

เนื้อหา บทที่ 4 อารีย์และสตริง

4.1 อารีย์

4.2 อารีย์หลายมิติ

4.3 String class

4.4 StringBuffer

4.5 StringTokenizer

4.6 Math class

4.7 Wrapper class



4.1 อาจารย์

อาจารย์ ชุดของตัวแปรที่มีชื่อและชนิดตัวแปรเดียวกัน จะใช้กับการทำงานที่ต้องทำงานกับตัวแปรชนิดเดียวกันหลาย ๆ ตัว เก็บอยู่ในหน่วยความจำแบบต่อเนื่องกันไป

การประกาศอาเรย์

แบบที่ 1

กำหนดนิยาม

```
dataType[] arrayName;  
//dataType []arrayName;  
//datatype arrayName[];
```

สร้างตัวแปร เป็นการสร้าง **object** ของอาเรย์

```
arrayName = new dataType[n];
```

เช่น

```
int [] score; // int score[]; // ประกาศคล้าย primitive datatype  
score = new int [10]; // allocate
```

อธิบาย

ตัวแปรอาเรย์ชื่อ **score** เป็นชนิด **int**

อาเรย์ ชื่อ **score** จองที่ **10** ที่ต่อเนื่อง เพื่อเก็บค่า **int**

สังเกตตำแหน่ง []

อยู่ชิด อยู่กลาง อยู่หน้า อยู่หลัง ได้หมด

array เป็น object ดังนั้น เราต้องใช้คำหลัก **new** เพื่อ actually create array

ตัวดำเนินการ **new** จะจองเนื้อที่หน่วยความจำให้ **object** ของ
อาร์เรย์

```
จากตัวอย่าง int[] score; //ประกาศ  
score = new int [10]; //จอง 10 ที่สำหรับสมาชิก
```

เมื่อใช้ตัวดำเนินการ **new** สร้าง **object** ของอาร์เรย์ เราต้องระบุ
จำนวนห้องว่างที่ใช้บรรจุสมาชิก



แบบที่ 2

นำ 2 ชั้นมารวมกัน

```
dataType[] arrayName = new dataType[arraySize];  
//dataType arrayName[] = new dataType[arraySize];
```

เช่น

```
int[] score = new int [10];
```

อธิบาย

สร้างอาร์เรย์ชื่อ **score** จองที่ต่อเนื่อง 10 ที่ มีชนิดเป็น **int**

1. สรุปคือวิธี 1. หรือ 2.

```
int[] score;  
score = new int [10];
```

2.

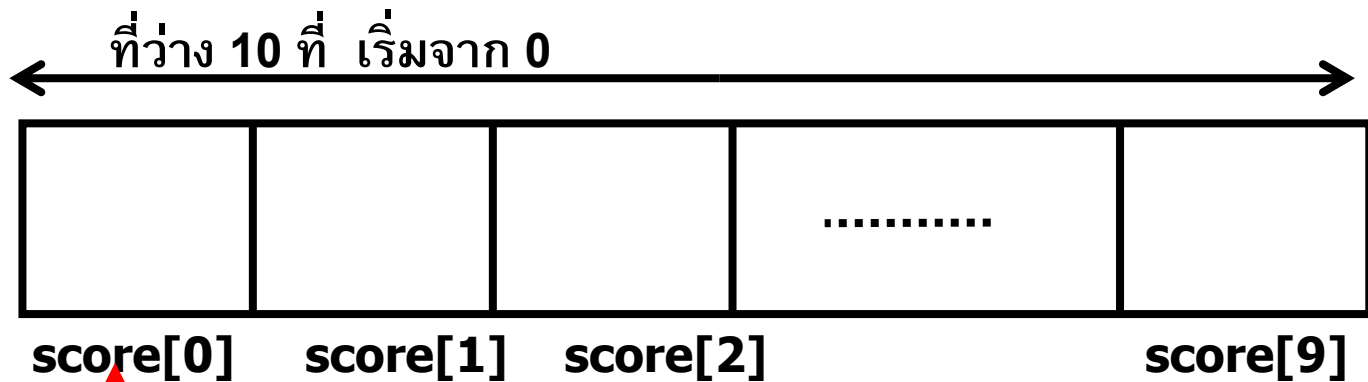
```
int[] score = new int [10];
```

การอ้างถึงค่าที่เก็บในอาร์เรย์

ภาษาจาวาจะนับตัวเลขลำดับที่ในอาร์เรย์ โดย **เริ่มต้น** ที่ **0**

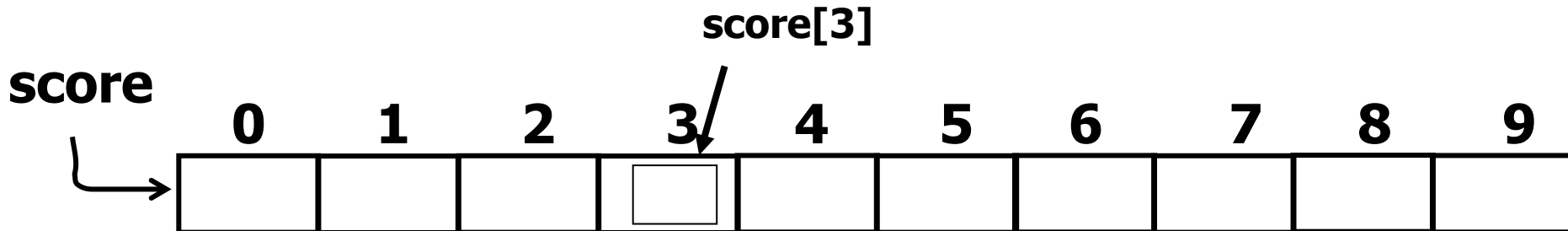
ไปจนถึง **arraySize-1**

ดังนั้นจากตัวอย่าง



ตำแหน่งแรกของ **array** คือ

ตำแหน่งที่ **0**



การอ้างอิง พร้อมกำหนดค่าตัวแปรในอาเรย์

การอ้างอิงถึงตัวแปรในอาเรย์

arrayName[index];

index คือตัวเลข 0 ถึง **arraySize-1**

การกำหนดค่าให้

score[2]=45;

หรือกำหนดตอนสร้าง คือประกาศอาเรย์พร้อมกำหนดค่าเริ่มต้น

double[] myArray = {455,200,712};

int[] number = { 1, 2, 3, 4, 5 };

double[] samplingData = { 1.35, 8.57, 2.38,45.009, 18.2, 9.00};

String[] monthName = {"January", "February"};

นอกจากนี้เรายังมีคุณลักษณะหรือ **attribute** ที่ชื่อว่า **length** ไว้ขนาดของอาร์เรย์

arrayName.length;

ตัวอย่าง

```
class C4Array1 {  
    public static void main(String[] args)  
    {  
        int[] a = {70,20,79,90};  
        System.out.println(a.length);  
    }  
}
```

4

ตัวอย่าง

```
public class C4Array2  
{  
    public static void main(String[] args)  
    {  
        int a[];  
        a = new int [10];  
        a[0]=1;  
        a[1]=4;  
        a[2]=11;  
        System.out.println(a.length);  
    }  
}
```

```

public class C4Arrchar
{
    public static void main(String[] args)
    {
        char a[];
        a = new char [10];
        a[0]='a';
        a[1]='b';
        a[2]='c';
        a[9]='b';
        for (int i =0; i < a.length; i++ )
        {
            System.out.print(a[i]+ " ");
        }
        System.out.println("\n"+a.length);
    }
}

```

```

a b c _ _ _ _ _ b
10

```

_ _ _ _ _ space

```
public class C4Arr1Dim
{
    public static void main(String[] args)
    {
        int i;
        int a[];
        a = new int[10];
        for (i=0; i<a.length ;i++ )
            a[i] = i;
        for (i=0; i < a.length ;i++ )
            System.out.print(a[i] + " ");
        System.out.println();
    }
}
```

```
----- Java Run -----
0  1  2  3  4  5  6  7  8  9
Normal Termination
Output completed (0 sec consumed).
```

สรุป อารีย์

1.

```
int[] score;
```



```
score = new int[10];
```



```
score[0] = 12;  
score[1] = 150";
```

```
int [] score = new int [10];
```

ขั้นที่ 1 การประกาศตัวแปรอารีย์

ประกาศชื่อตัวแปรอารีย์ตามต้องการ

ขั้นที่ 2 การสร้าง object ของอารีย์ใหม่

ใช้ตัวดำเนินการ new จองหน่วยความจำ

ให้อารีย์ตามจำนวนสมาชิกที่ต้องการ

ขั้นที่ 3 กำหนดค่าต่างๆ ให้อารีย์

กำหนดค่าต่างๆ ที่ต้องการบรรจุลงในสมาชิกของอารีย์

2.

สร้างอารีย์ ตามชื่อที่กำหนด มีจำนวนสมาชิกตามที่กำหนด

```
int[] score = new int [10];
```

3.

