

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>				
Centro Educativo <b>IES Campanillas (sede PTA)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>			Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Examen</b>	Trimestre <b>Segundo – Control 4</b>	Modelo <b>Único</b>	Fecha <b>12/02/2019</b>	Pág. <b>1/4</b>

## INSTRUCCIONES

- El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por "Ex" seguido del número de lista, seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el turno. También debe indicar una breve descripción de lo que hace el programa.
- Únicamente se necesita entregar el código fuente en java, **no se deben entregar los archivos con la extensión .class**.

## EJERCICIOS

1. [2,5 puntos]

OPCIÓN 1: Realice un programa en JSP para presentar en una página web una Criba de Eratóstenes como sigue:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

En esta tabla, se han de representar coloreados de gris los números no primos desde 1 hasta el número que indique el usuario. Los números primos irán coloreados en el color que indique el usuario (a elegir entre varios). Por tanto, desde el formulario inicial, el usuario indicará hasta qué número quiere llegar, y el color con el que quiere colorear los números primos.

=0=0=0=0=0=0=

OPCIÓN 2: Realice un programa en JSP que represente en una tabla los números desde el 1 hasta el número que indique el usuario, de manera que los números que contengan un determinado dígito (0-9) aparezcan de color rojo, y los que no contengan dicho dígito, aparezcan de color negro. El usuario indicará desde un formulario el dígito que quiere señalar, la cantidad de números que se representarán por cada fila de la tabla, y el número hasta el que se quiere llegar. Por ejemplo, si el usuario indica dígito 2, 10 números por fila y llegar hasta el número 30, la tabla resultante sería así:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>				
Centro Educativo <b>IES Campanillas (sede PTA)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>			Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Examen</b>	Trimestre <b>Segundo – Control 4</b>	Modelo <b>Único</b>	Fecha <b>12/02/2019</b>	Pág. <b>2/4</b>

2. [2,5 puntos]

OPCIÓN 1: Escriba un programa que implemente un diccionario de parejas de sinónimos. Este diccionario emparejará palabras. Por tanto, en este diccionario, cada palabra tendrá un solo sinónimo. Su programa debe ser capaz de hacer lo siguiente:

- Dar de alta parejas de sinónimos. En el caso de que la primera palabra, que hace las veces de índice, ya esté en el diccionario, el programa debe informar al usuario de este hecho y solicitarle confirmación para actualizar su sinónimo.
- Dar de baja parejas de sinónimos (a partir de la primera palabra, que hace las veces de índice).
- A partir de una palabra dada, deducir todos sus sinónimos. Para ello, una vez obtenga el primer sinónimo de una palabra indicada por el usuario, deberá comprobar si dicho sinónimo tiene, a su vez, otro sinónimo en este diccionario, debiéndolo mostrar en ese caso. Y así sucesivamente hasta que no se encuentren más sinónimos o “comiencen a repetirse”. Para comprobar repeticiones, puede usar la estructura de datos que desee, aunque se valorará más el uso de un ArrayList que otro tipo de estructuras de datos.

Se valorará la subdivisión del código en funciones/métodos, de manera que el programa esté subdividido modularmente, facilitando su lectura y comprensión por parte de otro programador. Las tres funcionalidades que debe implementar su programa deben estar accesibles a través de un menú.

=0=0=0=0=0=

OPCIÓN 2: Escriba un programa que memorice, para una palabra clave determinada, una frase que esté relacionada con dicha palabra. Para ello, deberá hacer uso de un diccionario, donde la palabra clave hará las veces de índice. El programa deberá realizar las siguientes tareas:

- Mostrar la frase correspondiente a una palabra. Para ello, el programa solicitará al usuario una palabra y si dicha palabra está en el diccionario, mostrará la frase correspondiente, y si no está solicitará al usuario una frase para, a continuación, darla de alta en el diccionario.
- Borrar una palabra y su correspondiente frase, para lo cual se le solicitará al usuario la palabra correspondiente.
- Mostrar por pantalla todo el diccionario (palabras y frases asociadas).

Se valorará la subdivisión del código en funciones/métodos, de manera que el programa esté subdividido modularmente, facilitando su lectura y comprensión por parte de otro programador. Las tres funcionalidades que debe implementar su programa deben estar accesibles a través de un menú.

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>				
Centro Educativo <b>IES Campanillas (sede PTA)</b>		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>			Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Examen</b>	Trimestre <b>Segundo – Control 4</b>	Modelo <b>Único</b>	Fecha <b>12/02/2019</b>	Pág. <b>3/4</b>

3. [2,5 puntos]

OPCIÓN 1: Haciendo uso de ArrayList, cree una lista de Personas, para lo cual tendrá que definir la clase oportuna. Su programa debe implementar, mediante opciones guiadas a través de un menú, la “cola” de una caja de supermercado. Sólo nos interesan los datos de la persona: DNI y nombre. Por tanto, tendrá que desempeñar las siguientes funciones:

- Llegada de una persona a la cola (se pondrá al final de dicha cola)
- Atención de la persona a la que le toca de la cola (saldrá de la cola la primera que esté en cola), si es que hay gente en cola
- Presentar por pantalla una lista con los nombres de las personas que están en cola

Se valorará la subdivisión del código en funciones/métodos, de manera que el programa esté subdividido modularmente, facilitando su lectura y comprensión por parte de otro programador. Las tres funcionalidades que debe implementar su programa deben estar accesibles a través de un menú.

