



Nombre y Apellidos:

Instrucciones. La tarea constará de diferentes ejercicios de manejo de arrays y matrices. Debéis realizar un proyecto como se indica a continuación.

Detalles de la tarea de esta unidad.

La tarea se entregará en el aula virtual en un ZIP **tareaTema4NombreApellidos.Zip** conteniendo los ejercicios contestados. Crear un proyecto en eclipse **tareaTema4NombreApellidos**. **Para el ejercicio 1 y 2 en Java** se guardará en una clase diferente como **EjercicioXNombreApellidos.java** si es el **Ejercicio 1**, etc.

Para el ejercicio 3 crear los paquetes y las clases como se indica en el enunciado.

Resultados de aprendizaje

RAE 6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Ejercicio 1. Manejo de arrays. (2,5 puntos) .

Indicador 54

Escribe una función “ConcatenaArraysPro” a la que le pasas dos arrays de enteros y te devuelve un array cuyo tamaño es la suma del tamaño de ambos y que contiene todos los elementos del primero y a continuación los del segundo. Esta función devolverá un array de enteros (*int[]*) el cuál se creará dentro de la propia función.

Ej. de uso: `int[] a = {1,2,3};`

`int[] b = {4,5,6,7,2};`

`int[] c;`



Nombre y Apellidos:

c = ConcatenaArraysPro(a, b);

Ejemplo de ejecución

c contiene 1|2|3|4|5|6|7|2

Ejercicio 2. Manejo de matrices (3,5 puntos)

Indicador 54, 59

Dadas las siguientes matrices se pedirán en los siguientes enunciados una serie de operaciones.

//DEFINIRLAS COMO VARIABLES DE CLASE, DESPUES DE LA CABECERA DE LA CLASE

```
static String[] cursos = { "1DAM", "2DAM", "1SMR", "2SMR" };
```

```
static String[][] asignaturas = {
```



Nombre y Apellidos:

{"PROG", "ENT", "LM", "SI", "ING", "FOL"},//asignaturas 1DAM

{"ADAT", "PSP", "SGE", "PMDM", "EIE", "DI"},//asignaturas 2DAM

{"AWEB", "SOM", "MMEQ", "ING", "REDL"},//asignaturas 1SMR

{"APOF", "SRED", "SEG"},//asignaturas 2SMR

};

static String[][]alumnos = {

{ "Ana", "Maria", "Pedro", "Jorge" }, //ALUMNOS DE 1DAM

{ "Luisa", "Sofia", "Javier", "Felipe" }, //ALUMNOS DE 2DAM

{ "Andrés", "Martin" }, //ALUMNOS DE 1SMR

{ "Antonio", "Josefa", "Manuel"}, //ALUMNOS DE 2SMR

};

- a) PROCESO REPETITIVO EN EL QUE SE PIDE POR TECLADO UN NOMBRE DE ALUMNO Y SE DEBE MOSTRAR SU CURSO Y SUS ASIGNATURAS, SI EL ALUMNO NO EXISTE MOSTRAR UN MENSAJE INDICÁNDOLO. EL PROCESO FINALIZA CUANDO EL NOMBRE ES *. EJEMPLO DE EJECUCIÓN:

Nombre de alumno (*) para finalizar: sss
No existe el Alumno

Nombre de alumno (*) para finalizar: Ana
CURSO: 1DAM
Las Asignaturas del Alumno son: [PROG, ENT, LM, SI, ING, FOL]

Nombre de alumno (*) para finalizar: Sofia
CURSO: 2DAM
Las Asignaturas del Alumno son: [ADAT, PSP, SGE, PMDM, EIE, DI]

Nombre de alumno (*) para finalizar: Manuel
CURSO: 2SMR
Las Asignaturas del Alumno son: [APOF, SRED, SEG]



Nombre y Apellidos:

Nombre de alumno (*) para finalizar: Josefina
No existe el Alumno

Nombre de alumno (*) para finalizar: *

Ejercicio 3. Array de objetos (4 puntos)

Indicador 54, 59

CREA UN PAQUETE LLAMADO datos, donde crearás la clase Alumno

CREA UN PAQUETE LLAMADO proceso, donde crearás las clases (que tienen el método main()) que usan Alumno

A partir de la siguiente clase **Alumno** que tiene esta estructura:

```
public class Alumno {  
    private String nombre;  
    private int nota;  
    private String curso;  
    //constructor vacio  
    //constructor con parámetros  
    //métodos get y set para acceso a los atributos  
}
```

Crea una clase AlumnosPrincipal con su método main() para probar. Y añádele los siguientes métodos estáticos indicados en el enunciado

Ejemplo1: Construimos un array para 4 objetos **Alumno**:

```
Alumno [] alumno = new Alumno [4];
```

Llenamos cada posición del array creando instancias de la clase Alumno:

```
alumno[0] = new Alumno("Ana", 5, "1DAM");  
alumno[1] = new Alumno("Luis", 6, "2DAM");  
alumno[2] = new Alumno("Pedro", 5, "1SMR");  
alumno[3] = new Alumno("Marta", 7, "2SMR");
```

//DESPUES DE INICIALIZAR EL ARRAY LLAMAR A ESTOS MÉTODOS ESTATICOS (LOS TIENES QUE CREAR Y DESARROLLAR):

```
VisualizarAlumnos(alumno); //VISUALIZAR LOS DATOS DE TODOS LOS ALUMNOS  
//SI LA NOTA ES 5 ponemos aprobado  
// SI LA NOTA ES 7 o mayor notable  
// SI LA NOTA ES 9 o mayor sobresaliente
```



Nombre y Apellidos:

```
ModificarAlumnos(alumno); // subirá un punto a la nota del alumno
                           // los pondrá en el curso superior
                           // si han aprobado
```