Ejercicios de persistencia de objetos (I)

1. Hacer un programa que dado el nombre de un fichero, lo busque en una ubicación predeterminada, y con extensión ".txt"
2. Hacer un programa que lea un archivo "nombres.txt" en el que hay un nombre de persona por línea (sin tildes y todo en minúsculas), que dado un nombre proporcionado por argumento de entrada, me indique cuántas veces aparece ese nombre en mi fichero. Adaptar el programa para que funcione también con líneas que contengan múltiples nombres, separados por espacios.
3. Hacer un programa que lea un archivo "nombres.txt" en el que hay un nombre de persona por línea (sin tildes y todo en minúsculas), e informe de qué nombre es el que aparece más veces, y cuántas veces aparece. Si hay más de un nombre que se dispute el "honor" de ser el más repetido, responderá alguno de ellos.
4. Hacer un programa que busque todas las apariciones de una palabra (proporcionada por argumento de entrada) en el fichero "e.txt" y escriba un fichero "s.htm" en el que aparezca dicha palabra en negrita. (Pista: investigar el método **replaceAll(...)** de String)
5. Hacer un programa que lea un fichero "palabras.txt", que contenga líneas del estilo "palabra:color", y a continuación lea un fichero "original.txt" y cree un fichero "coloreado.htm" en el que cada palabra que aparezca en el fichero palabras.txt que esté también en el fichero "original.txt", aparezca coloreada con el color indicado. Para conseguirlo el fichero "coloreado.htm" irá acompañado de un fichero "estilo.css" que también generará el programa. Cada palabra encontrada, se marcará con un ****, y en el css se indicará el color de cada clase en función de lo indicado en el fichero "palabras.txt". Para mayor sencillez, el color indicado en este fichero estará escrito en inglés, tal como lo puede reconocer CSS y HTML
6. Hacer un programa llamado **copiar.java** que reciba por argumentos de entrada la ruta a un archivo a copiar y la ruta a donde quiera ser copiado el archivo. El programa realizará la copia binaria de dicho archivo.
7. Hacer un programa que lea los argumentos de entrada "args".
    - Si args[0]=='R' modo lectura, lee un fichero clientes.txt (dni:nom), crea objetos Persona(dni,nom), los colecciona y los muestra (redefiniendo toString())
    - Si args[0]=='W' modo escritura, lee pares de args, crea objetos Persona(dni,nom), los colecciona, los muestra y los escribe en clientes.txt(dni,nom)
    - ¿Qué ocurriría si quisiéramos añadir el atributo "peso" a los datos de Persona? ¿Cómo cambiaría el código?
    - ¿Y si quisiéramos añadir el padre de cada persona a sus datos?
8. Realizar el ejercicio anterior, pero utilizando un fichero "clientes.srl" en el que los datos de las personas estén serializados.
    - ¿Qué fichero tiene mayor tamaño? Probar con ficheros de 100 personas. Utilizar números de DNI y pesos grandes para comprobar la diferencia de tamaño.
9. Probar a añadir al fichero anterior un objeto de otra clase (por ejemplo String). ¿Sigue funcionando?