

CICLO: DESARROLLO DE APLICACIONES MULTIPLATAFORMA	CALIFICACIÓN
MÓDULO: PROGRAMACIÓN	
FECHA: 21/02/2023	
NOMBRE Y APELLIDOS:	

EXAMEN 2ª EVALUACIÓN

Lea detenidamente las siguientes **instrucciones** antes de realizar el examen.

•El examen tiene una **duración de 1 hora y 30 min**

•Durante la realización del examen:

- no **se podrá consultar Internet, ni abrir apuntes u otros ejemplos** o ejercicios que no sean los propios del examen

- **solamente se podrá mantener abierto Visual Studio Code** para la realización de los ejercicios (o algún otro editor de código similar como Netbeans) y el Aula Virtual para subir los ejercicios al buzón de entrega.

- tampoco será posible salir de clase durante el examen, a no ser por una situación excepcional.

- el **móvil debe permanecer apagado** y fuera del alcance del alumno. S

El **no cumplimiento de las normas anteriores es motivo de suspenso**.

•El examen consta de 3 ejercicios prácticos, indicando en cada uno de ellos su puntuación

•Una vez terminado, deberás subir al buzón del Aula Virtual un fichero "Apellidos_Nombre.zip" donde estén los ficheros: Ejercicio1_Apellidos.java, Ejercicio2_Apellidos.java y Ejercicio3_Apellidos.java

Para todos los ejercicios será imprescindible aplicar las buenas prácticas de programación: tabular el código, utilizar identificadores apropiados, poner aclaraciones mediante código si fuera necesario y escribir de forma legible.

Todos los ejercicios deberán compilar sin errores y cumplir todos los requisitos del enunciado, teniendo en cuenta todas las opciones que el usuario pueda escribir.

NO es necesario que incluyas comentarios javadoc.

Ejercicio 1 [3 puntos]

Crea una clase llamada *Cuenta*, cuyo objetivo es gestionar las operaciones bancarias habituales. La clase deberá cumplir los siguientes requisitos:

- Almacenar: el número de la cuenta (String), el saldo de la cuenta, su fecha de apertura y el nombre completo del titular o titulares de la cuenta (se deberán almacenar en un ArrayList).
- Poder *ingresar* en la cuenta una cantidad de dinero
- Poder *sacar* de la cuenta una cantidad de dinero, siempre y cuando exista saldo suficiente en la misma.
- Poder *consultar el saldo* de la cuenta

La clase deberá tener sus correspondientes métodos getters/setters y dos constructores:

- Un constructor que inicializa la cuenta con valores para todos sus atributos, siendo la fecha de apertura la fecha actual del sistema y el saldo 0.
- Un constructor que inicializa la cuenta con valores para todos sus atributos, siendo la fecha de apertura la fecha actual del sistema y el saldo pasado como parámetro.

Nota:

- Se recuerda que en el paquete java.time dispones de la clase LocalDate para trabajar con fechas.
- LocalDate.now(); nos devuelve la fecha actual del sistema.

CICLO: DESARROLLO DE APLICACIONES MULTIPLATAFORMA	CALIFICACIÓN
MÓDULO: PROGRAMACIÓN	
FECHA: 21/02/2023	
NOMBRE Y APELLIDOS:	

Ejercicio 2 [3 puntos]

Crea una clase llamada Cajero, que contiene el método main() que permitirá gestionar una cuenta bancaria y hacer operaciones con la misma.

Al ejecutar el programa, se presentará un menú:

- 1.- Crear cuenta con saldo 0.
- 2.- Crear cuenta con saldo inicial.
- 3.- Ingresar dinero.
- 4.- Sacar dinero.
- 5.- Ver saldo.
- 6.- Salir

- La opción 1 solicita la información necesaria para crear un objeto Cuenta con saldo 0.
- La opción 2 solicita la información necesaria para crear un objeto Cuenta con un saldo inicial distinto de cero
- En la opción 3 se solicita una cantidad y la ingresa en el objeto creado en las opciones 1 ó 2 (debe haber pasado antes por estas opciones).
- En la opción 4 se solicita una cantidad y la extrae del objeto creado en las opciones 1 ó 2 (también debe haber pasado antes por las opciones 1 ó 2).
- La opción 5 muestra el saldo de la cuenta.
- La opción 6 finaliza el programa.
- El menú se vuelve a presentar en pantalla mientras no se elija la opción Salir.

Ejercicio 3 [4 puntos]

Se desea desarrollar un programa para la gestión de publicaciones en una biblioteca. Todas las publicaciones deben tener asociado un autor, un título y un año de publicación. Deben considerarse tres tipos de publicaciones distintas: libros, revistas y tesis doctorales.

- En el caso de los libros se debe agregar el nombre de la editorial correspondiente.
- Las revistas deben añadir el número (int) y el volumen (int).
- Las tesis doctorales deberán incluir la Universidad y la calificación (int).

Se deben codificar las clases Publicación, Libro, Revista y Tesis (dentro del paquete publicaciones) con los atributos mencionados, los métodos getters/setters correspondientes, el método toString o mostrar los valores de los atributos de cada objeto y dos constructores para cada clase.