Ian Montes
1ºA DAW N

Programación

Actividad 6.1

Ejercicio 1 – Main + ejecución

```java
import java.util.ArrayList;
import java.util.Iterator;

public class Main {
    public static void main(String[] args) {

        ArrayList<Object> miArray = new ArrayList();

        int miInt = 20;
        float miFloat = 10f;
        double miDouble = 15.50;
        char miChar = 'b';

        miArray.add(miInt);
        miArray.add(miFloat);
        miArray.add(miDouble);
        miArray.add(miChar);

        Iterator<Object> miIterator = miArray.iterator();
        while(miIterator.hasNext()){
            Object miObjeto = miIterator.next();
            System.out.println(miObjeto.toString()+ " es de tipo " + miObjeto.getClass());
        }
    }
}

```

```
20 es de tipo class java.lang.Integer
10.0 es de tipo class java.lang.Float
15.5 es de tipo class java.lang.Double
b es de tipo class java.lang.Character

Process finished with exit code 0
```

Ejercicio 2 - Main + ejecución

```java
public class Main {
    public static void main(String[] args) {

        Calculos miArray = new Calculos();

        miArray.GenerarAleatorios();

        System.out.println(miArray.toString());

        System.out.println("El mayor es " + miArray.Maximo());
        System.out.println("El menor es " + miArray.Minimo());
        System.out.println("La suma de todos los elementos es " + miArray.Sumar());
        System.out.println("La media de los elementos es " + miArray.Media());

    }
}
```

```
Calculos{[6, 35, 10, 6, 4, 58, 13, 76, 26, 92, 22, 67, 71, 6, 100, 55]}
El mayor es 100
El menor es 4
La suma de todos los elementos es 647
La media de los elementos es 40

Process finished with exit code 0
```

Ejercicio 2 – Clase Cálculos

```java
import java.util.ArrayList;

public class Calculos {
    ArrayList<Integer> miArrayList;

    public Calculos(){
        miArrayList = new ArrayList<>();
    }

    public void GenerarAleatorios(){

        for (int i = 0; i <= 10 +(Math.toIntExact(Math.round(Math.random() * 10))); i++) {
            Integer miInt = Math.toIntExact(Math.round(Math.random() * 100));
            miArrayList.add(miInt);
        }
    }

    public int Sumar(){
        int suma = 0;
        for (Integer num:miArrayList) {
            suma += num;
        }
        return suma;
    }

    public int Media(){
        return Sumar() / miArrayList.size();
```



```java
    }

    public int Maximo(){

        int max = miArrayList.get(0);
        for (Integer num:miArrayList) {

            if (num > max){
                max = num;
            }
        }
        return max;
    }

    public int Minimo(){

        int min = miArrayList.get(0);
        for (Integer num:miArrayList) {

            if (num < min){
                min = num;
            }
        }
        return min;
    }

    @Override
```



```java
    public String toString() {
        return "Calculos{" + miArrayList +
                '}';
    }
}
```

Ejercicio 3 – Main + ejecución

```java
public class Main {
    public static void main(String[] args) {

        PilaPalabras miPila = new PilaPalabras();

        miPila.apilarPalabra("palabra 1");
        miPila.apilarPalabra("palabra 2");
        miPila.apilarPalabra("palabra 3");
        miPila.apilarPalabra("palabra 4");
        System.out.println(miPila.toString());

        System.out.println("Solo mostrar palabra: " + miPila.obtenerPalabra());
        System.out.println("Quitar palabra: " + miPila.desapilarPalabra());
        System.out.println("Quitar palabra: " + miPila.desapilarPalabra());

        miPila.apilarPalabra("palabra 5");
        System.out.println(miPila.toString());

    }
}


```

```
PilaPalabras{[palabra 4, palabra 3, palabra 2, palabra 1]}
Solo mostrar palabra: palabra 4
Quitar palabra: palabra 4
Quitar palabra: palabra 3
PilaPalabras{[palabra 5, palabra 2, palabra 1]}

Process finished with exit code 0
```

Ejercicio 3 – Clase PilaPalabra

```java
import java.util.LinkedList;

public class PilaPalabras {
    LinkedList <String> miLinkedList;

    public PilaPalabras(){

        miLinkedList = new LinkedList<>();

    }

    public void apilarPalabra(String palabra){

        miLinkedList.add(0, palabra);

    }

    public String desapilarPalabra(){
        return miLinkedList.pop();
    }

    public String obtenerPalabra(){
        return miLinkedList.get(0);
    }

    @Override
    public String toString() {
```



```java
        return "PilaPalabras{" + miLinkedList.toString() +
                '}';
    }
}
```

