Arrays

Objetivo: guardar N datos del mismo tipo
Se declaran usando corchetes: [ ]
Característica principal: tiene tamaño fijo
Ej: int [] listaNumeros = {1,2,3,4}
Los elementos están indexados, se accede por posición empezando por 0.
El número de elementos con length : listaNumeros.length.

Recorrer un array
for(int i = 0; i<listaNumeros.length;i++){
sout(listaNumeros[i]);
}
for (int elem:listaNumeros){
sout(elem);
}
break; se usa para romper el for

si queremos un array de varios datos es int[ ][ ] conjuntoDoble = { {111,0} {222,0} {333,0} }

Arrays list

Es el mismo concepto de los arrays, sirve para guardar elementos de un tipo.
Están indexados, es decir, se accede por índice.
Los tipos de datos NO pueden ser primitivos es decir en vez de int seria Integer
No tienen un tamaño limitado
Para indicar el tipo de elemento que se almacena en un arrayList se usan las clases Wrapper.

ej : ArrayList<Integer> en vez de ArrayList<int>

Métodos de Arrays list

Declaración : Array <Integer> listaNumeros = new ArrayList<Integer>();

Añadir un elemento : listaNumeros.add(5);

Obtener un elemento de una posición : listaNumeros.get(0)

Modificar un elemento : listaNumeros.set(0,3) ; 0->posición 3->nuevo valor

Borrar un elemento: listaNumeros.set(0);

Tamaño del array: listaNumeros.size();

LinkedList

Son como los ArrayList.

La diferencia es en el funcionamiento interno.

Se usan cuando hay mucho borrado o inserción de elementos.

Ej: LinkedList<Integer> listaNumeros = new LinkedList<Integer>();

Method	Description
addFirst()	Adds an item to the beginning of the list.
addLast()	Add an item to the end of the list
removeFirst()	Remove an item from the beginning of the list.
removeLast()	Remove an item from the end of the list
getFirst()	Get the item at the beginning of the list
getLast()	Get the item at the end of the list

HashMap

La estructura es la de un diccionario

Cada elemento tiene una clave que siempre es un String y cada valor es un valor

HashMap<String,Integer>hijosPaga = new HashMap<String, Integer>();

hijosPaga.put(“Ana”,10)

Insertar un elemento: hijosPaga.put(“Ana”,10);

Obtener el valor con la clave: hijosPaga.get(“Ana”);

Borrar un elemento: hijosPaga.remove(“Ana”);

Tamaño del hashMap: hijosPaga.size();

Elemento= Clave + valor

Recorrer un hashMap

```
for (String clave : hijosPaga.keySet()){
sout(clave)     //imprime solo las claves
}

for (Integer valor hijosPaga.values()){
sout(valor)          //imprime solo los valores
}
```

HashSet

Es similar al concepto de conjunto matemático.

La característica fundamental es que lo elementos no se pueden repetir.

Declaración: HashSet<Integer> cjtNumeros = new HashSet<String>();

Añadir un elemento: cjtNumeros.add(“5”);

Comprobar si un elemento existe: cjtNumeros.contains(“cinco”); //true si existe

Tamaño del hashMap: cjtNumeros.size();

Borrar un elemento: cjtNumeros.remove(“cinco”);