

Operaciones con ArrayList. Joshua Sangareau 1ºDAM.

1. Importar la información contenida en el documento notas1.txt, donde la información tiene el siguiente formato:

grupo;nombre,matemáticas;lengua;física;química;inglés

He creado un método que recibe un ArrayList de tipo alumno en este caso y un objeto de tipo File, el cual se leerá y se irán creando objetos de tipo alumno en el ArrayList recibido.

```
public static void lectorArchivos(ArrayList<Alumno> lista, File archivo) {  
    try {  
        Scanner lector = new Scanner (archivo);  
  
        while(lector.hasNextLine()) {  
            String linea = lector.nextLine();  
            String[] lineaSeparada = linea.split(";");  
            Alumno alumno = new Alumno(lineaSeparada[0], lineaSeparada[1], Integer.parseInt(lineaSeparada[2]),Integer.parseInt(lineaSeparada[3]),  
                Integer.parseInt(lineaSeparada[4]),Integer.parseInt(lineaSeparada[5]), Integer.parseInt(lineaSeparada[6]));  
            lista.add(alumno);  
        }  
        lector.close();  
    } catch (FileNotFoundException e) {  
        System.out.println(e.getMessage());  
    }  
}
```

Se crea el ArrayList, el objeto de tipo File y se llama al método lectorArchivos en el main.

```
//Apartado 1  
File archivo = new File("ArrayListArchivos/notas1.txt");  
ArrayList<Alumno> listaAlumnos = new ArrayList <>();  
lectorArchivos(listaAlumnos, archivo);
```

2. Mostrar el número de registros que contiene.

Se muestra la cantidad de registros llamando al método size()

```
//apartado 2  
System.out.println("Mostrando el tamaño de la lista tras la importacion: " + listaAlumnos.size());
```

Genera esta salida por consola:

```
Mostrando el tamaño de la lista tras la importacion: 129
```

3. Ordenar los registros por grupo y nombre.

Se implementa el interfaz Comparable en la clase Alumno y se ordena por grupo y nombre:

```
public class Alumno implements Comparable<Alumno>{
```

```
    @Override
    public int compareTo(Alumno o) {
        int devolucion = 0;
        if(grupo.compareTo(o.grupo)<0) {
            devolucion = -1;
        }else if(grupo.compareTo(o.grupo)>0) {
            devolucion = 1;
        }else {
            if(nombre.compareTo(o.nombre)< 0) {
                devolucion = -1;
            }else if(nombre.compareTo(o.nombre)> 0) {
                devolucion = 1;
            }
        }
        return devolucion;
    }
}
```

```
//apartado 3
Collections.sort(listaAlumnos);
escribirNuevoArchivo("ArrayListArchivos/AlumnosOrdenados.txt", listaAlumnos, false);
```

Se crea método escribirNuevoArchivo para generar un nuevo archivo y poder mostrar la lista ordenada como se solicita. El archivo se llama AlumnosOrdenados y estará dentro de la carpeta del proyecto en una carpeta llamada ArrayListArchivos.

```
public static void escribirNuevoArchivo (String nombreArchivo, ArrayList<Alumno> lista, boolean sobrescribir) {
    try {
        FileWriter archivoEscritura = new FileWriter(nombreArchivo, sobrescribir);
        for(Alumno i: lista) {
            if(sobrescribir == true) {
                archivoEscritura.write("\n" + i.getGrupo() + ";" + i.getNombre() + ";" + i.getMatematicas() + ";" + i.getLengua()
                    + ";" + i.getFisica() + ";" + i.getQuimica() + ";" + i.getIngles() + ";");
            }else {
                archivoEscritura.write(i.getGrupo() + ";" + i.getNombre() + ";" + i.getMatematicas() + ";" + i.getLengua()
                    + ";" + i.getFisica() + ";" + i.getQuimica() + ";" + i.getIngles() + ";" + "\n");
            }
        }
        archivoEscritura.close();
    }catch (IOException e) {
        e.printStackTrace();
    }
}
```

Este método recibe como parámetros la ruta relativa(con el nombre del archivo a crear), el ArrayList de tipo Alumno de donde cogerá los datos y un boolean para saber si se sobrescribe el archivo o se debe añadir la información a un archivo ya existente.