Ejercicio 4 – Main + ejecución

```java
public class Main {
    public static void main(String[] args) {

        Tester prueba = new Tester();

        prueba.imprimirAscendente();

        System.out.println("----------");

        prueba.imprimirDescendente();

    }
}
```

```
Ficha{1|3}
Ficha{3|3}
Ficha{4|2}
Ficha{4|4}
Ficha{5|5}
Ficha{5|6}
----------
Ficha{5|6}
Ficha{5|5}
Ficha{4|4}
Ficha{4|2}
Ficha{3|3}
Ficha{1|3}

Process finished with exit code 0
```

Ejercicio 4 – Clase Ficha

```java
import java.util.TreeSet;

public class Ficha implements Comparable{
    private int valor1;
    private int valor2;

    public Ficha(int valor1, int valor2){
        this.valor1 = valor1;
        this.valor2 = valor2;
    }

    public int getValor1() {
        return valor1;
    }

    public void setValor1(int valor1) {
        this.valor1 = valor1;
    }

    public int getValor2() {
        return valor2;
    }

    public void setValor2(int valor2) {
        this.valor2 = valor2;
    }

    public int PesoPonderado(int valor1, int valor2){
        return (valor1*6) + valor2;
    }

    @Override
    public String toString() {
        return "Ficha{" + valor1 + "|" + valor2 +
                '}';
    }

    @Override
    public int compareTo(Object o) {
        Ficha ficha = (Ficha) o;
        Integer pesoPonderado1;
        Integer pesoPonderado2;

        if (valor1 > valor2){
            pesoPonderado1 = PesoPonderado(this.valor1, this.valor2);
        } else{
            pesoPonderado1 = PesoPonderado(this.valor2, this.valor1);
        }

        if (ficha.getValor1() > ficha.getValor2()){
            pesoPonderado2 = PesoPonderado(ficha.getValor1(), ficha.getValor2());
        } else{
            pesoPonderado2 = PesoPonderado(ficha.getValor2(), ficha.getValor1());
        }

        return pesoPonderado1.compareTo(pesoPonderado2);
    }
}
```

Ejercicio 4 – Clase Tester

```java
import java.util.Iterator;
import java.util.TreeSet;

public class Tester {

    private TreeSet<Ficha> miTreeSet;
    public Tester(){
        miTreeSet = new TreeSet<>();
        while (miTreeSet.size() < 6) {
            miTreeSet.add(new Ficha(1 +(Math.toIntExact(Math.round(Math.random() * 5))), (1 +(Math.toIntExact(Math.round(Math.random() * 5))) )));
        }
    }

    @Override
    public String toString() {
        return "Tester{" + miTreeSet.toString() +
                '}';
    }

    public void imprimirAscendente(){

        Iterator<Ficha> iterador = miTreeSet.iterator();

        while (iterador.hasNext()){
            System.out.println(iterador.next().toString());
        }

    }

    public void imprimirDescendente(){

        Iterator<Ficha> iterador = miTreeSet.descendingIterator();

        while (iterador.hasNext()){
            System.out.println(iterador.next().toString());
        }

    }
}
```

Ejercicio 5 – Main

```java
import java.util.Map;
import java.util.TreeMap;
import java.util.TreeSet;

public class Main {
    public static void main(String[] args) {

        Tester mapPersonas = new Tester();

        mapPersonas.aniadir("38856774E", new Persona("Ian", "Montes"));
        mapPersonas.aniadir("63467894V", new Persona("Naomi", "Perez"));
        mapPersonas.aniadir("58473612G", new Persona("Felipe", "Cañas"));

        System.out.println("\n---------- Mostrar por pantalla el HashMap ----------");
        mapPersonas.imprimirEntries();

        System.out.println("\n---------- Tratando de obtener una entrada que no existe ----------");
        System.out.println(mapPersonas.getColeccionPersonas().get("45763448F"));

        System.out.println("\n---------- Añadiendo una nueva entrada para reemplazar la primera ----------");
        mapPersonas.aniadir("38856774E", new Persona("Juan", "Reyes"));
        mapPersonas.imprimirEntries();

        TreeMap<String, Persona> miTreeMap = new TreeMap<>();
        miTreeMap.putAll(mapPersonas.getColeccionPersonas());


        System.out.println("\n---------- Imprimir el TreeMap (Ordenado por la Key) ----------");
        for (String key: miTreeMap.keySet()) {
            System.out.println(key + "=" + miTreeMap.get(key));
        }
    }
}
```

