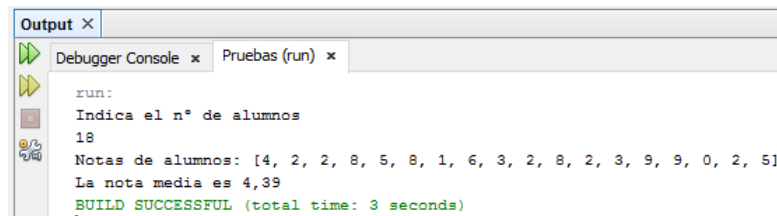


Para almacenar datos numéricos en cualquier Collection o Map, debemos usar las clases que representan a los tipos de datos primitivos: Integer, Double, Float, Short y Long

```
//Ejemplo de variable objeto que almacena el número 8
// esta variable la podremos usar como si fuera un int
Integer numero = new Integer(8);
```

ARRAYLIST

1. **SOLUCIONADO.** Implementa programa que genera y almacena las N notas de mis alumnos y saca la nota media.



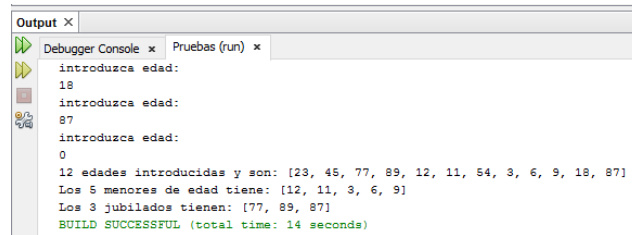
```
Scanner sc = new Scanner(System.in);
Random aleatorio = new Random();
List<Integer> notas = new ArrayList<>();
int nAlumnos = 0, nota, suma = 0;
double media;

System.out.println("Indica el n° de alumnos");
nAlumnos = sc.nextInt();

for (int i = 0; i < nAlumnos; i++) {
    nota = aleatorio.nextInt(11);
    notas.add(nota);
    suma += nota;
}

media = (double) suma / notas.size();
System.out.println("Notas de alumnos: " + notas);
System.out.printf("La nota media es %2.2f \n", media);
```

- 2. Implementa programa que recoge edades hasta que el usuario teclee 0 (no sabemos cuantas introducirá). Debes almacenarlas todas, las de menores de edad y las de jubilados en ArrayList diferentes, e indicar cuantas hay de cada tipo.**



```
Output X
Debugger Console x Pruebas (run) x
introduzca edad:
18
introduzca edad:
87
introduzca edad:
0
12 edades introducidas y son: [23, 45, 77, 89, 12, 11, 54, 3, 6, 9, 18, 87]
Los 5 menores de edad tiene: [12, 11, 3, 6, 9]
Los 3 jubilados tienen: [77, 89, 87]
BUILD SUCCESSFUL (total time: 14 seconds)
```

```
List<Integer> edades = new ArrayList<>();
List<Integer> menores = new ArrayList<>();
List<Integer> jubilados = new ArrayList<>();

int edad = 0;
Scanner sc = new Scanner(System.in);

do {
    System.out.println("introduzca edad:");
    edad = sc.nextInt();

    if (edad != 0) {
        edades.add(edad);
    }
    if (edad > 65) {
        jubilados.add(edad);
    }

    if (edad < 18 && edad != 0) {
        menores.add(edad);
    }
} while (edad != 0);

System.out.println(edades.size() + " edades introducidas y son: " + edades);
System.out.println("Los " + menores.size() + " menores de edad tiene: " + menores);
System.out.println("Los " + jubilados.size() + " jubilados tienen: " + jubilados);
```

HASHSET

3. SOLUCIONADO. Almacena en un HashSet, llamado "marcas_moviles", nombres de 5 marcas de móviles.

- Muestra por pantalla los datos introducidos
- Muestra cuantos elementos hay en marcas_moviles
- Indica si está "Samsung"

```
Set<String> marcas_moviles = new HashSet<>();

marcas_moviles.add("Samsung");
marcas_moviles.add("Jiayu");
marcas_moviles.add("Iphone");
marcas_moviles.add("Motorola");
marcas_moviles.add("Alcatel");

System.out.println("moviles: " + marcas_moviles);
System.out.println("Nº de marcas:" + marcas_moviles.size());

if (marcas_moviles.contains("Samsung")) {
    System.out.println("Si existe Samsung");
}
```

4. Almacena en un HashSet, llamado "coches", nombres de 5 marcas de coches .

- Muestra por pantalla los datos introducidos
- ¿Está "Seat"? si está bórralo e introdúcelo como "SEAT"
- Crea otro HashSet con 5 marcas de coches japoneses, y añádelas a coches (usa un único método)
- Muestra por pantalla los datos
- Muestra cuantos elementos hay en coches

```
Set<String> coches = new HashSet<>();
Set<String> cochesJaponeses = new HashSet<>();

coches.add("Renault");
coches.add("Seat");
coches.add("Ford");
coches.add("Volkswagen");

cochesJaponeses.add("Toyota");
cochesJaponeses.add("Nissan");
cochesJaponeses.add("Honda");

System.out.println(coches);

if (coches.contains("Seat")) {
    coches.remove("Seat");
    coches.add("SEAT");
}

coches.addAll(cochesJaponeses);
System.out.println("coches japoneses: " + cochesJaponeses);
System.out.println("Todos los coches: " + coches);

System.out.println("Nº de marcas:" + coches.size());
```

ARRAYDEQUE

5. SOLUCIONADO. Para gestionar la cola de espera para matricularse en el ciclo de DAW crea un **ArrayDeque** para almacenar las personas según orden de preinscripción.