สร้างพร้อมพร้อม กำหนดค่าเริ่มต้น

```
int[ ] score = { 1, 2, 3, 4, 5 };
```

แบบฝึกหัด

1. ข้อใดถูกต้อง? หากข้อใด ผิดให้อธิบายเหตุผลและแก้ไขให้ถูกต้อง

- a. `float number[23];`
- b. `float number = {1.0f,2.0f,3.0f};`
- c. `int number;`
`number = new Array[23];`
- d. `int[] number = [1,2,3,4];`

2. จงเขียน code เพื่อคำนวณผลรวมของทุกค่าที่ store ใน array
`double[] number = new double[25];`

3. จงหาผลลัพธ์ของโปรแกรมต่อไปนี้

```
public class C4Str9 {
    public static void main(String[] args) {
        int[] number= {1,2,3,4,5,6,7,8,9,10};
        for (int i=0; i<number.length; i++)
            if (i%2 == 0)
                System.out.print(number[i]+" ");
        System.out.println("\n");
        for (int i=0; i<number.length; i++)
            if (number[i]%2 == 0)
                System.out.print(number[i]+" ");
    }
}
```

```
public class C4ArrayEx4 {
    public static void main(String[] args) {
        int a ;
        char [] j = {'P','r','o','g','r','a','m','m','i','n','g'};
        for (a=0; a < j.length ;a++ )
            if (a%2==0)
                System.out.print(j[a]+" ");
        System.out.print("\n"+j.length+"\n");
    }
}
```

4.2 อาร์เรย์หลายมิติ

การสร้างอาร์เรย์มากกว่า 1 มิติ ทำได้โดยการใส่เครื่องหมาย [] เพิ่มลงไป

1) อาร์เรย์ 2 มิติ

รูปแบบ

ประกาศ

```
data_type ชื่อตัวแปร [][ ];
```

```
// data_type[ ][ ] ชื่อตัวแปร;
```

สร้าง

```
ชื่อตัวแปร = new data_type [ขนาด] [ขนาด];
```

`a[row][column]` เวลานั้น `a.length` จะแสดงผลด้วยขนาดของ row

หรือ

`a[row][column]`

ประกาศและสร้าง

row column

```
float[][] matrix= new float[4][5];
```

```
หรือ float matrix[][]= new float[4][5];
```

	0	1	2	3	4
0	10.50	12.00	14.50	16.75	18.00
1	20.50	22.25	24.00	26.25	28.00
2	34.00	36.50	38.00	40.35	43.00
3	50.00	60.00	70.00	80.00	99.99

`matrix[1][2]`

`matrix[2][3]`

การกำหนดค่าเริ่มต้น

การใส่ค่าจะเรียงไปตาม **row** ก่อน

โดยใช้เครื่องหมาย **{ }** คั่นในแต่ละ **row**

และใช้เครื่องหมาย, คั่น ในแต่ละ **column**

	0	1
0	1	2
1	3	4
2	5	6

```
int[][] number = {{1,2},{3,4},{5,6}};  
double[][] samplingData = {{1.35, 8.57, 2.38},  
                             {45.009, 18.2, 9.00},{3.123, 22.084, 18.08}};
```

```
String[][] monthName = {{"1","January"},  
                          {"2", "February"}};
```

หรือ

```
String S1 = "1";  
String S2 = "January";  
String S3 = "2";  
String S4 = "February";  
monthName[] = {{S1, S2},{S3,S4}};
```

จวงวาดรูป การจัดเก็บที่
เห็นมิติ(row column)
เห็นข้อมูลที่เก็บ ดูตย.ข้างบน

```
public class MultiDimArrayDemo
{
    public static void main(String[] args)
    {
        String[][] names = { {"Mr. ", "Mrs. ", "Ms. "},
                              {"Smith", "Jones"} };
        System.out.println(names[0][0] + names[1][0]);
        System.out.println(names[0][2] + names[1][1]);
    }
}
```

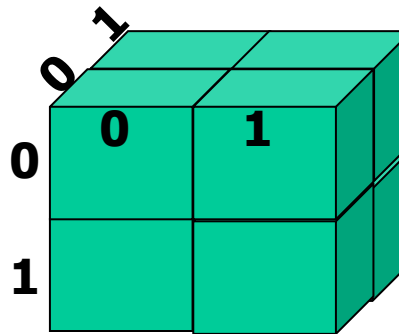
ในกรณีที่เป็นอาร์เรย์ 3 มิติ

ใส่เครื่องหมาย [] เพิ่มเข้าไป

เช่น

```
int[][][] threeDim = new int[2][2][2]
```

ตัวอย่างอาร์เรย์



attribute ชื่อ **length** แสดงขนาดของอาร์เรย์

กรณีเป็นอาร์เรย์สอง สามมิติ

```
public class C4A2D
{
    public static void main(String[] args)
    {
        int i;
        int[][][] a = new int[5][10][15];
        System.out.println(a.length);
        System.out.println(a[0].length);
        System.out.println(a[0][0].length);
    }
}
```

----- Java run -----

5

10

15

```

public class C4Arr1
{
    public static void main(String[] args)
    {
        int[][] m={ {1,2,3,4,5} ,
                    {6,7,8,9,10} ,
                    {11,12,13,14,15} ,
                    {16,17,18,19,20} };

        int i,j;
        for (i=0;i<3 ;i++ )
            { for (j=0;j<4 ;j++ )
                System.out.print(m[i][j] + " ");
                System.out.println(" ");
            }
    }
}

```

1	2	3	4
6	7	8	9
11	12	13	14

```

public class C4Arr2
{
    public static void main(String[] args)
    {
        int[][] twoDim = new int [4][5];
        int i,j,k=0;
        for (i=0;i<4 ;i++ )
            for (j=0;j<5 ;j++ ) {
                twoDim[i][j]=k;
                k++;
            }
        for (i=0;i<4 ;i++ ) {
            for (j=0;j<5 ;j++ )
                System.out.print(twoDim[i][j] + " ");
            System.out.println(" ");
        }
    }
}

```

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

```
public class C4Arr3
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int twoDim[][] = new int [4][4];
```

```
        int i,j,k=0;
```

```
        for (i=0;i<4 ;i++ )
```

```
            for (j=0;j< i+1 ;j++ ) {
```

```
                twoDim[i][j]=k;
```

```
                k++;
```

```
            }
```

```
        for (i=0;i<4 ;i++ ) {
```

```
            for (j=0;j<i+1 ;j++ )
```

```
                System.out.print(twoDim[i][j] + " ");
```

```
            System.out.println(" ");
```

```
        }
```

```
    }
```

```
}
```

0			
1	2		
3	4	5	
6	7	8	9

```

class C4Arr4 {
    public static void main(String[] args) {
        int twoDim[][][] = new int [3][4][5];
        int i,j,k;
        for (i=0;i<3 ;i++ )
            for (j=0;j<4 ;j++ )
                for (k=0;k<5 ;k++ )
                    twoDim[i][j][k]=i*j*k;
        for (i=0;i<3;i++ ) {
            for (j=0;j<4 ;j++ ) {
                for (k=0;k<5 ;k++ )
                    System.out.print(twoDim[i][j][k]+" ");
                System.out.println(" ");
            }
            System.out.println(" ");
        }
    }
}

```

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	2	3	4
0	2	4	6	8
0	3	6	9	12
0	0	0	0	0
0	2	4	6	8
0	4	8	12	16
0	6	12	18	24

การทำงานกับอักขระ

method ที่ทำงานกับ Character class

1) isLowerCase(char ch)

ใช้ตรวจสอบว่าตัวอักขระนั้นเป็นตัวเล็กหรือไม่

ถ้าใช่จะส่งผลเป็น true

return true/false

2) isUpperCase(char ch)

ใช้ตรวจสอบว่าตัวอักขระนั้นเป็นตัวใหญ่หรือไม่

ถ้าใช่จะส่งผลเป็น true

return true/false

3) toLowerCase(char ch)

ใช้เปลี่ยนตัวอักขระนั้นให้เป็นตัวเล็ก

return char

4) toUpperCase(char ch)

ใช้เปลี่ยนตัวอักขระนั้นให้เป็นตัวใหญ่

return char

5) `isDigit()` ใช้ตรวจสอบว่าตัวอักษรนั้นเป็นตัวเลขหรือไม่
ถ้าใช่จะส่งผลเป็น `true`

`return true/false`

6) `isLetter()` ใช้ตรวจสอบว่าเป็นตัวอักษรหรือไม่
ถ้าใช่จะส่งผลเป็น `true`

`return true/false`

7) `isLetterOrDigit()` ใช้ตรวจสอบว่าตัวอักษรนั้นเป็นตัวเลข
หรือตัวอักษร
ถ้าใช่จะส่งผลเป็น `true`

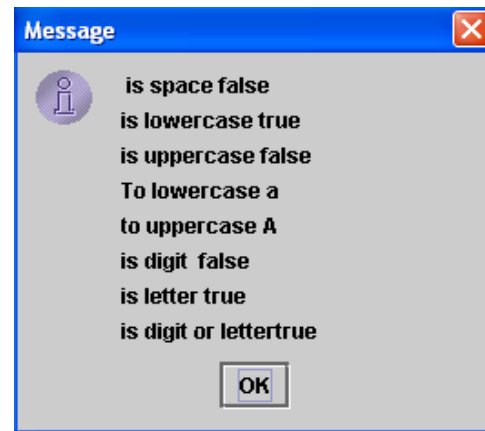
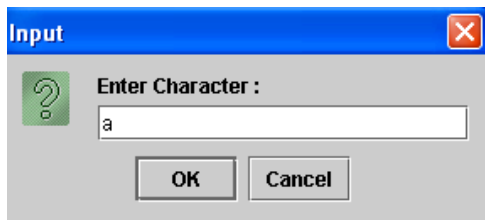
`return true/false`

ตัวอย่าง

```

import javax.swing.*;
public class Carr {
    public static void main(String[] args) {
        String data,message;
        char c;
        data = JOptionPane.showInputDialog("Enter Character :");
        c = data.charAt(0);
        message = " is space " + Character.isSpace(c) +
            "\nis lowercase " + Character.isLowerCase(c) +
            "\nis uppercase " + Character.isUpperCase(c) +
            "\nTo lowercase " + Character.toLowerCase(c) +
            "\nto uppercase " + Character.toUpperCase(c) +
            "\nis digit " + Character.isDigit(c) +
            "\nis letter " + Character.isLetter(c) +
            "\nis digit or letter " + Character.isLetterOrDigit(c);
        JOptionPane.showMessageDialog(null,message);
        System.exit(0);
    }
}

```



4.3 String class

String เป็น class ใน package java.lang

ทำหน้าที่เก็บข้อมูลที่เป็น "ชุดของตัวอักษร"

ปกติ ชนิดข้อมูลพื้นฐานคือ char คือ 1 ตัวอักษร

หากต้องการทำงานกับอักขระหลายๆ ตัว

เราจะใช้ **String** class

คำภาษาไทย → สายอักขระ

1) การสร้าง **Object** ของ **Class String**

รูปแบบ 1 การประกาศแบบสั้น

```
String objectName = message;
```

เช่น

```
String name1 = "Pooh";
```

```
String name2;  
name2 = "Piglet";
```

การประกาศแบบสั้นใช้ได้เฉพาะสำหรับ **class String** เท่านั้น !!