4. Insertar siguiente registro:

1º ESO D;López Muro, Rosario;3;3;4;4;3

Se genera un nuevo Alumno y se mete en una variable de tipo Alumno. Tras esto se añade a la lista de esta forma:

```
//apartado 4
Alumno nuevoAlumno = new Alumno("1º ESO D", "López Muro, Rosario", 3, 3, 4, 4, 3);
listaAlumnos.add(nuevoAlumno);
```

5. Insertar el siguiente registro en mitad de la lista:

1º ESO B;Duarte Durán, Alba;2;3;4;1;3

Se realiza el mismo procedimiento del apartado anterior, lo único que a la hora de introducirlo en la lista le indicamos la posición en la que lo debe insertar. (En mi caso divido entre 2 el tamaño de la lista y restándole 1 para que esté en el medio de esta. Se resta 1 porque las posiciones del ArrayList empiezan en 0)

```
//apartado 5
Alumno nuevoAlumno2 = new Alumno("1º ESO B", "Duarte Durán, Alba", 2, 3, 4, 1, 3);
listaAlumnos.add((listaAlumnos.size()/2)-1, nuevoAlumno2);
```

6. Mostrar la posición que ocupa el registro anterior insertado.

Se muestra la información por consola

```
System.out.println("Mostrando la posición del objeto añadido en el apartado 4: " + listaAlumnos.indexOf(nuevoAlumno));
System.out.println("Mostrando la posición del objeto añadido en el apartado 5: " + listaAlumnos.indexOf(nuevoAlumno2));
```

Genera esta salida:

```
Mostrando la posición del objeto añadido en el apartado 4: 130
Mostrando la posición del objeto añadido en el apartado 5: 65
```

7. Insertar los registros del documento notas2.txt

Se llama al método anteriormente mencionado escribirNuevoArchivo, esta vez con el nombre del archivo al que debe añadir la información, mandándole como parámetro true.

```
//apartado 7
escribirNuevoArchivo("ArrayListArchivos/notas2.txt", listaAlumnos, true);
```

8. Buscar el siguiente registro:

1º ESO B;Cordero Benítez, Alejandro;2;3;2;1;4

He creado un método que recibe la lista a recorrer, el nombre y el grupo del alumno a encontrar.

Genero un iterador para recorrer la lista y que se detenga cuando encuentre al alumno con el nombre indicado y el grupo indicado.

Devuelve un int con la posición en la que se encuentra en la lista:

```
public static int buscadorPosicionesAlumnos(ArrayList<Alumno> lista, String nombre, String grupo) {
    int devolucion = 0;
    Iterator<Alumno> iterador = lista.iterator();
    boolean encontrado = false;
    Alumno alumno;
    while(iterador.hasNext() && encontrado == false) {
        alumno = iterador.next();
        if(alumno.getNombre().equals(nombre) && alumno.getGrupo().equals(grupo)) {
            encontrado = true;
            devolucion = lista.indexOf(alumno);
        }
    }
    return devolucion;
}
```

Se llama al método desde el main:

```
//apartado 8
System.out.println("El registro de Alejandro Cordero Benitez se encuentra en la posicion: " + buscadorPosicionesAlumnos(listaAlumnos, "Cordero Benítez, Alejandro", "1º ESO B"));
```

Genera esta salida:

```
El registro de Alejandro Cordero Benitez se encuentra en la posicion: 26
```

9. Buscar el siguiente registro.

1º ESO C;Zoodsma Villalón, Natali;2;3;4;5;3

Se realiza el mismo procedimiento que el apartado anterior:

```
//apartado 9
System.out.println("El registro de Natali Zoodsma Villalón se encuentra en la posicion: " + buscadorPosicionesAlumnos(listaAlumnos, "Zoodsma Villalón, Natali", "1º ESO C"));
```

Genera esta salida:

```
El registro de Natali Zoodsma Villalón se encuentra en la posicion: 79
```