- Rellena con 3 nombres personas
- Pide un nombre de alumno que quiere entrar en la cola e introdúcelo al final
- Muestras cuantos y quienes estan en la cola
- Recupera el ultimo
- Se admite al primero, muéstralo y elimínalo de la cola
- Muestras cuantos y quienes estan en la cola

```
Deque<String> colaDaw = new ArrayDeque<>();

colaDaw.add("Juan");
colaDaw.add("Berta");
colaDaw.add("Salma");

//añade elemento al final
colaDaw.addLast("Luis");

System.out.println(colaDaw.size() + " personas en espera " + colaDaw);

//recupera el ultimo elemento
System.out.println(" ultimo: " + colaDaw.getLast());

// recupera y borra el primero
System.out.println(" primero admitido: " + colaDaw.removeFirst());

System.out.println(colaDaw.size() + " personas en espera " + colaDaw);
```

6. Implementa un ArrayDeque que sirva para organizar los pedidos a nuestra empresa. Crea una menú con las siguientes opciones:

- Dar de alta pedido e introducirlo en la cola
- Conocer cuántos pedidos tenemos
- Visualizar el primer pedido
- Servir un pedido, que desaparecerá de la cola
- Salir del programa

```
Deque<String> pedidos = new ArrayDeque<>();
Scanner sc = new Scanner(System.in);
int opcion;
String pedido, servido;
String menu = "Elige opcion: \n 1.Alta  \n2.Cuantos \n3.Primeros \n4.Servir pedido \n5.Salir ";

do {
    System.out.println(menu);
    opcion = sc.nextInt();
    sc.nextLine();

    switch (opcion) {
        case 1:
            System.out.println("Indique el pedido: ");
            pedido = sc.nextLine();
            pedidos.add(pedido);
            break;
```

```
        case 2:
            System.out.println("Hay " + pedidos.size() + " pedidos registrados");
            break;
        case 3:
            System.out.println("Primer pedido en cola: " + pedidos.getFirst());
            break;
        case 4:
            servido = pedidos.remove();
            if (servido != null) {
                System.out.println(" Pedido: " + servido + " SERVIDO.");
            }
            break;
        default:
            System.out.println("Gracias por usar nuestra aplicacion");
    }
}

} while (opcion < 5);
```

HASHMAP

7. SOLUCIONADO. Almacena en un HashMap los códigos y descripciones de los 5 primeros pokemon.

- Muestra por pantalla los datos almacenados
- Pide un código y muestra el nombre del pokemon
- Lista solo los nombres de todos los pokemon

```
Map<String, String> pokemon = new HashMap<>();
Scanner sc = new Scanner(System.in);
String codigo = "", NomPokemon;

pokemon.put("001", "Bulbasaur");
pokemon.put("002", "Ivysaur");
pokemon.put("003", "Venusaur");
pokemon.put("004", "Charmander");
pokemon.put("005", "Charmeleon");

System.out.println(pokemon);

System.out.print("codigo pokemon a buscar:");
String codPokemon = sc.next();

System.out.println("Nombre del pokemon " + codPokemon + ":" + pokemon.get(codPokemon));

//Lista todos los pokemon
Iterator<String> claves = pokemon.keySet().iterator();
while (claves.hasNext()) {
    codigo = claves.next();
    NomPokemon = pokemon.get(codigo);
    System.out.println(NomPokemon);
}
```

8. Almacena en un HashMap los códigos postales de las provincias de la Comunidad Valenciana y Aragón.

- d. Muestra por pantalla los datos introducidos
- e. Pide un código postal y muestra la provincia asociada si existe sino avisa al usuario
- f. Elimina la provincia con código 8
- g. Elimina la provincia Castellón (no conoces el código). Haz un recorrido eficiente.
- h. Muestra por pantalla los datos

```
Map<Integer, String> provincias = new HashMap<>();
Scanner sc = new Scanner(System.in);
String pro = "";

provincias.put(46, "Valencia");
provincias.put(03, "Alicante");
provincias.put(12, "Castellon");
provincias.put(50, "Zaragoza");
provincias.put(44, "Teruel");
provincias.put(22, "Huesca");

System.out.println(provincias);

System.out.print("CP a buscar:");
int cp = sc.nextInt();

if (provincias.containsKey(cp)) {
    System.out.println("Provincia: " + provincias.get(cp));
} else {
    System.out.println("No existe");
}

//borra provincia sabiendo cp
provincias.remove(22);
System.out.println("CP 22 eliminado");

//borra Teruel sabiendo nombre de provincia
Iterator<Integer> claves = provincias.keySet().iterator();
while (claves.hasNext() && !pro.equalsIgnoreCase("Teruel")) {
    Integer codigo = claves.next();
    pro = provincias.get(codigo);
    if (pro.equals("Teruel")) {
        provincias.remove(codigo);
        System.out.println("Teruel eliminado");
    }
}
System.out.println(provincias);
```