รูปแบบ 2 การประกาศในลักษณะ เป็น **object** ของ **class**

```
String objectName = new String(message);
```

ความหมาย

objectName คือ ชื่อของ **object**

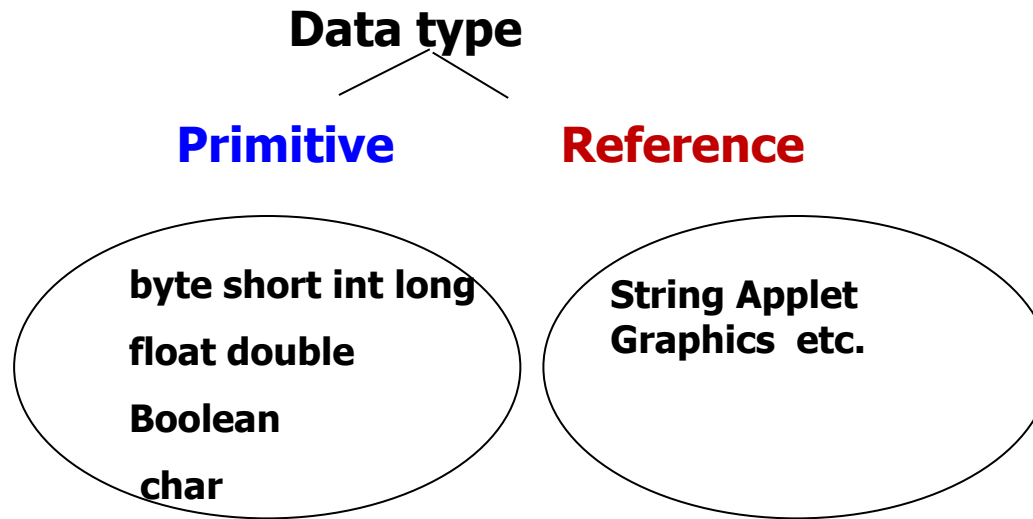
message คือข้อความ

เช่น

```
String name1 = new String ("Pooh");
```

```
String name2;  
name2 = new String ("Piglet");
```

Data type classified เป็น 2 group

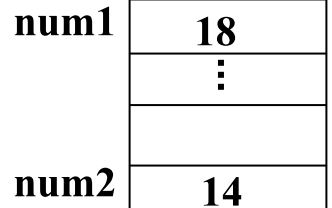
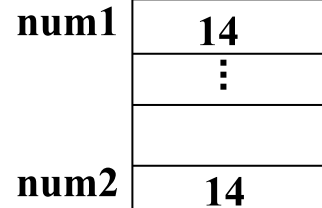
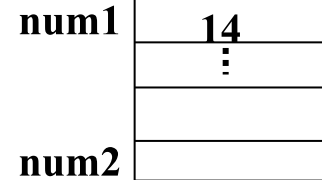
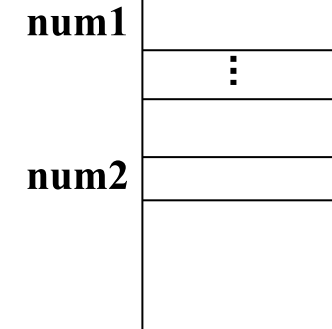


Data type classified เป็น 2 group คือ primitive and reference
primitive – short byte int long , float double , char, boolean
reference จะเป็นพวก object ของ class เราจะเรียนต่อไป

การจองเนื้อที่ในหน่วยความจำของ **Primitive data type**

state of memory

```
int num1 , num2;  
num1=14;  
num2=num1;  
num1 +=4;
```



การจัดเก็บใน
หน่วยความจำเป็น
ลักษณะ direct access

กรณีการประกาศ class String
การทำงานกับ operator ==
และ method equals

```
String name1="java";
```

```
String name2=new String("java");
```

```
System.out.println(name1==name2);
```

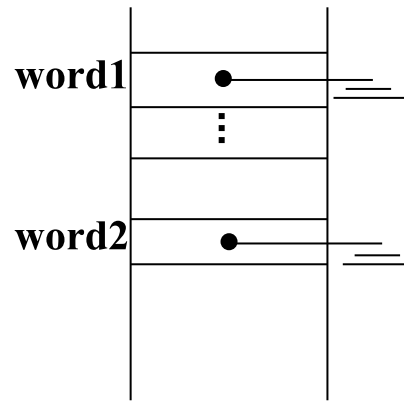
```
System.out.println(name1.equals(name2));
```

```
String word1 , word2;
word1 = new String("Java");
word2 = word1;
```

== ✓
equals() ✓

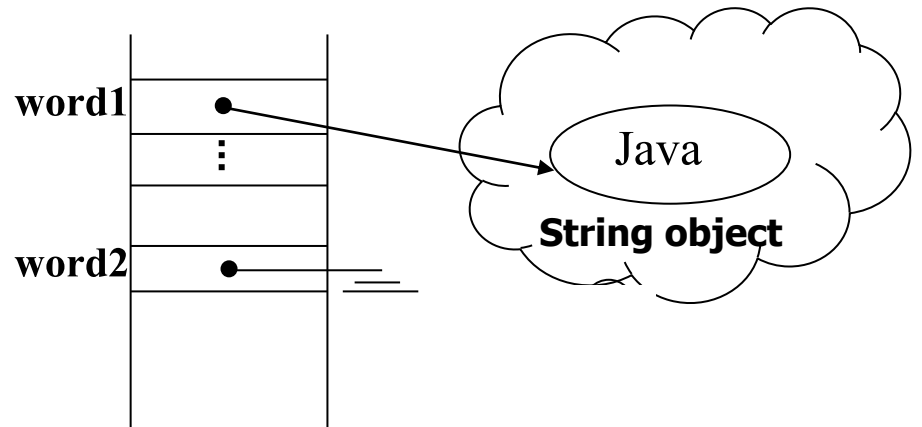
= assignment statement เท่ากัน
ทุกกรณีทั้งสิ่งที่เก็บและ ref.

state of memory



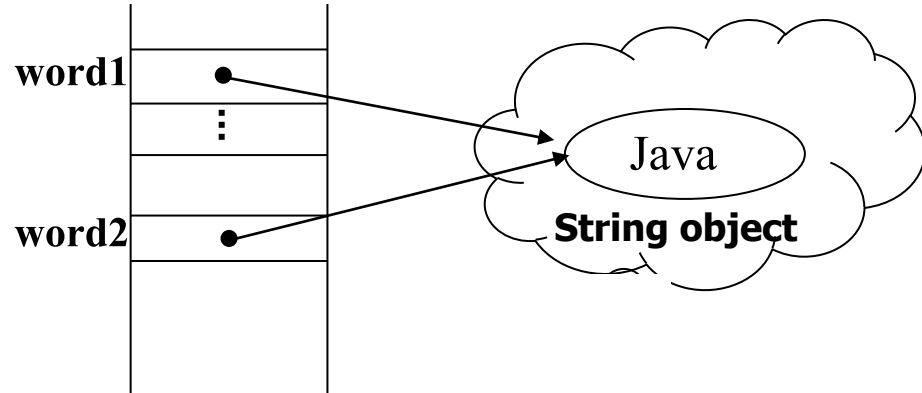
เวลาเราเปรียบเทียบ string

== equality operator
equals() method เปรียบเทียบ
string return true/false



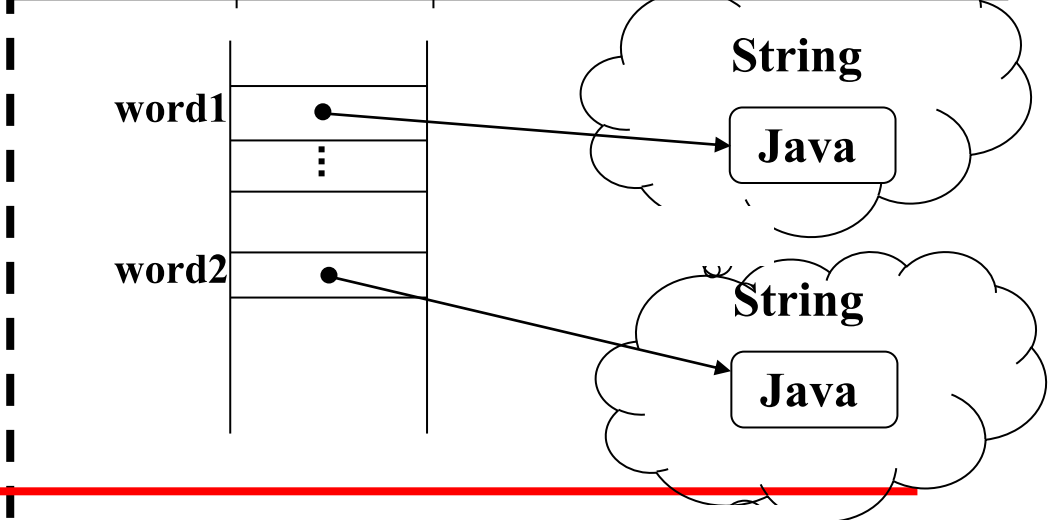
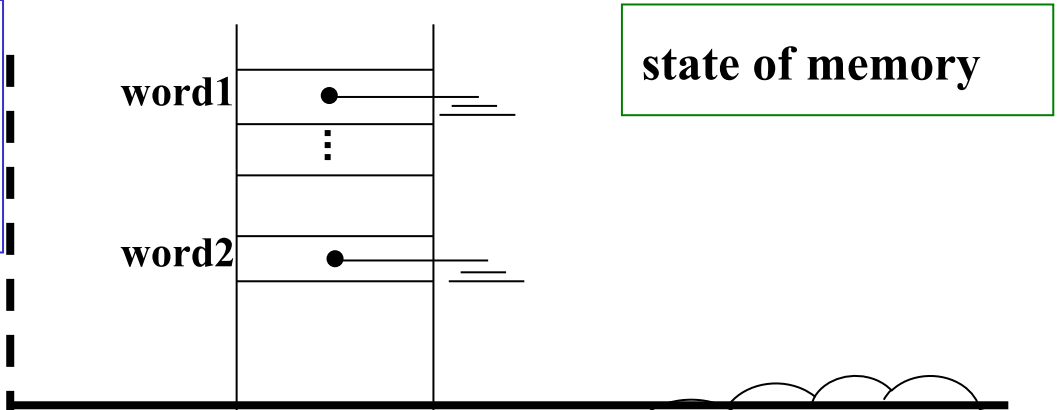
if(str1.equals(str2)) → true