10. Borra el registro que está en la posición mitad más 1, indicando la información del registro que borras.

Se llama al método remove en la lista de alumnos y se le indica la posición del alumno a borrar. En este caso la mitad +1. Yo le pongo la posición resultado de dividir entre 2 el tamaño de la lista(ya que al empezar el la posición 0 el ArrayList ya estaría sumando 1)

```
//apartado 10
System.out.println("Borrando el alumno de la posicion mitad + 1: " + listaAlumnos.remove((listaAlumnos.size()/2)).toString());
```

Se muestra la información del alumno borrado:

```
Borrando el alumno de la posición mitad + 1: Alumno [grupo=1º ESO B, nombre=Duarte Durán, Alba, matematicas=2, lengua=3, fisica=4, quimica=1, ingles=3]
```

11. Borra los primeros 10 registros.

Genero un bucle para recorrer las 10 primeras posiciones y borrarlas.

(Existe un método `removeRange()` pero está definido como `protected` y no es visible. No se puede usar)

```
//apartado 11
for(int i = 0; i < 10; i++) {
    listaAlumnos.remove(i);
}
```

12. Borra todos los registros que tengan en matemáticas un 2. Indica cuántos registros borras.

He generado un método `borrarAlumnosConMatematicas2` que recibe la lista como parámetro y se borran los alumnos con un 2 en matemáticas. Se genera un iterador para poder borrar a la vez que se recorre el `ArrayList`.

```
public static int borrarAlumnosConMatematicas2(ArrayList<Alumno> lista) {
    Alumno alumno;
    Iterator<Alumno> iterador2 = lista.iterator();
    int devolucion = 0;
    while(iterador2.hasNext()) {
        alumno = iterador2.next();
        if(alumno.getMatematicas() == 2) {
            devolucion++;
            iterador2.remove();
        }
    }
    return devolucion;
}
```

Se llama al método en el main y se muestra por pantalla el resultado

```
//apartado 12
System.out.println("Se han borrado: " + borrarAlumnosConMatematicas2(listaAlumnos) + " registros");
```

Genera esta salida:

```
Se han borrado: 20 registros
```

13. Calcula la nota media para cada registro.

Se crea un método para calcular las medias y escribirlas en un nuevo archivo. Es muy parecido al método escribirNuevoArchivo pero este método añadirá las medias de cada registro en el archivo

```
public static void escribirArchivoMedias (String nombreArchivo, ArrayList<Alumno> lista, boolean sobrescribir) {
    try {
        FileWriter archivoEscritura = new FileWriter(nombreArchivo, sobrescribir); //si ponemos true se situa al final del
        for(Alumno i: lista) {
            int notasSumadas = i.getMatematicas() + i.getLengua() + i.getFisica() + i.getQuimica() + i.getIngles();
            double mediaNotas = (double)notasSumadas/5;
            archivoEscritura.write(i.getGrupo() + ";" + i.getNombre() + ";" + i.getMatematicas() + ";" + i.getLengua()
            + ";" + i.getFisica() + ";" + i.getQuimica() + ";" + i.getIngles() + ";" + "NOTA MEDIA: " + mediaNotas + "\n");
        }
        archivoEscritura.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Llamo al método en el main pasándole los parámetros necesarios.

```
//apartado 13
escribirArchivoMedias("ArrayListArchivos/mediasCadaRegistro.txt", listaAlumnos, false);
```

El archivo se llama mediasCadaRegistro y estará dentro de la carpeta del proyecto en una carpeta llamada ArrayListArchivos.

14. Crea una sublista por cada grupo y ordénala por nombre descendente.

Se crean 5 ArrayList para poder guardar la información de cada uno de ellos y se recorre la lista para ir añadiendo a cada lista sus alumnos correspondientes

