	IES LUIS VIVES - 1º DAW - **EXAMEN 2ª EVALUACIÓN** Módulo "Programación"	Curso 2022/2023 Fecha mar/23
Nombre y Apellidos		Nota:

Parte de Teoría (20%)

Cada pregunta tiene una o varias respuestas válidas.

Cada pregunta vale 0,5 puntos y si en una pregunta hay n respuestas correctas y el alumno marca p respuestas correctas la puntuación obtenida en la pregunta será de p*0,5/n puntos. Teniendo en cuenta que se eliminará una respuesta correcta por cada respuesta incorrecta marcada.

El valor mínimo de una pregunta es 0 puntos.

1.-Señala las afirmaciones correctas para las versiones actuales de java.

a) Las instrucciones String s="Hola"; s=s.concat("Ana"); crean 2 objetos
b) Las instrucciones StringBuffer sb=new StringBuffer("Hola"); sb.append("Ana"); crean 2 objetos
c) StringBuffer sb=new StringBuffer("Hola"); int x=sb.capacity(); guarda en x el valor 4
d) StringBuffer sb= new StringBuffer("Hola Juan"); int x=sb.substring(4,7).indexof("a"); guarda en x el valor -1;
e) La instrucción c='3'; int y=c+5; guarda en y el valor 8

2.-Indicar que afirmaciones son CIERTAS para el lenguaje java para la siguiente declaración de un array

```
int v[][][]=new int[4][3][];
```

a) La declaración es incorrecta.
b) La declaración es correcta y también lo sería realizar después lo siguiente `v[2][2]=new int[2];`
c) La declaración es correcta y también lo sería realizar después lo siguiente `v[v.length][v.length]=new int[2];`
d) La declaración es correcta y también lo sería realizar después lo siguiente `v[2][2]=2;`
e) La declaración es correcta y también lo sería realizar después lo siguiente `v[v.length][v.length]=2;`
f) Ninguna de las anteriores es correcta.

3.-Indicar que aparece en pantalla al ejecutar el siguiente código

```
int r=0;
for (int i=1, j=0; i<=9; i=i+4,j=j+2) {
     try {
          r+= i/j;
     }catch (ArithmeticException e) {
          r=20;
     }catch (Exception e) {
          r=10;
     }
}
System.out.println(r);
```

a) Al ejecutarlo aparece en pantalla un mensaje de error
b) Al ejecutarlo aparece en pantalla un 0
c) Al ejecutarlo aparece en pantalla un 20
d) Al ejecutarlo aparece en pantalla un 10
e) Al ejecutarlo aparece en pantalla un 14
f) Al ejecutarlo aparece en pantalla un 24
g) Al ejecutarlo aparece en pantalla un 16
h) Al ejecutarlo aparece en pantalla un 26
i) Ninguna de las anteriores es cierta.

4.-Indicar que afirmaciones son correctas para el lenguaje Java

a) Se puede utilizar la clase File para crear ficheros porque tiene un método para ello
b) La clase FileInputStream posee un método para leer números reales (readDouble ())
c) La clase DataInputStream posee un método para leer números reales (readDouble ())
d) La clase ObjectInputStream permite leer objetos completos con el método readObject aunque, como el método readObject devuelve un objeto es necesario hacer una conversión de lo que devuelve el mismo al tipo del objeto real que se lee.
e) La clase RandomAccessFile permite leer y escribir en registros intermedios de un fichero. Requiere que se utilice siempre el método seek para poder realizar una lectura o una escritura.
f) Ninguna de las afirmaciones anteriores es correcta.

Solución

Ej1: a,d,

2.- b.- La declaración es correcta y también lo sería realizar después lo siguiente `v[2][2]=new int[2];`

3.- f. sale 24

4.- a,c,d

IES LUIS VIVES - 1º DAW **- EXAMEN 1ª EVALUACIÓN**
Módulo "Programación" -- **Parte práctica (80%)**

1.- (3.5 ptos) Se tiene un fichero de texto con los datos de ocupación de un vuelo. El nombre del fichero tiene la palabra vuelo seguida del número de vuelo y la extensión .txt. Por ejemplo **vuelo3423.txt**

En la primera línea está, con el siguiente formato, los datos del vuelo:

númeroVuelo#numeroFilas#numeroAsientosPorFila

Todos los campos son números enteros. Por ejemplo 3423#20#6 indica que hay 20 filas y cada fila tiene 6 asientos. Se supondrá que, si el fichero existe, no hay errores en dicha línea

En las filas siguientes están las plazas vendidas en el vuelo con el formato:

fila;asiento;numAsientos;dniComprador

donde se indica la fila, el primer asiento de esa fila vendido, y cuantos asientos consecutivos se han vendido. Por ejemplo 3;4;2;34765436A indica que se han vendido, de la fila 3, los asientos 4 y 5 al cliente 34765436A

Realizar una aplicación que permita realizar lo siguiente.

A.-Al empezar la ejecución se le pedirá al usuario el número de vuelo y el programa mirará si existe un fichero para ese vuelo. Si no existe se pedirá el número de filas y el número de asientos por fila y se creará el fichero correspondiente con la primera línea del mismo incluida.

B.-Si el fichero existe se comprobará que todas las ventas que hay en el fichero son correctas (es decir que no se han vendido asientos que no existen o asientos vendidos en una venta anterior) y se mostrará por pantalla el número de todas las líneas incorrectas, el texto de la línea y el error. Hay 3 tipos de errores que se comprobarán en este orden, no es necesario comprobar errores posteriores si se ha detectado uno previo:

- Fila incorrecta. El número de fila no existe en el avión.
- Asiento incorrecto, ese número de asiento no existe
- Asiento vendido, alguno de los asientos de la venta ya está ocupado por una venta anterior.