Ejercicio 5 - Ejecución

```
---------- Mostrar por pantalla el HashMap ----------
38856774E={nombre='Ian', apellidos='Montes'}
63467894V={nombre='Naomi', apellidos='Perez'}
58473612G={nombre='Felipe', apellidos='Cañas'}

---------- Tratando de obtener una entrada que no existe ----------
null

---------- Añadiendo una nueva entrada para reemplazar la primera ----------
38856774E={nombre='Juan', apellidos='Reyes'}
63467894V={nombre='Naomi', apellidos='Perez'}
58473612G={nombre='Felipe', apellidos='Cañas'}

---------- Imprimir el TreeMap (Ordenado por la Key) ----------
38856774E={nombre='Juan', apellidos='Reyes'}
58473612G={nombre='Felipe', apellidos='Cañas'}
63467894V={nombre='Naomi', apellidos='Perez'}

Process finished with exit code 0
```

Ejercicio 5 - Clase Tester

```java
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;

public class Tester {

    private HashMap<String, Persona> coleccionPersonas;

    public Tester(){
        coleccionPersonas = new HashMap<>();
    }

    public HashMap<String, Persona> getColeccionPersonas() {
        return coleccionPersonas;
    }

    public void setColeccionPersonas(HashMap<String, Persona> coleccionPersonas) {
        this.coleccionPersonas = coleccionPersonas;
    }

    public void aniadir(String nif, Persona persona){
        coleccionPersonas.put(nif, persona);
    }

    public void imprimirKeys(){
        for (String key: coleccionPersonas.keySet()) {
            System.out.println(key);
```



```java
        }
    }

    public void imprimirValues(){
        for (Persona value:coleccionPersonas.values()) {
            System.out.println(value.toString());
        }
    }

    public void imprimirEntries(){
        for (HashMap.Entry entry:coleccionPersonas.entrySet()) {
            System.out.println(entry.getKey()+"="+entry.getValue().toString());
        }
    }


}
```

Ejercicio 5 - Clase Persona

```java
public class Persona {
    private String nombre;
    private String apellidos;

    public Persona(String nombre, String apellidos){
        this.nombre = nombre;
        this.apellidos = apellidos;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    @Override
    public String toString() {
        return "{nombre='" + nombre + '\'' +
                ", apellidos='" + apellidos + '\'' +
                '}';
    }
}
```

Ejercicio 6 - Main

```java
import java.util.HashSet;
import java.util.Iterator;
import java.util.Scanner;
import java.util.TreeSet;

public class Main {
    public static void main(String[] args) {
        HashSet<Coche> misCoches = new HashSet<>();

        misCoches.add(new Coche("Audi", "A1", "5423HGL", 90754, 140));
        misCoches.add(new Coche("Audi", "A4", "6452TDF", 150675, 150));
        misCoches.add(new Coche("Volkswagen", "Polo", "0745HLF", 99567, 90));
        misCoches.add(new Coche("Opel", "Corsa", "3456CD", 456573, 80));

        Scanner scanner = new Scanner(System.in);
        boolean condicion = true;

        while (condicion){
            System.out.println("Introduzca la marca");
            String marca = scanner.nextLine();
            System.out.println("Introduzca el modelo");
            String modelo = scanner.nextLine();
            System.out.println("Introduzca la matrícula");
            String matricula = scanner.nextLine();
            int kilometraje;
            while (true){
                try {
```



```java
                    System.out.println("Introduzca el kilometraje");
                    kilometraje = scanner.nextInt();
                    scanner.nextLine();
                    break;
                } catch (Exception e) {
                    System.out.println("Asegúrese de introducir sólo caracteres numéricos");
                    scanner.nextLine();
                }
            }
            int potencia;
            while (true){
                try {
                    System.out.println("Introduzca la potencia");
                     potencia = scanner.nextInt();
                    scanner.nextLine();
                    break;
                } catch (Exception e) {
                    System.out.println("Asegúrese de introducir sólo caracteres numéricos");
                    scanner.nextLine();
                }
            }

            misCoches.add(new Coche(marca, modelo, matricula, kilometraje, potencia));

            boolean repetir = true;

            while (repetir){
```

