ThS.Trần Lê Như Quỳnh

ARRAY 1D

DATA STUCTURE

ThS.Trần Lê Như Quỳnh

WHAT'S ARRAY ?

DEFINITION OF ARRAY

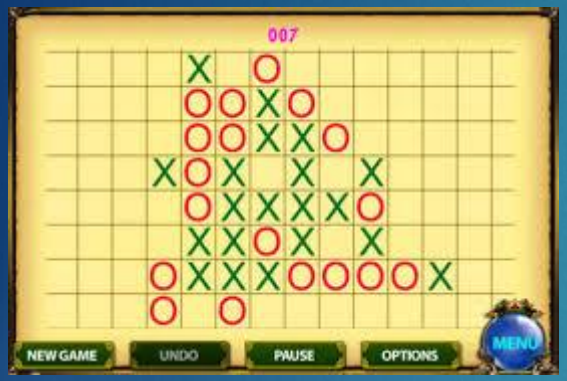
ThS.Trần Lê Như Quỳnh

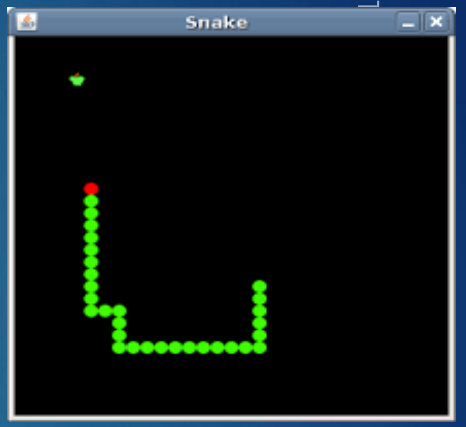
- Array IS A group of elements
  - HAS same data type (ex: String, Integer, Double)
  - HAS limit length
- Each element of Array HAS index and value (index begin 0)

ThS.Trần Lê Như Quỳnh

WHERE'S ARRAY USED?