if(str1==str2) → true



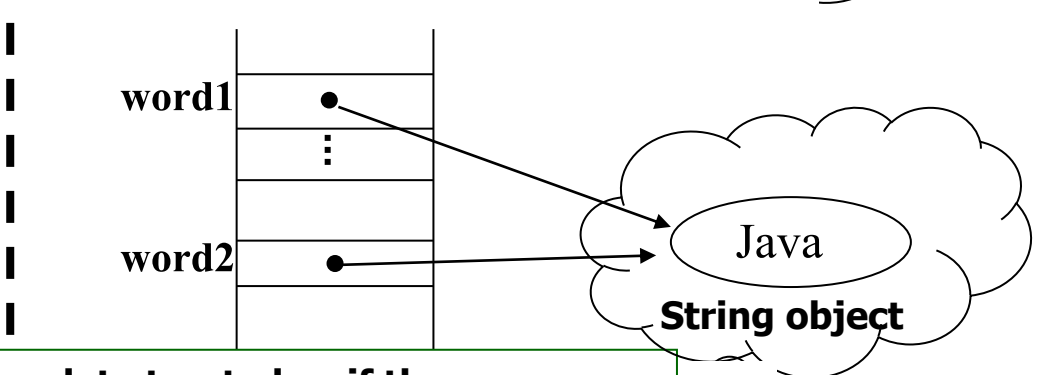
```
String word1 , word2;  
word1 = new String("Java");  
word2 = new String ("Java");
```

== x
equals() ✓



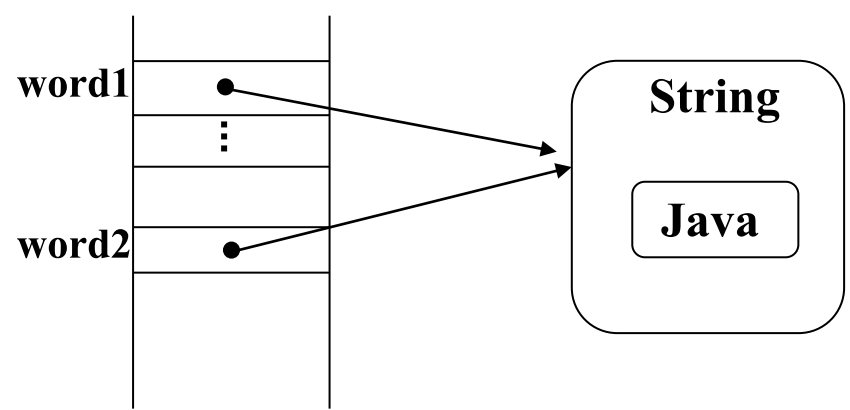
```
String word1 , word2;  
word1 = "Java";  
word2="Java";
```

== ✓
equals() ✓



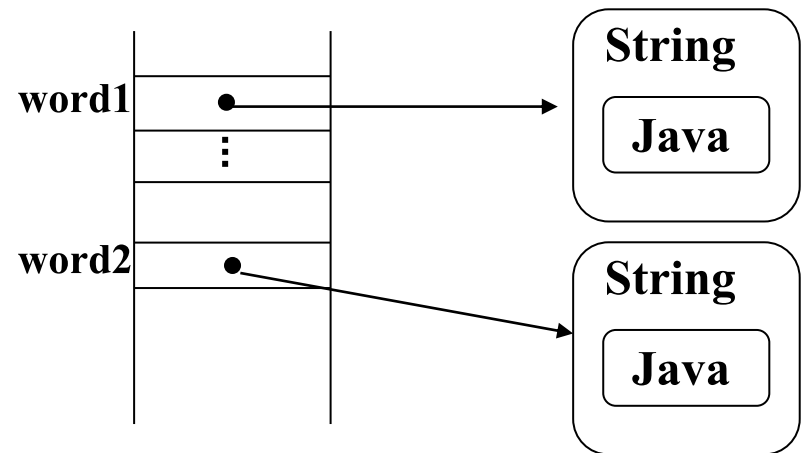
if we **do not use** the **new** operator, then string data treated as if they are **primitive data type**. When we **use the same literal String** constants in a program ,there will be **exactly one String object**

`word2 == word1;` is true
`word1.equals(word2);` is true

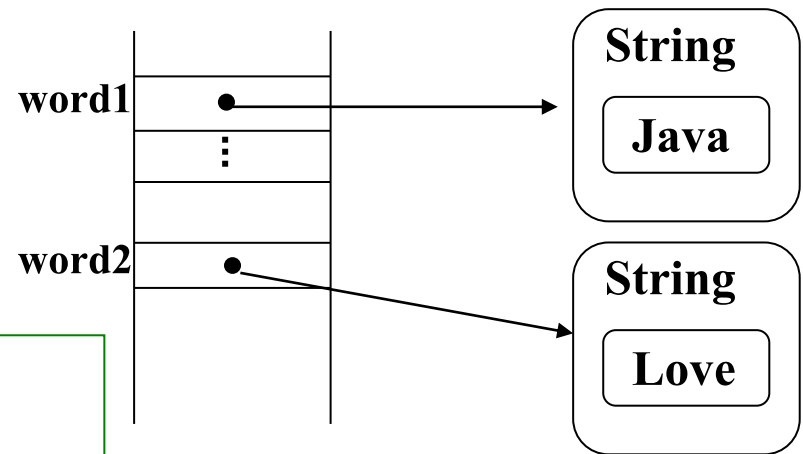


ถ้า `== false` แล้ว `equal` อาจ `true/false` ก็ได้

1
`word2 == word1;` is false
`word1.equals(word2);` is true



2
`word2 == word1;` is false
`word1.equals(word2);` is false



Note if `x == y` is true, then `x.equals(y)` is also true. The reverse is not always true.

จงวาด state of memory เพื่อดูการจัดเก็บข้อมูล

1. String word1,word2;
word1 ="Hello";
word2 = word1;

equals() ตอบ true/false
== ตอบ true/false

2. String word1,word2;
word1 ="Hello";
word2 ="Hello";

equals() ตอบ true/false
== ตอบ true/false

3.String word1,word2;
word1 =new String("Hello");
word2 =new String("Hello");

equals() ตอบ true/false
== ตอบ true/false

4. String word1,word2;
word1 =new String("Hello");
word2 =word1;

equals() ตอบ true/false
== ตอบ true/false

Practice

show state of memory after the following statements are execute

```
String str1,str2,str3;  
str1="Love";  
str2="Love";  
str3=str2;  
str2=str1;
```

```
String str1,str2,str3;  
str1= new String ("Love");  
str2= new String ("Love");  
str3=str2;  
str2=str1;
```

2) String operation

คำสั่งที่ใช้ในการทำงานกับ String ตั้งแต่ 2 String Object

1. concatenation เชื่อมต่อข้อความ 1.1 ใช้ + 1.2 method concat()
2. การเปรียบเทียบ string 2.1 equals() 2.2 compareTo()
3. substring() การตัดข้อความบางส่วนของ String
4. replace() การเปลี่ยนตัวอักษรใน string ด้วยตัวอักษรอื่น
5. charAt() การอ้างถึงตัวอักขระ ณ ตำแหน่งใดในString
6. toUpperCase() การแปลงอักขระจากตัวพิมพ์เล็กเป็นตัวพิมพ์ใหญ่
toLowerCase() การแปลงอักขระจากตัวพิมพ์ใหญ่เป็นตัวพิมพ์เล็ก
7. length() การหาความยาวของ String
8. trim() ตัดตัวอักษร white space ที่อยู่หน้าและต่อท้ายออก

1. Concatenation

เป็นการนำ **string object** ตั้งแต่ 2 ตัวขึ้นไปมาเชื่อมต่อกัน

1.1 ใช้เครื่องหมาย +

รูปแบบ

```
String myString = "message" + " and " + "message"ว
```


1.Concatenation

1.2 method ที่ชื่อ **concat()** เพื่อใช้ในการเชื่อมต่อข้อความ

รูปแบบ

String1 concat(String2)

ตัวอย่าง

```
System.out.println("Nobi "+ "how "+ "do you do?");  
System.out.println("I am "+"10 years old.");  
String a = "Nice ";  
System.out.println(a.concat("to meet you."));
```

Nobi how do you do?

I am 10 years old.

Nice to meet you.