Ejercicio 6 - Main

```
55              System.out.println("¿Quiere añadir otra entrada de Coche? Responda Y/N");
56              String respuesta = scanner.nextLine();
57              if (respuesta.equals("Y")){
58                  repetir = false;
59              } else if (respuesta.equals("N")) {
60                  repetir = false;
61                  condicion = false;
62              }else {
63                  System.out.println("Respuesta fuera de rango, responda con Y/N");
64
65              }
66          }
67
68      }
69
70      Iterator<Coche> iterador1 = misCoches.iterator();
71      System.out.println("\nEl iterador imprime todos los coches de forma aleatoria");
72
73      while (iterador1.hasNext()){
74          Coche coche = (Coche) iterador1.next();
75              System.out.println(coche.toString());
76      }
77
78      Iterator<Coche> iterador2 = misCoches.iterator();
79      System.out.println("\nEl iterador imprime solo los coches de marca Audi");
80
81      while (iterador2.hasNext()){
```



```
82          Coche coche = (Coche) iterador2.next();
83          if (coche.getMarca().equals("Audi")){
84              System.out.println(coche.toString());
85          }
86      }
87
88      Iterator<Coche> iterador3 = misCoches.iterator();
89
90      System.out.println("\nEl iterador imprime solo los coches de menos de 100.000km");
91
92      while (iterador3.hasNext()){
93          Coche coche = (Coche) iterador3.next();
94          if (coche.getKilometraje() < 100000){
95              System.out.println(coche.toString());
96          }
97      }
98
99
100     TreeSet<Coche> misCochesOrdenados = new TreeSet<>(misCoches);
101
102     Iterator<Coche> iterador4 = misCochesOrdenados.iterator();
103     System.out.println("\nEl iterador imprime el último elemento");
104
105     while (iterador4.hasNext()){
106         Coche coche = (Coche) iterador4.next();
107         if (!iterador4.hasNext()){
108             System.out.println(coche.toString());
```

Ejercicio 6 - Main

```
109             }
110         }
111
112         Iterator<Coche> iterador5 = misCochesOrdenados.iterator();
113         System.out.println("\n El iterador imprime los coches de forma ordenada por , de menor a mayor");
114
115         while (iterador5.hasNext()){
116             Coche coche = (Coche) iterador5.next();
117             System.out.println(coche.toString());
118         }
119
120
121     }
122 }
```

El ejercicio pedía añadir los coches por consola, pero para poder agilizar la escritura del código al principio de éste (línea 10) añado 4 coches hardcodeados.

En la ejecución sin embargo se añaden algunos más por consola para mostrar que ésto funciona.

No he eliminado las líneas éstas para evitar tener que enviar 10 capturas de pantalla de la consola.

Ejercicio 6 - Ejecución

```
Introduzca la marca
Opel
Introduzca el modelo
Adam
Introduzca la matrícula
4534GLF
Introduzca el kilometraje
cienmil trescientos cuarenta y seis
Asegúrese de introducir sólo caracteres numéricos
Introduzca el kilometraje
100346
Introduzca la potencia
90
¿Quiere añadir otra entrada de Coche? Responda Y/N
Y
Introduzca la marca
Seat
Introduzca el modelo
Ibiza
Introduzca la matrícula
4567DFG
Introduzca el kilometraje
756645
Introduzca la potencia
100
¿Quiere añadir otra entrada de Coche? Responda Y/N
No
Respuesta fuera de rango, responda con Y/N
¿Quiere añadir otra entrada de Coche? Responda Y/N
N

●El iterador imprime todos los coches de forma aleatoria
Coche{marca='Volkswagen', modelo='Polo', kilometraje=99567, matricula='0745HLF', potencia=90}
Coche{marca='Opel', modelo='Adam', kilometraje=100346, matricula='4534GLF', potencia=90}
Coche{marca='Audi', modelo='A1', kilometraje=90754, matricula='5423HGL', potencia=140}
Coche{marca='Seat', modelo='Ibiza', kilometraje=756645, matricula='4567DFG', potencia=100}
Coche{marca='Opel', modelo='Corsa', kilometraje=456573, matricula='3456CD', potencia=80}
Coche{marca='Audi', modelo='A4', kilometraje=150675, matricula='6452TDF', potencia=150}

●El iterador imprime solo los coches de marca Audi
Coche{marca='Audi', modelo='A1', kilometraje=90754, matricula='5423HGL', potencia=140}
Coche{marca='Audi', modelo='A4', kilometraje=150675, matricula='6452TDF', potencia=150}

●El iterador imprime solo los coches de menos de 100.000km
Coche{marca='Volkswagen', modelo='Polo', kilometraje=99567, matricula='0745HLF', potencia=90}
Coche{marca='Audi', modelo='A1', kilometraje=90754, matricula='5423HGL', potencia=140}

●El iterador imprime el último elemento
Coche{marca='Seat', modelo='Ibiza', kilometraje=756645, matricula='4567DFG', potencia=100}

●El iterador imprime los coches de forma ordenada por , de menor a mayor
Coche{marca='Audi', modelo='A1', kilometraje=90754, matricula='5423HGL', potencia=140}
Coche{marca='Volkswagen', modelo='Polo', kilometraje=99567, matricula='0745HLF', potencia=90}
Coche{marca='Opel', modelo='Adam', kilometraje=100346, matricula='4534GLF', potencia=90}
Coche{marca='Audi', modelo='A4', kilometraje=150675, matricula='6452TDF', potencia=150}
Coche{marca='Opel', modelo='Corsa', kilometraje=456573, matricula='3456CD', potencia=80}
Coche{marca='Seat', modelo='Ibiza', kilometraje=756645, matricula='4567DFG', potencia=100}

Process finished with exit code 0
```