```
//apartado 14
ArrayList<Alumno> listaAlumnos1A = new ArrayList <>();
ArrayList<Alumno> listaAlumnos1B = new ArrayList <>();
ArrayList<Alumno> listaAlumnos1C = new ArrayList <>();
ArrayList<Alumno> listaAlumnos1D = new ArrayList <>();
ArrayList<Alumno> listaAlumnos1E = new ArrayList <>();
for(Alumno i: listaAlumnos) {
    if(i.getGrupo().equals("1º ESO A")) {
        listaAlumnos1A.add(i);
    }else if(i.getGrupo().equals("1º ESO B")) {
        listaAlumnos1B.add(i);
    }else if(i.getGrupo().equals("1º ESO C")) {
        listaAlumnos1C.add(i);
    }else if(i.getGrupo().equals("1º ESO D")) {
        listaAlumnos1D.add(i);
    }else if(i.getGrupo().equals("1º ESO E")) {
        listaAlumnos1E.add(i);
    }
}
OrdenarPorNombre ordenar = new OrdenarPorNombre();
Collections.sort(listaAlumnos1A, ordenar);
Collections.sort(listaAlumnos1B, ordenar);
Collections.sort(listaAlumnos1C, ordenar);
Collections.sort(listaAlumnos1D, ordenar);
Collections.sort(listaAlumnos1E, ordenar);

escribirNuevoArchivo("ArrayListArchivos/lista1ºA.txt", listaAlumnos1A, false);
escribirNuevoArchivo("ArrayListArchivos/lista1ºB.txt", listaAlumnos1B, false);
escribirNuevoArchivo("ArrayListArchivos/lista1ºC.txt", listaAlumnos1C, false);
escribirNuevoArchivo("ArrayListArchivos/lista1ºD.txt", listaAlumnos1D, false);
escribirNuevoArchivo("ArrayListArchivos/lista1ºE.txt", listaAlumnos1E, false);
```

Tras esto se ha creado una clase llamada OrdenarPorNombre para poder ordenar por el nombre y se llama al método sort() de la clase Collections indicando la lista que se quiere ordenar y el comparador a usar en este caso OrdenarPorNombre.

```
public class OrdenarPorNombre implements Comparator<Alumno>{

    @Override
    public int compare(Alumno o1, Alumno o2) {
        int devolucion;
        if(o1.getNombre().compareTo(o2.getNombre()) < 0) {
            devolucion = 1;
        }else if(o1.getNombre().compareTo(o2.getNombre()) > 0) {
            devolucion = -1;
        }else {
            devolucion = 0;
        }
        return devolucion;
    }

}
```

A continuación se llama al método crearNuevoArchivo para generar archivos con el contenido de las listas. Ordenados por nombre y cada grupo en su archivo y lista diferente.

Los archivos se llaman lista1ºA, lista1ºB, lista1ºC, lista1ºD, lista1ºE y estarán dentro de la carpeta del proyecto en una carpeta llamada ArrayListArchivos.

15. Calcula para cada grupo la nota media de cada materia.

Se crean 2 métodos ImprimirMedias y sacarMedias. En el primer método se imprime el texto deseado y se llama al segundo método para calcular las medias.

```
public static void imprimirMedias(ArrayList<Alumno> lista, String grupo) {
    System.out.println("\n\nEstas son las medias de cada asignatura de " + grupo + ":");
    System.out.printf("Matemáticas= %.2f", sacarMedia(lista, 1));
    System.out.printf("\nLengua= %.2f", sacarMedia(lista, 2));
    System.out.printf("\nFísica= %.2f", sacarMedia(lista, 3));
    System.out.printf("\nQuímica= %.2f", sacarMedia(lista, 4));
    System.out.printf("\nInglés= %.2f", sacarMedia(lista, 5));
    System.out.println("\n////////////////////////////////////");
}
```

El segundo método recibe como parámetro la lista y las asignaturas. Este método recorrerá la lista para la asignatura pasada por parámetro y hará el cálculo de la media.

Se suman todas las notas y se divide por el tamaño de la lista.