2. การเปรียบเทียบ **string**

- **equals()**
- **compareTo()**

equals()

ทำการเปรียบเทียบค่าของ **String** 2 ตัวว่าเท่ากันทุกตัวอักษรหรือไม่

รูปแบบ **str1.equals(str2)**

method นี้จะให้

ค่าเป็น **true** เมื่อ **str1** และ **str2** เท่ากัน

ค่าเป็น **false** ถ้า **str1** และ **str2** ไม่เท่ากัน

ตัวอย่าง

```
String data1,data2,message;
```

```
data1 = JOptionPane.showInputDialog("Enter String1 :");
```

```
data2 = JOptionPane.showInputDialog("Enter String2 :");
```

```
if (data1.equals(data2))
```

```
    message = " same string ";
```

```
else
```

```
    message = " different string ";
```

```
String s0 = " Java";  
String s1 = "Welcome to" + s0;  
String s2 = "Welcome to Java" ;  
System.out.println("s1 == s2 is " + (s1==s2));  
System.out.println("s1.equals(s2) is " + (s1.equals(s2)));
```

----- **Java run** -----

s1 == s2 is **false**

s1.equals(s2) is **true**

- compareTo()

ทำการเปรียบเทียบค่าของ **String** 2 ตัวว่าเท่ากันหรือไม่

รูปแบบ **str1.compareTo(str2)**

โดยผลที่ได้จาก **method** นี้จะมี 3 ค่า

ค่าเป็นศูนย์ (0) ถ้า **str1** และ **str2** เท่ากัน

ค่าเป็นบวก (+) ถ้า **str1** มีค่ามากกว่า **str2**

ค่าเป็นลบ (-) ถ้า **str1** มีค่าน้อยกว่า **str2**

ค่าที่ return จาก compareTo() depend on the offset of the first two distinct characters in str1 and str2 from left to right.

```
เช่น str1 = "abc";
```

```
    str2 = "abg";
```

```
    System.out.println(s1.compareTo(s2)); // -4
```

ตัวอย่าง

```
int result;  
String data1,data2,message;  
data1 = "test";  
data2 = "test";  
result = (data1.compareTo(data2));  
if (result == 0)  
    message = "same string ";  
else if (result > 0)  
    message = "more than ";  
else  
    message = "less than ";
```

test	test	same string
test01	test1	less than
Test	test	less than

ABC..Zabc..z
↓ ↓
65 97

3. substring() การตัดข้อความบางส่วนของ String

เป็นการนำข้อความบางส่วนของ **string** ออกมา

รูปแบบ

String substring(int beginIndex);

String substring(int beginIndex ,int endIndex);

ความหมาย

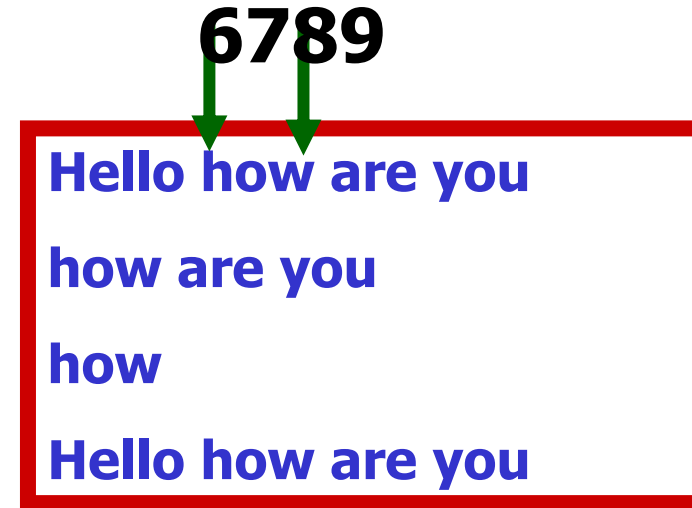
แบบ 1 จะให้ **String** ย่อยของ **String** ตั้งแต่ตำแหน่ง **beginIndex** คือ ตำแหน่งเริ่มต้น ไปจนจบ

แบบ 2 จะให้ **String** ย่อยของ **String** ตั้งแต่ตำแหน่ง **beginIndex** คือ ตำแหน่งเริ่มต้น ไปจนถึง **endIndex-1**

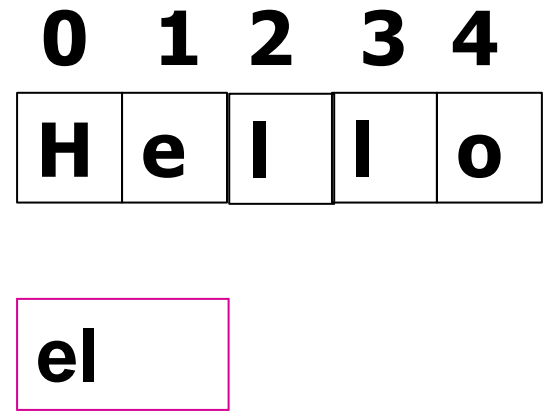
โดย **method** นี้จะส่งค่ากลับเป็น **String**

ตัวอย่าง เช่น

```
String a="Hello how are you";  
System.out.println(a);  
System.out.println(a.substring(6));  
System.out.println(a.substring(6,9));  
System.out.println(a);
```



```
public class C4Str9 {  
    public static void main(String[] args) {  
        String said="Hello";  
        System.out.println(said.substring(1,3));  
    }  
}
```



**String x = "0123456789"; // as if by magic, the value of each char
// is the same as its index!**

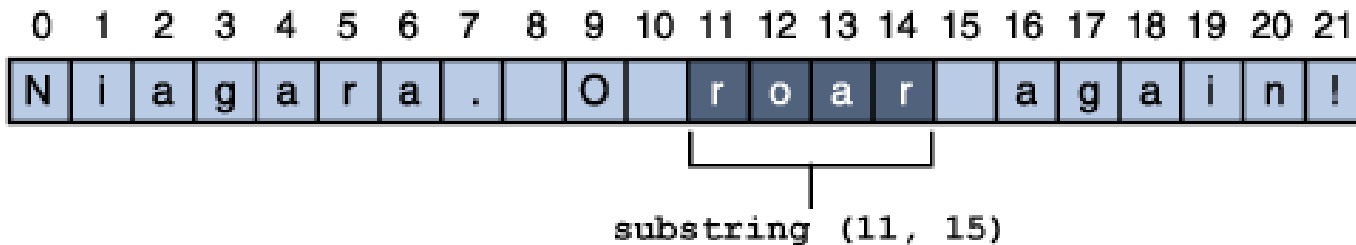
System.out.println(x.substring(5)); // output is "56789"

System.out.println(x.substring(5, 8)); // output is "567"

The first example should be easy: start at index 5 and return the rest of the String.

The second example should be read as follows: start at index 5 and return the characters up to and including the 8th position (index 7).

ตย.อื่นๆ เช่น



4. `replace()` เป็นการเปลี่ยนตัวอักษรใน `String`

รูปแบบ

`String replace(char oldChar, char newChar);`

`String` ที่เป็นผลลัพธ์จะเป็น `String` เดิม ตัวอักษร `oldChar` จะถูกเปลี่ยนเป็น `newChar`

เช่น `String said = "Hello";`

`String newSaid = said.replace('l','L');`



Hello
HeLLO

5. `charAt()` การอ้างอิงถึงแต่ละอักขระ ณ ตำแหน่งที่ต้องการใน String

→ return char

รูปแบบ `charAt(ตำแหน่ง)`

ตำแหน่งเริ่มต้นของอักขระในตัวแปร String คือ ตำแหน่งที่ 0

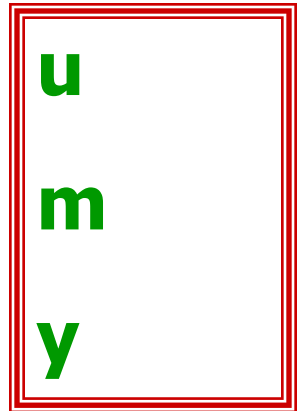
เช่น

```
String name = "Dummy";
```

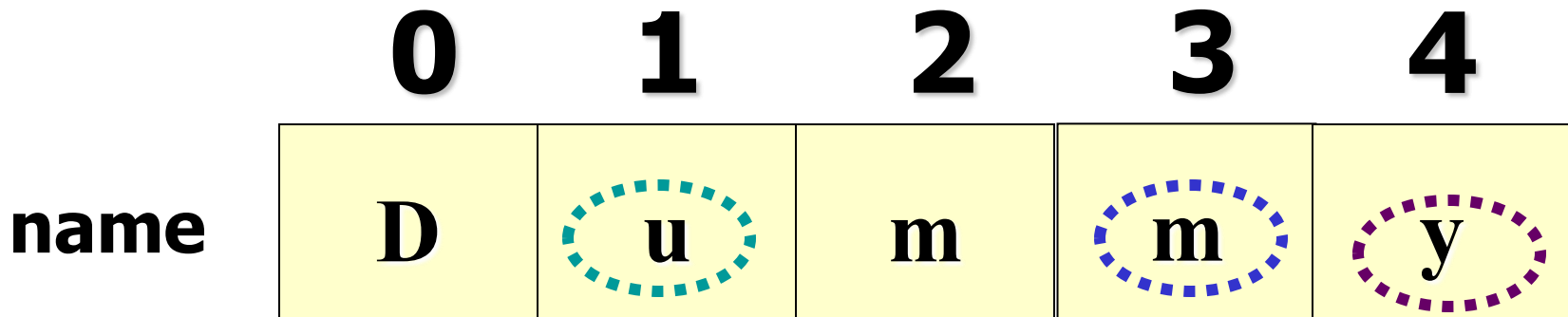
```
System.out.println(name.charAt(1));
```

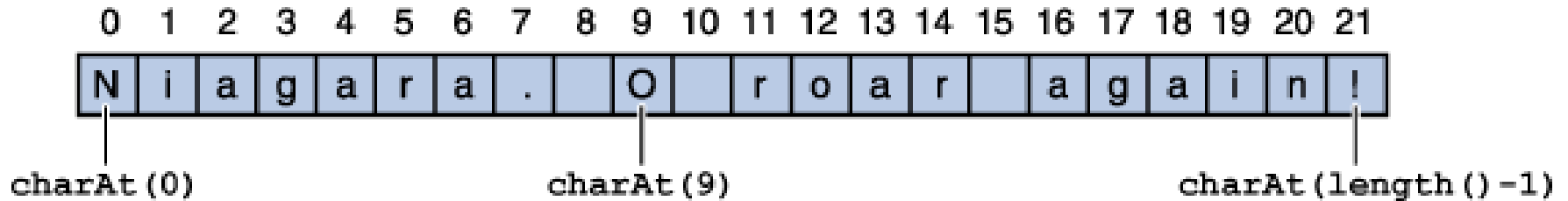
```
System.out.println(name.charAt(3));
```

```
System.out.println(name.charAt(4));
```



u
m
y





chap2

```
//Input char data
```

```
data = JOptionPane.showInputDialog("Enter character value:");
```

```
ch = data.charAt(0);
```

6. การเปลี่ยนตัวอักษร

- **toUpperCase()** เปลี่ยนเป็นตัวพิมพ์ใหญ่
- **toLowerCase()** เปลี่ยนเป็นตัวพิมพ์เล็ก

ให้ผลลัพธ์เป็น **String** เช่น

```
String name = "What it does ?";  
System.out.println(name.toUpperCase());  
System.out.println(name.toLowerCase());  
System.out.println(name);
```

WHAT IT DOES ?

what it does ?

