "Alemán"
 - **Defina** los siguientes arrays constantes de texto:
 - **DIAS_ESP** – Nombre de los días de la semana en **español**.
 - **DIAS_ING** – Nombre de los días de la semana en **inglés**.
 - **DIAS_ALE** – Nombre de los días de la semana en **alemán**.
 - **Defina** un método con la siguiente signature:


```
public static String obtenerNombreDia(int dia, int idioma);
```


en donde **se devuelve el nombre del día de la semana** indicado por el **número de día de la semana** (Lunes ≡ 1) y el **código de idioma**.
- **Permita introducir** al usuario por teclado el **número de su día de la semana favorito** y el **código de idioma de salida** y el programa muestre por la pantalla el **nombre de su día de la semana favorito** en el **idioma elegido** y en el siguiente **formato**:

```
Número de día ....: 5  
Idioma elegido ...: 44  
Nombre de día ....: Friday (Inglés)
```
- En caso de **error** el programa tendrá la siguiente salida:

```
Número de día ....: 123  
Idioma elegido ...: 44  
Nombre de día ....: Desconocido
```

25. Crear una **Aplicación Java** llamada **TestPrimitiva** que:

- **Defina** una clase de librería llamada **UtilesPrimitiva** que:

- **Defina** las constantes enteras:

NUMEROS_DISPONIBLES – 49

NUMEROS_UTILIZABLES – 6

- **Defina** un método con la siguiente signatura:

```
public static void generarApuestaPrimitiva(int[] sorteo);
```

en donde se recibe un **array** para guardar la lista de **números de una apuesta de Lotería Primitiva**. Estos números tienen que estar **dentro del rango correcto** y deben de ser **diferentes entre sí**. El método tiene que **generar aleatoriamente** estos números.

- **Permita** generar aleatoriamente una apuesta de Lotería Primitiva y la muestre por pantalla, primero **NO ordenada** y luego **SI ordenada**, en el siguiente formato:

APUESTA DE LOTERÍA PRIMITIVA

=====

Apuesta NO ordenada

Número 1: 21

Número 2: 44

Número 3: 17

Número 4: 28

Número 5: 3

Número 6: 32

Apuesta SI ordenada

Número 1: 3

Número 2: 17

Número 3: 21

Número 4: 28

Número 5: 32

Número 6: 44