=0=0=0=0=0=0=

OPCIÓN 2: Haciendo uso de ArrayList, implemente una pila de prendas de vestir. De las prendas de vestir sólo nos interesa su código, descripción y talla. Cuando uno se viste, se pone determinadas prendas en un orden, y cuando se desviste, lo hace en orden inverso. Es decir, lo último que uno se pone es lo primero que se quita (supondremos que todas las prendas van unas encima de otras; que son “de cuerpo entero”). Deberá definir una clase Prenda, con la que creará los objetos del ArrayList. Su programa deberá desempeñar las siguientes funciones:

- Ponerse una prenda de vestir
- Quitarse una prenda de vestir, informando de qué prenda es la que se ha quitado (sólo la descripción). Si no hay prendas que quitarse, indicará que se está desnudo.
- Mostrar la prenda que está ahora visible (si es que no se está desnudo), así como el n.º de prendas que se tienen puestas.

Se valorará la subdivisión del código en funciones/métodos, de manera que el programa esté subdividido modularmente, facilitando su lectura y comprensión por parte de otro programador. Las tres funcionalidades que debe implementar su programa deben estar accesibles a través de un menú.

Debemos preparar la clase Prenda para que a la hora de crear las prendas, podamos crearlas indicando el código, el nombre y, opcionalmente, la talla (si no se indica la talla, se entenderá que es la talla L. En este programa, podrá crear las prendas sin especificar la talla.

Familia Profesional <b>Informática y Telecomunicaciones</b>		Nombre del Ciclo Formativo <b>Título de Técnico Superior en Desarrollo de Aplicaciones Web</b>				
Centro Educativo <b>IES Campanillas</b> (sede PTA)		Módulo Profesional <b>Programación</b> Código: <b>0485</b> N.º de créditos ECTS: <b>14</b>			Profesor <b>Juan Antonio Jiménez Morales</b>	
Curso lectivo <b>2018 / 2019</b>	Grupo <b>1º DAW</b>	Tipo de documento <b>Examen</b>	Trimestre <b>Segundo – Control 4</b>	Modelo <b>Único</b>	Fecha <b>12/02/2019</b>	Pág. <b>4/4</b>

4. [2,5 puntos]

OPCIÓN 1: Escribe un programa para una biblioteca que contenga libros y revistas.

- Las características comunes que se almacenan tanto para las revistas como para los libros son el código, el título, y el año de publicación. Estas tres características se pasan por parámetro en el momento de crear los objetos.
- Los libros tienen además un atributo prestado. Los libros, cuando se crean, no están prestados.
- Las revistas tienen un número. En el momento de crear las revistas se pasa el número por parámetro.
- Tanto las revistas como los libros deben tener (aparte de los constructores) un método toString() que devuelve el valor de todos los atributos en una cadena de caracteres. También tienen un método que devuelve el año de publicación, y otro el código.
- Para prevenir posibles cambios en el programa se tiene que implementar una interfaz Prestable con los métodos prestar(), devolver() y prestado(). La clase Libro implementa esta interfaz. El método prestado() devuelve true si el libro está prestado, y false si no lo está.

Para comprobar el funcionamiento de estas clases, puede escribir un pequeño programa no interactivo en el que cree algunos objetos sobre los que testear.

=0=0=0=0=0=0=

OPCIÓN 2: Escriba un programa para tienda que gestione regalos por San Valentín. Sólo vende ramos de flores y perfumes.

- Ambos tipos de regalos tendrán un código, una denominación, y un precio de venta, datos que se especifican cuando se crean ramos o perfumes.
- Los ramos de flores tienen como característica adicional el n.º de flores que componen el ramo y el tipo de flor. Por defecto, si no se especifican estos datos cuando se crea el ramo, el n.º de flores es 6 y el tipo de flor es rosa. Pero de debe preparar la clase para que puedan o no indicarse alguno o ninguno de estos dos parámetros (n.º de flores y tipo de flor).
- Los perfumes tienen, por su parte, un aroma, y hay que indicar también el año de creación el perfume. Estos parámetros hay que indicarlos cuando se crea el perfume.
- Ambos tipos de objetos deben tener un método toString(), que devuelve el valor de todos los atributos del objeto en formato cadena de caracteres. También habrá métodos para decirnos: el n.º de flores del ramo y el tipo de flor, así como el aroma del perfume.
- Ambos tipos de objetos deben implementar una interfaz Vendible con los métodos precio(), fijarPrecio(<valor>), que nos aseguran que en ambos objetos podremos consultar su precio y cambiarlo.

Para comprobar el funcionamiento de estas clases, puede escribir un pequeño programa no interactivo en el que cree algunos objetos sobre los que testear.