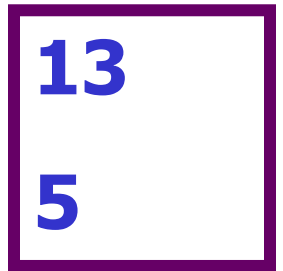
What it does ?

7. **length()** ระบุขนาดของ **string**

เป็น **method** ใน **class string**

เช่น

```
String name1 = "What it does?";  
String name2 = "Dummy";  
System.out.println(name1.length());  
System.out.println(name2.length());
```



13
5

8. `trim()` ตัด `space` ที่อยู่หน้าและต่อท้าย `string` ออก

เช่น

```
String name1 = "  Dummy  ";  
System.out.println(name1+name1.length());  
System.out.println(name1.trim()+name1.length());  
String a = name1.trim();  
System.out.println(a+a.length());
```

Dummy 20

Dummy20

Dummy5

package java.lang class String

บทที่ 4 อาร์เรย์และสตริง

✓4.1 อาร์เรย์

✓4.2 อาร์เรย์หลายมิติ

✓4.3 String class

1. concat() 2.compareTo() 3.equal() 4.substring() 5.replace() 6.charAt() 7.length() 8.trim()

4.4 StringBuffer

4.5 StringTokenizer

4.6 Wrapper class

4.7 Math class

4.4 StringBuffer

เราต้องการ **manipulate** กับ **String** แต่ **String** มีลักษณะ **immutable** คือ **never change** , **forzen solid**

เป็น **class** ที่ทำงานกับ สตริง

มีความยืดหยุ่นและต้องการเปลี่ยนแปลงค่าใน สตริงนั้น

เช่น ต้องการแก้ไข ตัวอักขระ หรือทำการแทรกอักขระหรือทำการลบ อักขระบางตัวออกจากข้อความ

การประกาศ **object** ของ **class StringBuffer**

```
StringBuffer inpData = new StringBuffer();
```

method ที่เราจะเรียน

- **append** (อักขระใหม่)
- **insert** (ตำแหน่งที่ต้องการ,อักขระใหม่)
- **setCharAt**(ตำแหน่งที่ต้องการ,อักขระใหม่)

- **append()** เป็นการเพิ่มความใน **string**

เช่น

```
StringBuffer name = new StringBuffer("Java");  
name.append("A");  
System.out.println(name);
```

เช่น

```
StringBuffer name = new StringBuffer();  
name.append("Java ");  
name.append("with ");  
name.append("love");  
System.out.println(name);
```

JavaA

Java with love

-insert(ตำแหน่งที่ต้องการแทรก,ข้อความที่ต้องการแทรก)

เป็นการเพิ่มข้อความใน **String** เดิม จะไม่ทับข้อความเดิมเป็นการแทรกข้อมูล

ตัวอย่าง

```
StringBuffer can = new StringBuffer("Java");  
can.insert(2,'+');
```

0	1	2	3	4
J	a	+	v	a

ตัวอย่าง

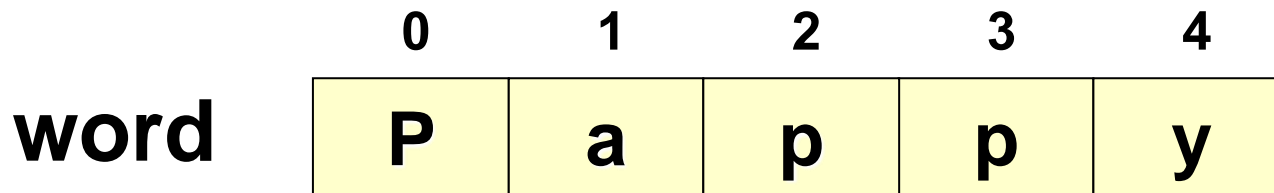
```
StringBuffer word = new StringBuffer("Number");  
word.insert(3,"operation ");  
System.out.println(word);
```

- **setCharAt(ตำแหน่งที่ต้องการ,อักขระใหม่)**

กำหนด/ใส่ตัวอักษร 1 ตัวลงไปแทนที่ตัวอักษรเดิมในข้อความโดยระบุตำแหน่งที่ต้องการ

ตัวอย่าง

```
StringBuffer word = new StringBuffer("Happy");  
word.setCharAt(0,'P');
```



word.setCharAt(0,'P');

- reverse()

จะทำการเรียงข้อความใหม่จากหลังไปหน้า

```
StringBuffer word = new StringBuffer("Number");  
System.out.println(word);  
System.out.println(word.reverse());
```

Number

rebmuN

```
----- Java run -----  
Java with love The best of times.  
.semit fo tseb ehT evol htiw avaJ
```

เนื้อหา บทที่ 4 อาร์เรย์และสตริง

✓ 4.1 อาร์เรย์

✓ 4.2 อาร์เรย์หลายมิติ

✓ 4.3 String class `concat()` `compare()` `equals()` `substring()` `charAt()` `length()` `trim()`

4.4 StringBuffer

4.5 StringTokenizer

4.6 Wrapper class

4.7 Math class

การประกาศอาร์เรย์

1. `int [] score;`
`score = new int [10];`
2. `int [] score = new int [10];`
3. `int[] score = { 1, 2, 3, 4, 5 };`

- `append` (อักขระใหม่)
- `insert` (ตำแหน่งที่ต้องการ,อักขระใหม่)
- `setCharAt`(ตำแหน่งที่ต้องการ,อักขระใหม่)
- `reverse()` เรียงข้อความใหม่จากหลังไปหน้า

โจทย์ sheet 5 ข้อ

การประกาศ String

```
String objectName = message; เช่น String name = "Wassana";
```

```
หรือ String objectName = new String(message);
```

```
เช่น String name1 = new String ("Wassana");
```

4.5 StringTokenizer class

StringTokenizer เป็น **class** หนึ่งใน **package** ชื่อ “**java.util**” ทำหน้าที่แบ่งข้อความออกเป็นส่วน ๆ ด้วย **space** **tab** **new line** ข้อความที่ถูกแบ่งเรียกว่า **token**

การประกาศว่าเป็น **object** ของ **class StringTokenizer**

```
StringTokenizer inpData = new StringTokenizer();
```

constructor

StringTokenizer(String S)

Default คือ space แต่เปลี่ยนได้

แบ่ง string ชื่อ “S” เป็นส่วน ๆ ด้วยสัญลักษณ์มาตรฐาน คือ space

StringTokenizer(String s,String delim)

แบ่ง string ชื่อ “S” เป็นส่วน ๆ ด้วยสัญลักษณ์ที่ส่งมากับ argument delim

<http://download.oracle.com/javase/6/docs/api/index.html>

method ที่เราจะทำงานกับ class StringTokenizer

boolean hasMoreTokens()

จะให้ค่าเป็น **true** ถ้ามี **token** อยู่ใน **StringTokenizer**

String nextToken()

ให้ค่า **token** ถัดไป **return String** ของ **token** ออกมา

int countTokens()

นับจำนวน **token** ทั้งหมดที่แบ่งได้ใน **StringTokenizer**

ตัวอย่าง


```
import java.util.*;
public class C4StrToken2 {
    public static void main(String[] args) {
        String word = new String("Today is Sunday);
        StringTokenizer st = new StringTokenizer(word);
        System.out.println(word);
        System.out.println("count#Token= "+st.countTokens());
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}
```

```
----- Java Run -----
Today is Sunday
Count#Token= 3
Today
is
Sunday
```

```

import java.util.*;
public class C4StrToken2    {
    public static void main(String[] args)    {
        String word = new String("Today \tis : \n Sunday a");
        StringTokenizer st = new StringTokenizer(word);
        System.out.println(word);
        System.out.println("#Token "+st.countTokens());
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}

```

```

----- Java Run -----
Today is :
Sunday a
#Token 5
Today
is
:
sunday
a
Normal Termination
Output completed (0 sec consumed).

```

```
import java.util.*;
public class C4StrToken2{
```

```
    public static void main(String[] args) {
        String word = new String("Today \tis :\n Sunday a"+
        "i love u ");
        StringTokenizer st = new StringTokenizer(word,":");
        System.out.println(word);
        System.out.println("#Token "+st.countTokens());
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}
```

```
----- Java Run -----
Today is :
Sunday ai love u
#Token 2
Today is

Sunday ai love u
Normal Termination
Output completed (0 sec consumed).
```

