Arrays Unidimensionales

UNIVERSIDAD DE LIMA

Arrays (Arreglos) Unidimensionales

Un array unidimensional es un **tipo de datos estructurado** formado de una colección finita, homogénea y ordenada de elementos.

Finita : tiene un tamaño fijo
Homogénea : todos los elementos son del mismo tipo
Ordenada : se puede determinar el primer elemento, el segundo, … etc.

Formato de Declaración:

tipoDato[] nombreVariable

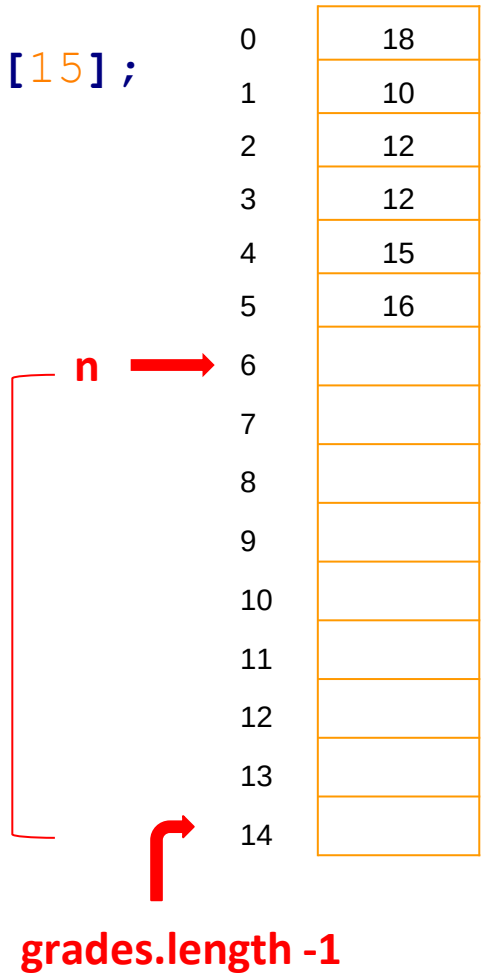
```
int[] grades;
//Para inicializar el array
grades = new int[20];

//Es posible utilizar una sola linea
float[] weights = new float[30];
```

Arrays Unidimensionales

Ejemplo:

```
int grades[15] = new grades[15];
n = 0;
grades[0] = 18;   n++;
grades[1] = 10;   n++;
grades[2] = 12;   n++;
grades[3] = 12;   n++;
grades[4] = 15;   n++;
grades[5] = 16;   n++;
```



- El acceso a cualquier elemento del array es directo:

  NombreDelArray [índice]

- La operación de recorrido consiste en acceder y **procesar** cada elemento de la lista (representada por el array) exactamente una vez.

```
for(int i = 0; i < n; i++){
    //Procesar nota i
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

OPERACIONES BASICAS

- Inserción
- Búsqueda
- Eliminación

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

PARA ARRAYS NO ORDENADOS

- Inserción
- Búsqueda Secuencial
- Eliminación

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Inserción

Especificaciones

Objetivo:	Insertar el elemento ítem en la lista (array A).
Entrada:	array A, n, item
Precondición:	Ninguna
Salida:	array A, n
Postcondición:	Lista A con un nuevo elemento, siempre que exista espacio.

```
public static int insert(int[] A, int n, int item){
    if(n < A.length){
        A[n] = item;
        n++;
    }
    return n;
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Búsqueda Secuencial (Lineal)

Especificaciones

Objetivo:	Buscar posición index del elemento X a buscar.
Entrada:	array A, n, X
Precondición:	array A no vacío (n>0)
Salida:	index
Postcondición:	index > 0 contiene la posición del elemento X; index = -1 caso contrario.

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Búsqueda Secuencial (Lineal)

```
public static int linearSearch(int[] A, int n, int X){
    int i = 0;
    int index = -1;
    while(i < n && index == -1){
        if(A[i] == X){
            index = i;
        }else{
            i++;
        }
    }
    return index;
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Eliminación

Especificaciones

Objetivo:	Eliminar el elemento **item** de la lista.
Entrada:	array A, n, dato
Precondición:	array A no vacío (n>0)
Salida:	array A, n
Postcondición:	Lista A con un elemento menos, o queda igual si **item** no existe.

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Eliminación

```
public static int delete(int[] A, int n, int item){
    int pos = linearSearch(A, n, item);
    System.out.println(pos);
    if(pos != -1){
        for(int i = pos; i < n - 1; i++){
            A[i] = A[i + 1];
        }
        n--;
    }
    return n;
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

PARA ARRAYS ORDENADOS: A[1] < A[2] <……< A[n]

- Inserción
- Búsqueda Binaria
- Eliminación

Arrays Unidimensionales

- Operación De Inserción

Especificaciones

Objetivo:	Insertar el elemento ítem en la lista (array A).
Entrada:	array A, n, item
Precondición:	array A no lleno (n<TAM)
Salida:	array A, n
Postcondición:	Lista A con un nuevo elemento, siempre que no exista.

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Inserción

```
public static int insert(int[] A, int n, int item){
    boolean found = false;
    for(int i = 0; i < n && !found; i++){
        if(A[i] > item){
            for(int j = n + 1; j > i; j--){
                A[j] = A[j - 1];
            }
            found = true;
            A[i] = item;
        }
    }
    if(!found){
        A[n] = item;
    }
    n++;
    return n;
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Búsqueda Binaria

Especificaciones

Objetivo:	Buscar la posición (index) del elemento X en la lista (array A).
Entrada:	A, n, X
Precondición:	A no vacío (n>0)
Salida:	index
Postcondición:	index > 0, posición de X, index = -1 en caso contrario

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Búsqueda Binaria

```
public static int binarySearch(int A[], int n, int X){
    int left = 0, right = n, index = -1, center = 0;
    while(left <= right && index == -1){
        center = (left + right) / 2;
        if(X == A[center]){
            index = center;
        }else if(X > A[center]){
            left = center + 1;
        }else{
            right = center - 1;
        }
    }
    return index;
}
```

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Eliminación

Especificaciones

Objetivo:	Buscar la posición (index) del elemento X a eliminar
Entrada:	array A, n, X
Precondición:	array A no vacío (n>0)
Salida:	array A, n
Postcondición:	array A con un elementos menos, siempre que X exista, caso contrario A queda igual

UNIVERSIDAD DE LIMA

Arrays Unidimensionales

- Operación De Eliminación

```
public static int delete(int[] A, int n, int X){
    int pos = binarySearch(A, n, X);
    System.out.println(pos);
    if(pos != -1){
        for(int i = pos; i < n - 1; i++){
            A[i] = A[i + 1];
        }
        n--;
    }
    return n;
}
```