Ejercicio 6 - Clase Coche

```java
import java.util.HashSet;
import java.util.TreeSet;

public class Coche implements Comparable<Coche> {
    private String marca;
    private String modelo;
    private Integer kilometraje;
    private String matricula;
    private int potencia;

    public Coche(String marca, String modelo, String matricula, Integer kilometraje, int potencia){
        this.marca = marca;
        this.modelo = modelo;
        this.kilometraje = kilometraje;
        this.matricula = matricula;
        this.potencia = potencia;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModelo() {
```



```java
        return modelo;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }

    public double getKilometraje() {
        return kilometraje;
    }

    public void setKilometraje(Integer kilometraje) {
        this.kilometraje = kilometraje;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public int getPotencia() {
        return potencia;
    }

```

Ejercicio 6 - Clase Coche

```
55      public void setPotencia(int potencia) {
56          this.potencia = potencia;
57      }
58
59      @Override
60      public String toString() {
61          return "Coche{" +
62                  "marca='" + marca + '\'' +
63                  ", modelo='" + modelo + '\'' +
64                  ", kilometraje=" + kilometraje +
65                  ", matricula='" + matricula + '\'' +
66                  ", potencia=" + potencia +
67                  '}';
68      }
69
70      @Override
71      public int compareTo(Coche coche) {
72          return this.kilometraje.compareTo(coche.kilometraje);
73      }
74  }
75
```

Ejercicio 7 – Main + ejecución

```java
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> miArray = new ArrayList<>();

        //Comprobando que si no añades ningún String salta un mensaje indicándolo
        calcularString.cadenaMasLarga(miArray);

        //Lectura de Strings por pantalla
        calcularString.leerArray(miArray);

        //Calcular cadena más larga
        calcularString.cadenaMasLarga(miArray);

    }
}
```



```
No ha añadido ningún String todavia al ArrayList
Escribe el String que quieres añadir, has añadido 0 Strings de momento.
Hola
Escribe el String que quieres añadir, has añadido 1 Strings de momento.
Qué tal
Escribe el String que quieres añadir, has añadido 2 Strings de momento.
¿Qué String saldrá...?
Escribe el String que quieres añadir, has añadido 3 Strings de momento.
Adiós
Escribe el String que quieres añadir, has añadido 4 Strings de momento.
FIN
El String más largo es "¿Qué String saldrá...?" con 22 caracteres

Process finished with exit code 0
```

Ejercicio 7 – Clase CalcularString

```java
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

public class calcularString {

    public static void leerArray(ArrayList miArray){
        boolean condicion = true;

        Scanner scanner = new Scanner(System.in);

        while (condicion){
            System.out.println("Escribe el String que quieres añadir, has añadido " + miArray.size() + " Strings de momento.");
            String respuesta = scanner.nextLine();
            if (respuesta.equals("FIN")){
                condicion = false;
            }else{
                miArray.add(respuesta);
            }
        }

    }

    public static void cadenaMasLarga(ArrayList miArray){

        if (miArray.size() != 0){
```

Ejercicio 7 – Clase CalcularString

```
27              Iterator<String> miIterador = miArray.iterator();
28              String cadena = miIterador.next();
29              while (miIterador.hasNext()){
30                  if (miIterador.next().length() > cadena.length()){
31                      cadena = miIterador.next();
32                  }
33              }
34              System.out.println("El String más largo es " + "\"" + cadena + "\" con " + cadena.length() + " caracteres
    ");
35          }else {
36              System.out.println("No ha añadido ningún String todavia al ArrayList");
37          }
38      }
39  }
40
```