Sheet-deriveหา
ผลลัพธ์ 1-4

4.6 Math class → sheet formatting output

เป็น class หนึ่งใน package ชื่อ java.lang

เพื่อใช้ในการคำนวณ return **double**

รูปแบบคือ

```
ตัวแปร = Math.method(parameter_list)
```

เช่น

```
a = Math.sin(0.0);
```

<http://download.oracle.com/javase/6/docs/api/index.html>

๓๕.

```
System.out.println( "Math.abs( 23.7 ) = " + Math.abs( 23.7 ) );
System.out.println( "Math.abs( 0.0 ) = " + Math.abs( 0.0 ) );
System.out.println( "Math.abs( -23.7 ) = " + Math.abs( -23.7 ) );
System.out.println( "Math.ceil( 9.2 ) = " + Math.ceil( 9.2 ) );
System.out.println( "Math.ceil( -9.8 ) = " + Math.ceil( -9.8 ) );
System.out.println( "Math.cos( 0.0 ) = " + Math.cos( 0.0 ) );
System.out.println( "Math.exp( 1.0 ) = " + Math.exp( 1.0 ) );
System.out.println( "Math.exp( 2.0 ) = " + Math.exp( 2.0 ) );
System.out.println( "Math.floor( 9.2 ) = " + Math.floor( 9.2 ) );
System.out.println( "Math.floor( -9.8 ) = " + Math.floor( -9.8 ) );
System.out.println( "Math.log( Math.E ) = " + Math.log( Math.E ) );
System.out.println( "Math.log( Math.E * Math.E ) = " + Math.log( Math.E * Math.E ) );
System.out.println( "Math.max( 2.3, 12.7 ) = " + Math.max( 2.3, 12.7 ) );
System.out.println( "Math.max( -2.3, -12.7 ) = " + Math.max( -2.3, -12.7 ) );
System.out.println( "Math.min( 2.3, 12.7 ) = " + Math.min( 2.3, 12.7 ) );
System.out.println( "Math.min( -2.3, -12.7 ) = " + Math.min( -2.3, -12.7 ) );
System.out.println( "Math.pow( 2.0, 7.0 ) = " + Math.pow( 2.0, 7.0 ) );
System.out.println( "Math.pow( 9.0, 0.5 ) = " + Math.pow( 9.0, 0.5 ) );
System.out.println( "Math.sin( 0.0 ) = " + Math.sin( 0.0 ) );
System.out.println( "Math.sqrt( 900.0 ) = " + Math.sqrt( 900.0 ) );
System.out.println( "Math.sqrt( 9.0 ) = " + Math.sqrt( 9.0 ) );
System.out.println( "Math.tan( 0.0 ) = " + Math.tan( 0.0 ) );
```

```
----- Java Run -----  
Math.abs( 23.7 ) = 23.7  
Math.abs( 0.0 ) = 0.0  
Math.abs( -23.7 ) = 23.7  
Math.ceil( 9.2 ) = 10.0  
Math.ceil( -9.8 ) = -9.0  
Math.cos( 0.0 ) = 1.0  
Math.exp( 1.0 ) = 2.7182818284590455  
Math.exp( 2.0 ) = 7.38905609893065  
Math.floor( 9.2 ) = 9.0  
Math.floor( -9.8 ) = -10.0  
Math.log( Math.E ) = 1.0  
Math.log( Math.E * Math.E ) = 2.0  
Math.max( 2.3, 12.7 ) = 12.7  
Math.max( -2.3, -12.7 ) = -2.3  
Math.min( 2.3, 12.7 ) = 2.3  
Math.min( -2.3, -12.7 ) = -12.7  
Math.pow( 2.0, 7.0 ) = 128.0  
Math.pow( 9.0, 0.5 ) = 3.0  
Math.sin( 0.0 ) = 0.0  
Math.sqrt( 900.0 ) = 30.0  
Math.sqrt( 9.0 ) = 3.0  
Math.tan( 0.0 ) = 0.0  
Normal Termination  
Output completed (0 sec consumed).
```

```

class MathFn9 {
    public static void main( String args[] ) {
        //sin2x=2sinxcosx
        double x;
        System.out.println("sin2x=2sinxcosx");
        for (x=0.0; x<1 ; x += 0.1) {
            System.out.println(Math.sin(2.0*x)+"\t"+
                2.0 * (Math.sin(x)* Math.cos(x)));
        }
    } // end main
}

```

```

----- Java Run -----
sin2x=2sinxcosx
0.0      0.0
0.19866933079506122      0.19866933079506122
0.3894183423086505      0.38941834230865047
0.5646424733950355      0.5646424733950354
0.7173560908995228      0.7173560908995228
0.8414709848078965      0.8414709848078965
0.9320390859672263      0.9320390859672264
0.9854497299884601      0.9854497299884601
0.9995736030415052      0.9995736030415051
0.9738476308781953      0.9738476308781951
0.9092974268256818      0.9092974268256816
Normal Termination
Output completed (0 sec consumed).

```

```

public class MathFn9Format {

    public static void main( String args[] )
    {
        //sin2x=2sinxcosx
        double x;
        System.out.println("sin2x=2sinxcosx");
        for (x=0.0; x<1 ; x += 0.1)
            System.out.printf("%6.3f \t%6.3f\n" ,Math.sin(2.0*x),2*(Math.sin(x)* Math.cos(x)));
    } // end main
} // end class MathTest

```

```

----- Java run -----
sin2x=2sinxcosx
0.000    0.000
0.199    0.199
0.389    0.389
0.565    0.565
0.717    0.717
0.841    0.841
0.932    0.932
0.985    0.985
1.000    1.000
0.974    0.974
0.909    0.909

```



```

import java.text.DecimalFormat;
public class MathFnfor {
    public static void main( String args[] )
    {
        //sin2x=2sinxcosx
        double x;
        System.out.println("sin2x=2sinxcosx");
        DecimalFormat twoDigit = new DecimalFormat("0.00");
        DecimalFormat threeDigit = new DecimalFormat("0.000");
        for (x=0.0; x<1 ; x += 0.1)
        {
            System.out.println(twoDigit.format(Math.sin(2.0*x))+"\t"+
                threeDigit.format(2.0 * (Math.sin(x)* Math.cos(x))));
        }
    }
}

```

----- Java Run -----

sin2x=2sinxcosx

0.00 0.000

0.20 0.199

0.39 0.389

0.56 0.565

0.72 0.717

0.84 0.841

0.93 0.932

0.99 0.985

1.00 1.000

0.97 0.974

0.91 0.909

การสุ่มตัวเลข Random number generator

มี 2 แนวทาง

- Math class เรียกใช้ method random()
- Random class -method มี 2 method

วิธีที่ 1 Math class

รูปแบบ

```
double randomValue = Math.random();
```

method random() จะทำการ generate เลขทศนิยม ชนิด double rang from 0.0 up to, but not including 1.0

วิธีที่ 2 Random class

ตัวเลขสุ่มจาก Random class โดย Random class มี 2 method ที่เรานำมาใช้
งานได้

`nextInt(n)` ให้ค่าเลขสุ่มแบบ integer ระหว่าง 0 ถึงค่าน้อยกว่า n

`nextDouble()` ให้ค่าเลขสุ่มแบบ double ระหว่าง 0 ถึงค่าน้อยกว่า 1

```
int i;  
double value;  
for (i=1;i<6 ;i++ ) {  
    value = Math.random();  
    System.out.println("value "+ value);  
}
```

ระหว่าง 0 ถึงค่าที่น้อยกว่า 1

```
----- Java Run -----  
value 0.020286957584622067  
value 0.7249644210710521  
value 0.8173309414534058  
value 0.8781245968433625  
value 0.9754384812218874
```

```
import java.util.*;
public class RanNo9 {
    public static void main(String[] args) {
        Random ran = new Random();
        int i,value1;
        for (i=1;i<=10 ;i++ ) {
            value1 = ran.nextInt(6);
            System.out.print( value1+" ");
        }
        System.out.println();
    }
}
```

0 ถึงค่าที่น้อยกว่า n

```
----- Java Run -----
0 3 3 1 1 4 5 2 0 2
```

```
import java.util.*;
public class RanNo8 {
    public static void main(String[] args) {
        Random ran = new Random();
        int i;
        double value1;
        for (i=1;i<=10 ;i++ ) {
            value1 = ran.nextDouble();
            System.out.println(i+" "+value1);
        }
        System.out.println();
    }
}
```

ระหว่าง 0 ถึงค่าที่น้อยกว่า 1

```
----- Java Run -----
1 0.5984306375176593
2 0.8132785331665024
3 0.9275405178973355
4 0.6326592392664591
5 0.6633863829938437
6 0.9463871201937138
7 0.7229821004730123
8 0.7832807668594415
9 0.9483345225648045
10 0.2392702631403838
```

boolean
char
byte
short
int
long
float
double

4.7 Wrapper class

Class ในกลุ่ม wrapper คือ class ที่ช่วยให้การใช้งานกับตัวแปรชนิดข้อมูลพื้นฐาน (primitive)

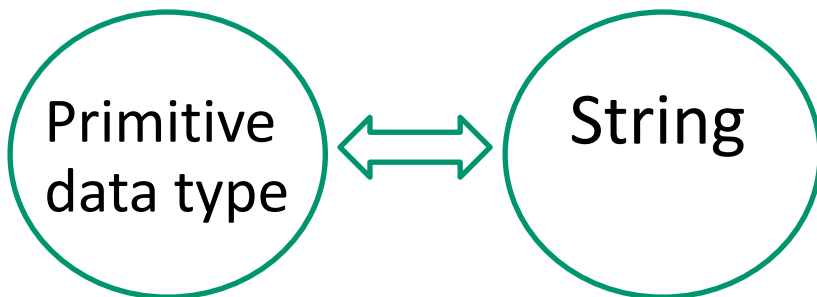
ได้สะดวก ช่วยในการแปลงข้อมูลไปอยู่ในรูปแบบที่ต้องการ ซึ่ง class ประเภท wrapper

ใช้แปลงชนิดข้อมูลพื้นฐานไปเป็นข้อความและกลับกัน

1.1 - แปลงวัตถุ string เป็นจำนวนเต็ม เรียกใช้ method `Integer.parseInt()` ✓ เรียนแล้ว

1.2 - การแปลงจำนวนเต็ม เป็นวัตถุ string เรียกใช้ method `toString()`

ใน part นี้ เพื่อให้เข้าใจง่าย เราจะเริ่มจากแปลง String เป็น int และกลับกัน int เป็น String



เราเคยแปลงแล้ว เช่น

ตัวอย่าง รับข้อมูลเข้า เป็น String

แปลง String เป็น int

```
String s = JOptionPane.showInputDialog("Enter data");
```

```
int number = Integer.parseInt(s)
```

```
// รับมาเป็น string เก็บไว้ที่ s
```

```
// แปลง string เป็น int
```

```
// number นำไปคำนวณได้
```

การทำงานดังกล่าวเราเรียกใช้ Wrapper Class

:กลุ่มของ Class ที่ช่วยแปลงชนิดข้อมูล

Wrapper class มีดังนี้

ชนิดข้อมูลพื้นฐาน	Class ในกลุ่ม wrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