```
public static double sacarMedia(ArrayList<Alumno> lista, int asignatura) {  
    double devolucion = 0;  
    if(asignatura == 1) {  
        for(Alumno i: lista) {  
            devolucion += i.getMatematicas();  
        }  
        devolucion = devolucion/lista.size();  
    }else if(asignatura == 2) {  
        for(Alumno i: lista) {  
            devolucion += i.getLengua();  
        }  
        devolucion = devolucion/lista.size();  
    }else if(asignatura == 3) {  
        for(Alumno i: lista) {  
            devolucion += i.getFisica();  
        }  
        devolucion = devolucion/lista.size();  
    }else if(asignatura == 4) {  
        for(Alumno i: lista) {  
            devolucion += i.getQuimica();  
        }  
        devolucion = devolucion/lista.size();  
    }else if(asignatura == 5) {  
        for(Alumno i: lista) {  
            devolucion += i.getIngles();  
        }  
        devolucion = devolucion/lista.size();  
    }  
    return devolucion;  
}
```

Tras esto se llama al metodo imprimirMedias en el main para cada una de las listas y se muestra por pantalla

```
//apratado 15  
imprimirMedias(listaAlumnos1A, "1º ESO A");  
imprimirMedias(listaAlumnos1B, "1º ESO B");  
imprimirMedias(listaAlumnos1C, "1º ESO C");  
imprimirMedias(listaAlumnos1D, "1º ESO D");  
imprimirMedias(listaAlumnos1E, "1º ESO E");
```



```
Estas son las medias de cada asignatura de 1º ESO A:
Matemáticas= 4,83
Lengua= 4,33
Física= 5,08
Química= 4,50
Inglés= 5,33
////////////////////////////////////////////////////////////////

Estas son las medias de cada asignatura de 1º ESO B:
Matemáticas= 4,82
Lengua= 4,95
Física= 5,82
Química= 3,64
Inglés= 4,86
////////////////////////////////////////////////////////////////

Estas son las medias de cada asignatura de 1º ESO C:
Matemáticas= 3,89
Lengua= 4,17
Física= 4,67
Química= 2,78
Inglés= 3,83
////////////////////////////////////////////////////////////////

Estas son las medias de cada asignatura de 1º ESO D:
Matemáticas= 4,56
Lengua= 4,28
Física= 5,61
Química= 3,72
Inglés= 5,17
////////////////////////////////////////////////////////////////

Estas son las medias de cada asignatura de 1º ESO E:
Matemáticas= 5,80
Lengua= 5,57
Física= 6,40
Química= 4,07
Inglés= 5,97
////////////////////////////////////////////////////////////////
```

16. Para cada grupo muestra las diez mejores notas en matemáticas ordenadas por nota y nombre.

Se crea la clase OrdenarPorNotaNombre que implementa el interfaz `comprator` y se completa el método `compare` para ordenar primero por nota y luego por nombre.

```
public class OrdenarPorNotaNombre implements Comparator<Alumno>{

    @Override
    public int compare(Alumno o1, Alumno o2) {
        int devolucion;
        if(o1.getMatematicas() > o2.getMatematicas()) {
            devolucion = -1;
        }else if(o1.getMatematicas() < o2.getMatematicas()) {
            devolucion = 1;
        }else {
            if(o1.getNombre().compareTo(o2.getNombre()) > 0) {
                devolucion = 1;
            }else if(o1.getNombre().compareTo(o2.getNombre()) < 0) {
                devolucion = -1;
            }else {
                devolucion = 0;
            }
        }
        return devolucion;
    }
}
```

Se crea un método `mostrarMejoresNotas` que recibe como parámetro la lista, ordena esta lista con la clase comparador `OrdenarPorNotaNombre` y se añade al `String` las 10 primeras notas más altas con sus respectivos nombres

```
public static String mostrarMejoresNotas(ArrayList<Alumno> lista) {
    String devolucion = "";
    Collections.sort(lista, new OrdenarPorNotaNombre());
    for(int i = 0; i < 10; i++) {
        devolucion += "Nota: " + lista.get(i).getMatematicas() + " nombre: " + lista.get(i).getNombre() + "\n";
    }
    return devolucion;
}
```

Se llama al método `mostrarMejoresNotas` y se muestra por pantalla

