

Lab 1: Array Class from scratch

2720 Data Structures

Kiril Kuzmin, Akshay Juyal, Md Abdullah Al Rahat Kutubi

January 12, 2022

MAIN PROBLEM. Implement the class `Array` that imitates dynamic¹ arrays of integers. Introduce the following methods:

- ✓ `print` that prints the content of the array;
- ✓ `insert` that adds a new element to the end of the array;
- ✓ `removeAt` that removes an element at a specific index and shifts the elements to left (to get rid of “the hole”);
- ✓ `indexOf` that finds a specific element, and returns the index of its first from the left encounter; if the element is not present in the array, it returns -1 .

FUN PROBLEM. You are given an array that consists of $n \geq 0$ pairs of integers, and one unique number.

- a. Find the unique number.
- b. Find the unique number for $O(n)$.
- c. Find the unique number for $O(n)$ and using no more than $O(1)$ additional memory.

input	output
<code>[0, 2, -4, 5, 2, 0, -4]</code>	5
<code>[3, 3, 3, 3, 6, 6, 7]</code>	7
<code>[1, 1, 1, 1, 1, 1, 1, 1, 2]</code>	2
<code>[1, 0, 1, 2, 4, 2, 4]</code>	0
<code>[3]</code>	3

¹i.e., the array that automatically grows as we add elements and automatically shrinks as we remove elements

POSSIBLE SOLUTION TO THE MAIN PROBLEM.

Use it only for your reference, do not copy it. Try to implement it yourself first!

```
public class BasicsOfArrays {

    public static void main(String[] args) {
        Array numbers = new Array(3);
        numbers.insert(34);
        numbers.insert(42);
        numbers.insert(35);
        numbers.insert(42);
        numbers.insert(-20);
        numbers.print();
        numbers.removeAt(0);
        numbers.print();
        System.out.println(numbers.indexOf(42));
    }
}

// the class Array made from scratch

public class Array {
    private int[] theArray;
    private int nElements;

    public Array(int size) {
        theArray = new int[size];
    }

    public void print() {
        System.out.print("[");
        for (int i = 0; i < nElements; ++i)
            if (i < nElements-1) {
                System.out.print(theArray[i] + ", ");
            }
            else {
                System.out.println(theArray[i] + "]");
            }
    }

    public void insert(int element) {
        // if the array is full, doubles its size
        if (theArray.length == nElements) {
            int[] doubledArray = new int[nElements * 2];
            for(int i = 0; i < nElements; ++i) {
                doubledArray[i] = theArray[i];
            }
            theArray = doubledArray;
        }
        // otherwise, it adds a new element at the end of the array
        theArray[nElements] = element;
        nElements++;
    }
}
```

```

    public void removeAt(int index) {
        if (index < 0 || index >= nElements) {
            throw new IllegalArgumentException();
        }
        else {
            for(int i = index; i < nElements; ++i) {
                theArray[i] = theArray[i+1];
            }
            nElements--;
        }
    }

    public int indexOf(int element) {
        // if it finds the element, it returns the index of the first encounter
        // otherwise, it returns -1
        for(int i = 0; i < nElements; ++i) {
            if (theArray[i] == element) {
                return i;
            }
        }
        return -1;
    }
}

```