จากตารางจะเห็นว่า class ในกลุ่มนี้มีชื่อที่สอดคล้องกับชนิดข้อมูลพื้นฐานเพียงแต่ Wrapper class ขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้น Character กับ Integer

การทำงานกับ primitive & Wrapper class เพื่อแปลงชนิดข้อมูล

String → primitive

1.1 แปลง **String** เป็น **จำนวนเต็ม** โดยใช้ Wrapper class Integer

```
String s = JOptionPane.showInputDialog("Enter data"); //เรียนแล้ว OK✓  
int number = Integer.parseInt(s); // นำ number ไป คำนวณได้
```

OK✓ ข้อ 5 ในsheet

primitive → String

1.2 การแปลงชนิดข้อมูลพื้นฐานประเภท int เป็นString

ตัวอย่าง

```
int number = 10;
```

```
Integer i = new Integer(number);
```

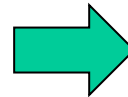
```
String s = i.toString(); // แปลงเป็น string
```

ข้อ 4 ในsheet

Sheet ตย. P.2 parseInt() toString()

String → int นำตัวเลขมาคำนวณ

int → String



```
public class C4Wrap01
{
    public static void main(String[] args)
    { // String เป็น int
        String s = "10";
        int i = Integer.parseInt(s);; // นำ i ไปคำนวณได้
        i=i*8;
        System.out.println("Hello World! "+i);
// แปลง int เป็น String
        int number =10;
        Integer j = new Integer(number);
        String st = j.toString();
        st += "love";
        System.out.println("Hello World! "+st);
    }
}
```

Hello World! 80

Hello World! 10love

นอกจากนี้ยังมี ค่าคงที่ใน **Wrapper class** ให้เราสามารถเรียกใช้งาน เพื่อดูค่าต่ำสุด/สูงสุดของชนิดข้อมูลพื้นฐาน ตัวอย่างเช่น

Integer.MIN_VALUE

Integer.MAX_VALUE

Byte.MIN_VALUE

Byte.MAX_VALUE

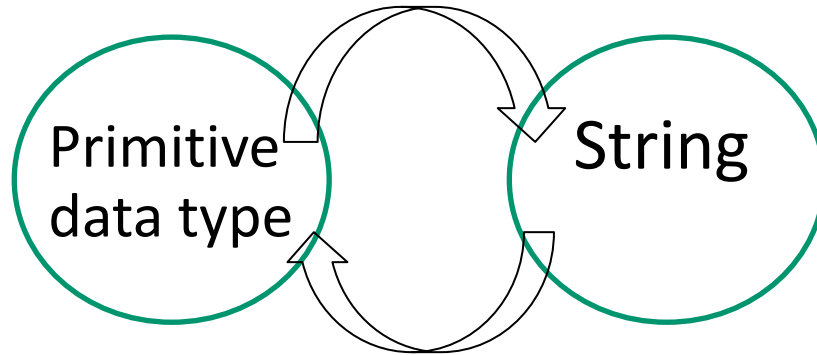
ในบทที่ 2 มีตัวอย่างโปรแกรม ที่แสดงการเรียกใช้ **max min**

OK เราเริ่มเห็นภาพ การทำงานกับ wrapper class

คราวนี้มองภาพใหญ่ขึ้น

การทำงาน Primitive data type ↔ String

สามารถเรียกใช้ method ต่างๆ ได้ดังนี้



✓ 1. แปลง primitive to String

✓ 2. แปลง String เป็น primitive

3. The Wrapper constructor

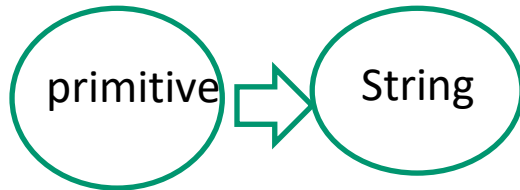
4. The valueOf() methods

5. using Wrapper conversion utilities

1. แปลง primitive to String

เพื่อแปลงค่า primitive to String เรียกใช้ **method toString()**

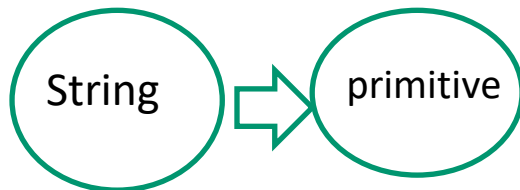
เช่น `String s2 = Integer.toString(number);` // แปลง int เป็น string



2. แปลง String เป็น primitive

เพื่อแปลงค่า String to primitive เรียกใช้ **method parseXXX(String)**

เช่น `int number = Integer.parseInt(s)` // แปลง string เป็น int



OK

3.The Wrapper constructor

String → int , ..

เมื่อสร้าง object ของ Wrapper class (keyword new) จะเรียกใช้ constructor

ตาราง Wrapper class และ constructor



Primitive ชนิด ข้อมูลพื้นฐาน	Wrapper class	Constructor arguments
boolean	Boolean	boolean or String
char	Character	char
byte	Byte	byte or String
short	Short	short or String
int	Integer	<u>int or String</u>
long	Long	long or String
float	Float	<u>float or String</u>
double	Double	double or String

Constructor (คอนสตรักเตอร์)

เมื่อสร้าง object จะเรียก
method นี้อัตโนมัติ

เปรียบเทียบกับ เมื่อเราทำงานกับ

primitive data type

เราประกาศตัวแปร พร้อม

กำหนดค่าเริ่มต้น

เช่น int i=0;

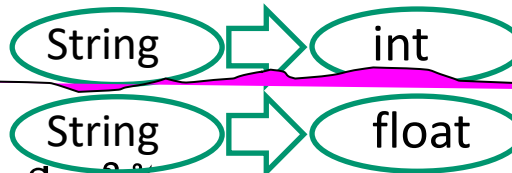
ตัวอย่าง

Integer i1 = new Integer(42);

Integer i2 = new Integer("42");

Float f1 = new Float(3.14f);

Float f2 = new Float("3.14f");



ex

จากตัวอย่าง primitive int float เรียกใช้ Wrapper class Integer Float ตามลำดับ โดย
Constructor argument เป็น int or String

ตัวอย่าง การแปลง String เป็นชนิดข้อมูลพื้นฐานประเภท int

```
String s = JOptionPane.showInputDialog("Enter data"); // รับมาเป็น string เก็บไว้ที่ s  
Integer i = new Integer(s); //แปลง String เป็น int i นำไปคำนวณได้
```

```
String s = new String("123");  
Integer i = new Integer(s);  
System.out.println(i*2);
```

4. The valueOf() methods

Wrapper class ที่เป็นตัวเลขจะมี method valueOf() ใช้ **แปลงค่า String เป็นตัวเลข**

เช่น

```
Integer i1 = Integer.valueOf("43");
```

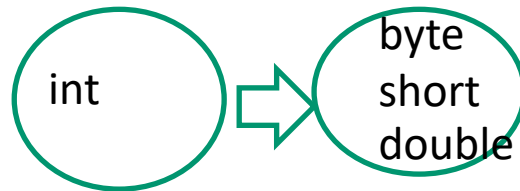
```
Float f1 = Float.valueOf("3.14f");
```



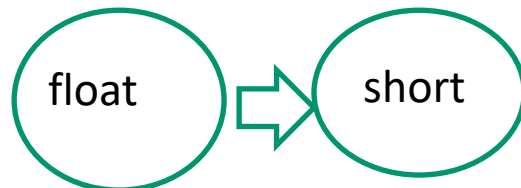
5.using Wrapper conversion utilities

Wrapper class ที่เป็นตัวเลขจะมี method สำหรับแปลงค่าเป็น primitive โดยเรียกใช้ method [primitive type]XXXValue() ซึ่งชื่อของ method จะเปลี่ยนไปตามค่า primitive ที่ return เช่น

```
Integer i1 = new Integer(42); // เริ่มต้น
byte b = i1.byteValue(); // convert i1 integer to primitive : byte
short s = i1.shortValue(); // convert i1 integer to primitive : short
double d = i1.doubleValue(); // convert i1 integer to primitive : double
```



```
Float f1 = new Float(3.94f); // เริ่มต้น
short s = f1.shortValue(); //convert f1 to primitive : short and value s
// is truncated that is 3
```



เนื้อหา บทที่ 4 อาร์เรย์และสตริง

4.1 อาร์เรย์

4.2 อาร์เรย์หลายมิติ

4.3 String class

4.4 StringBuffer

4.5 StringTokenizer

4.6 Wrapper class

4.7 Math class

การบ้าน บทที่ 4

1. เขียนโปรแกรมเล่นเกม Eggy-Peggy

เขียนโปรแกรมรับข้อความแล้วแสดงผลลัพธ์จนกว่าผู้ใช้จะพิมพ์คำว่า "END"

วิธีการเล่นคือ จะแทรกคำว่า egg ที่หน้าสระ เช่น หากผู้ใช้ใส่ข้อความ

Twinkle twinkle little star

จะได้ผลลัพธ์ดังนี้

Twegginklegge twegginklegge leggittlegge steggar

เช่น I love you

eggI leggovegge yeggoeggu

2. สร้างรหัสพนักงานคนต่อไป

กำหนดให้ รหัสพนักงานคนสุดท้ายที่ระบบเก็บไว้คือ EM01227

ให้เขียนโปรแกรมเพื่อสร้าง ผลที่ได้คือ รหัสพนักงาน EM01228

Hint: ศึกษาโปรแกรมหน้าถัดไป

หาตัวเลขใน String เราต้องการนำตัวเลขไป manipulate คือนำตัวเลขไปคำนวณได้

```
public class C4Str200{
    public static void main(String[] args) {
        String word = new String("test(123)");
        String fdata="";
        char te;
        int i;
        for (i=0;i<word.length() ;i++ ) {
            te = word.charAt(i);
            if (Character.isDigit(te)) {
                fdata+=te;
                System.out.println(te+" ");
            }
        }
        System.out.println(fdata);
        int number = Integer.parseInt(fdata);
        System.out.println(20*number);
    }
}
```

```
----- Java run -----
1
2
3
123
2460
```