Por ejemplo, si el vuelo tiene 20 filas y 5 asientos por fila una línea como 23;4;3;34765436A sería incorrecta porque la fila 23 no existe y porque, de existir, estaría indicando que se han vendido los asientos 4,5 y 6 de la fila 23 y solo hay 5 asientos por fila. Para ese ejemplo se mostraría el error de Fila incorrecta.

Se recomienda utilizar un array bidimensional para ir colocando las plazas vendidas y detectar los errores de doble venta de asientos.

C.-Si el fichero es correcto, se pedirá un número de fila, uno de columna, un número de asientos y el dni y, si esos asientos están libres y son correctos se añadirá, al final del fichero, una línea con dicha venta. Si no están libres se indicará :"Asientos ocupados". Esta operación se realizará directamente en el fichero.

Tanto el número de fila como el de asiento comienzan en '0.

2.- (1,5 ptos) Realizar una **función recursiva int contarDígitos(int numero, int dígito)** que, sin utilizar ningún bucle, devuelva el número de dígitos que hay en el número iguales al dígito recibido como parámetro. Por ejemplo, si se invoca con …contarDigitos (20145100 , 0) devolverá 3.

No se podrá hacer uso de variables globales o atributos de la clase en la que esté codificado el método indicado.

3.- (3 ptos) Para gestionar los viajes en transporte público se tienen las siguientes posibilidades de adquisición de títulos de transporte.

- Adquirir un **pase anual** que sirve para el año en el que se adquiere el billete (si se adquiere el 31 de diciembre sólo servirá para un día (el 31 de diciembre). El precio será de 1 € por cada día que quede desde la adquisición al final del año incluido el día de adquisición.
- Adquirir un billete para **un viaje de 90 minutos**. Precio de 1 €. Es válido desde que se adquiere y durante 90 minutos.

Se tiene un fichero de **acceso directo "billetes.dat"** en el que se almacena, en este orden y con tipo int para los números: el **número del billete** (que es un número entero que se **generará automáticamente**

por la aplicación empezando en 0 y continuando de forma consecutiva e indica la posición del billete en el fichero), el **tipo de billete** (**1- para anual, 2- para 1 viaje de 90 minutos**), información adicional sobre el billete (**año, mes, día, hora y minuto de adquisición y activo o gastado** (valor 1 para activo y 0 para gastado)) y el **dni** del **cliente** que adquirió el billete.
Cada vez que un cliente hace **uso del billete** en la entrada de un medio de transporte en una **máquina**, esta **escribe una línea** en un fichero de texto **"viajes.txt"** con el **número** del **billete** y la fecha (**año, mes, día, hora, minuto**) de la operación **separando** cada **campo** por **#.** Ejemplo de línea para un viaje de hoy a las 15:32h. : *567#2023#3#13#15#32*
Para simplificar se supondrá que **desde** las **00:00h. hasta las 06:00h** los transportes **no** pueden realizar **viajes** y si se adquiere un billete de 90 minutos a las 23:59 sólo se podrá utilizar durante 1 minuto. Es decir, los **billetes de tipo 2 sólo valen para 90 minutos**.

Se deben **codificar los métodos** con la siguiente funcionalidad para la aplicación en la clase Transporte, se podrán crear clases adicionales o añadir métodos a la clase Fecha:
A.- **Adquirir billetes** de transporte. Se pedirá el tipo de billete y los datos del mismo por pantalla excepto la **fecha y hora que se necesiten que se tomarán del sistema operativo** haciendo uso de la clase:

```
public class Fecha {
  int año, mes, dia, hora, minuto;
  //Crea un objeto Fecha con año, mes, día, hora y minuto del momento en el que se ejecuta
  Fecha() {
              Calendar calendario=Calendar.getInstance();  //Calendar: clase abstracta
              Date fecha=calendario.getTime();
              calendario.setTime(fecha);
              año=calendario.get(Calendar.YEAR);
              mes=calendario.get(Calendar.MONTH)+1;
              dia=calendario.get(Calendar.DAY_OF_MONTH);
              hora=calendario.get(Calendar.HOUR_OF_DAY);
              minuto=calendario.get(Calendar.MINUTE);
      }
}
```

B.- **Detectar** los **billetes** utilizados **irregularmente**. Se procesará el fichero *viajes.txt* y se comprobará que el billete era válido para ese viaje. Se creará un fichero ***NoValidos.txt*** con el **dni**, **número** del billete y fecha (**año, mes, día, hora, minuto**) del viaje, separado cada campo por **#**. Si el billete era de **tipo 2**, al procesar el billete, se **marcará como utilizado** en el fichero de billetes.
C.- Pedir el **dni de un cliente** y **mostrar en pantalla el precio total** de todos los billetes adquiridos por ese cliente. Se supondrá que los billetes anuales valen 100€ y los de 90 minutos valen 1€

Se supondrá que los ficheros existen y no hay error en el formato de los mismos ni en los datos (por ejemplo no se puede fichar con un billete antes de ser comprado)

En todos los ejercicios los datos pedidos al usuario se leerán de consola (System.in). No se utilizará la entrada gráfica.
Se supondrá que los datos introducidos por el usuario tienen el formato correcto, es decir, si se pide un número el usuario introducirá un número.