APPLICATION OF ARRAYS





1 DIRECTION ARRAY & ALGORITHMS

ARRAY AND ALGORITHMS

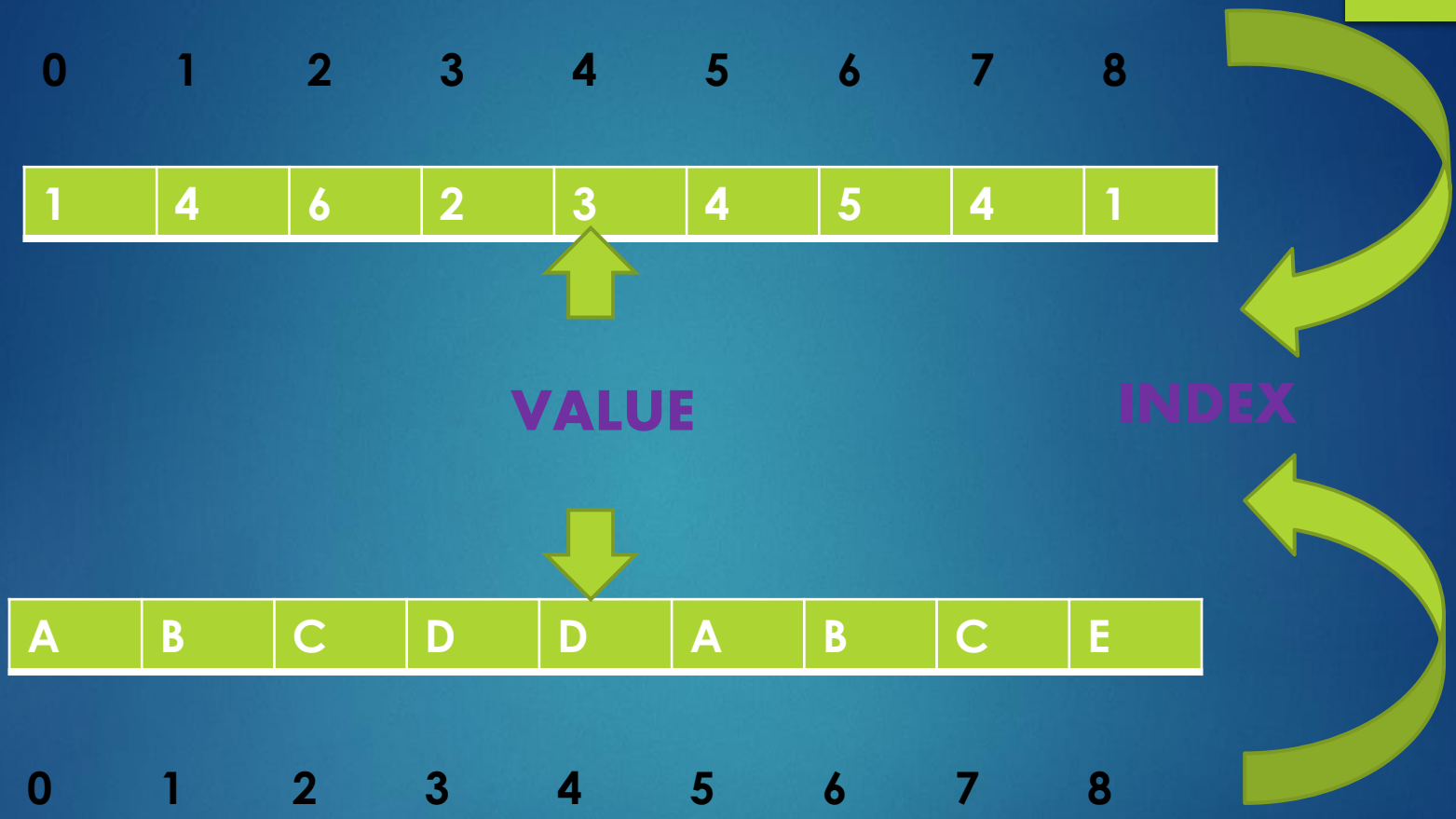
ThS.Trần Lê Như Quỳnh

ThS.Trần Lê Như Quỳnh

1 DIMENSION ARRAY

EXERCISES

DEFINITION 1D ARRAY



DECLERA, SET AND GET VALUE IN 1D ARRAY IN JAVA

- DECLERA
  - Data type[ ] or Data type[array size]
  - VD:
    - int[] demo = new int[9]; // create demo array that has 9 elements, index begin at 0 to 8
    - String[] demo1 = new String[]{“lemon”,”apple”};
- SET VALUE
  - Datatype[index] = value
  - VD:
    - demo[0] = 1; // SET begin demo index is 1
- GET VALUE
  - ArrayName[index]
  - VD:
    - **int** num = demo[0] ; // GET value of begin demo index

WORKING WITH ELEMENTS IN 1D ARRAY IN JAVA

- Normal way:

```
for (int i=0; i< array.length; i++){
array[i] = ....
}
```

- Foreach:

```
for(Datatype a : arrayName) {
a = .....
}
```

SOME SUPPORTED 1D ARRAY METHODS

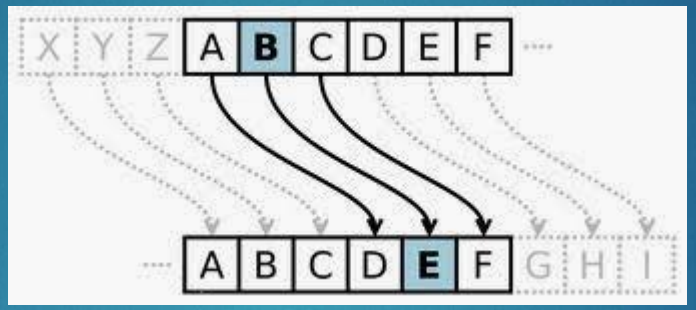
- length : return size of array
- clone(): return the copy of array
- Arrays.toString(array): print array to string
- READ MORE :
- **Chapter 3. Fundamental Data Structures, p112 + 113**

EXERCISE 1

- **TARGET: FAMILIAR WITH SOME METHODS OF ARRAY**
- **CREATE ARRAY1D_UTILS CLASS**
- **WRITE METHODS:**

1. **UPDATE VALUE**
2. **FIND VALUE**
3. **PRINT 1D ARRAY**
4. **SUM**
5. **AVERAGE**

EXERCISE 2



EXERCISE 2

- RULE:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- k: moving step
- p: index of character in alphabet
- Encrypt: $c = (p + k) \bmod 26$
- Decrypt: $p = (c - k) \bmod 26$

EXERCISE 2

- Example: with k =3, encrypt this string "VENI, VIDI, VICI"

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

"YHQL, YLGL, YLFL"

REQUIREMENT OF EXERCISE 2

1. **READ, RUN PROGRAM, TEST AND UNDERSTAND 117pg**
   - TEST STRING: ‘‘Return to Caesar what belongs to Caesar, and to God what belongs to God’’.
2. **PROGRAM EXERCISE 2 BY YOUR SELF**
3. ***UPGRADE YOUR PROGRAM HOW TO READ L.O.V.E IN TEXT FILE AND ENCRYPT, DECRYPT**

SOLUTION IN BOOK


```
/** Class for doing encryption and decryption using the Caesar Cipher. */
public class CaesarCipher {
  protected char[ ] encoder = new char[26];          // Encryption array
  protected char[ ] decoder = new char[26];          // Decryption array
  /** Constructor that initializes the encryption and decryption arrays */
  public CaesarCipher(int rotation) {
    for (int k=0; k < 26; k++) {
      encoder[k] = (char) ('A' + (k + rotation) % 26);
      decoder[k] = (char) ('A' + (k - rotation + 26) % 26);
    }
  }
  /** Returns String representing encrypted message. */
  public String encrypt(String message) {
    return transform(message, encoder);              // use encoder array
  }
  /** Returns decrypted message given encrypted secret. */
  public String decrypt(String secret) {
    return transform(secret, decoder);               // use decoder array
  }
```

BUILD UP YOUR SELF SOLUTION

- TIP:

1. CREATE MyCaesarCipher CLASS has constructor that has k input parameter
2. CREATE alphabet array that store 26 characters from A to Z
3. CREATE METHOD TO CREATE encrypted alphabet array with k parameter.
4. CREATE encrypt METHOD has input parameter is String type
5. CREATE decrypt METHOD has input parameter is String type
6. *CREATE readTextFile(String url)
7. *CREATE writeTextFile(String url)