```
//aparatado 16
System.out.println("Mostrando las 10 mejores notas del grupo 1º ESO A: \n" + mostrarMejoresNotas(listaAlumnos1A));
System.out.println("Mostrando las 10 mejores notas del grupo 1º ESO B: \n" + mostrarMejoresNotas(listaAlumnos1B));
System.out.println("Mostrando las 10 mejores notas del grupo 1º ESO C: \n" + mostrarMejoresNotas(listaAlumnos1C));
System.out.println("Mostrando las 10 mejores notas del grupo 1º ESO D: \n" + mostrarMejoresNotas(listaAlumnos1D));
System.out.println("Mostrando las 10 mejores notas del grupo 1º ESO e: \n" + mostrarMejoresNotas(listaAlumnos1E));
```

Genera esta salida por pantalla:

Mostrando las 10 mejores notas del grupo 1ª ESO A:

Nota: 9 nombre: Vilches Moreno, Lola Isabel

Nota: 6 nombre: González Pérez, Laura

Nota: 6 nombre: Pérez Pellisero, Braian

Nota: 6 nombre: Rueda Cruz, María

Nota: 5 nombre: González Pereau, Saray

Nota: 5 nombre: Ruiz Jiménez, Rafael

Nota: 4 nombre: Bintana , Esteban Martín

Nota: 4 nombre: García Benítez, Nazareth

Nota: 4 nombre: Peralta , Maximiliano

Nota: 4 nombre: Russell , Natalie

Mostrando las 10 mejores notas del grupo 1ª ESO B:

Nota: 10 nombre: Luque González, Christian

Nota: 9 nombre: Rueda Calzada, Laura

Nota: 8 nombre: Aguilera Delgado, Enrique

Nota: 7 nombre: Ramírez Farfán, Carmen

Nota: 6 nombre: Donaire Rodríguez, Miguel Ángel

Nota: 6 nombre: Donaire Vergara, Eva María

Nota: 6 nombre: Durán , Mariano

Nota: 6 nombre: Martínez Calderero, Aarón

Nota: 6 nombre: Sánchez Ortiz, Álvaro José

Nota: 5 nombre: Benítez Gaitán, Juan Cruz

Mostrando las 10 mejores notas del grupo 1ª ESO C:

Nota: 8 nombre: Mantero González, Juan

Nota: 8 nombre: Villar Risco, Omar

Nota: 5 nombre: Mora Castillo, Paola

Nota: 5 nombre: Roberto Jaimes, Kelly Ximena

Nota: 4 nombre: Castillo Alcázar, Alejandro

Nota: 4 nombre: Ramírez Pérez, David

Nota: 4 nombre: Recchia , María Fiorella

Nota: 4 nombre: Villatoro Postigo, Adrián

Nota: 3 nombre: Aranda Vázquez, José

Nota: 3 nombre: Cerdón López, Rocío

Mostrando las 10 mejores notas del grupo 1ª ESO D:

Nota: 9 nombre: García Bernal, María Esther

Nota: 8 nombre: Reina Romero, José Daniel

Nota: 7 nombre: Benítez Maya, Luciana

Nota: 7 nombre: Fontaneda Macías, Borja

Nota: 6 nombre: García López, Elena

Nota: 6 nombre: Álvarez Rodríguez, Dayana Vanessa

Nota: 4 nombre: García Sánchez, Carmen

Nota: 4 nombre: González Bernal, Cristóbal

Nota: 4 nombre: Hernández Jack, Irene

Nota: 4 nombre: Navarro García, María Azahara

Mostrando las 10 mejores notas del grupo 1ª ESO e:

Nota: 9 nombre: Benítez Berenjano, Laura

Nota: 9 nombre: Rico Pinazo, Pedro

Nota: 8 nombre: Delgado Rodríguez, Daniel

Nota: 8 nombre: Rodríguez Horcajo, Noelia

Nota: 8 nombre: Rodríguez Márquez, Ezequiel

Nota: 7 nombre: Abdeslam Blanco, Hillary

Nota: 7 nombre: Bernat Rodríguez, Sandra

Nota: 7 nombre: Cañas Arcos, Esther

Nota: 7 nombre: Romero González, Nazaret

Nota: 7 nombre: Rueda Blanco, Oscar