

BCM2301 :



เกณฑ์ให้คะแนน

โครงสร้างข้อมูลและอัลกอริธึม (Data Structures & Algorithms)

Course Outline



ตำรา

1. Introduction to Data Structures
2. Type of Data Structure
3. Physical Data Structure
4. Logical Data Structure
5. Stack
6. Queue
7. Linked List
8. Tree
9. Midterm



Course Outline



ตำรา

Binary Tree

Binary Search Tree

Searching(Sequential, Binary)

Searching(Hashing, Analysis)

Sorting(Bubble, Selection, Insertion)

Sorting(Shell, Merge, Quick)

Data Structures Application

Final



การให้คะแนน

- สอบกลางภาค 30 %
- สอบปลายภาค 30 %
- รายงาน 20 %
- ทดสอบย่อย 10 %
- การมีส่วนร่วม 10 %

คะแนน เกรด

86 - 100

A

82 - 85

A-

78 - 81

B+

74 - 77

B

70 - 73

B-

66 - 69

C+

62 - 65

C

58 - 61

C-

54 - 57

D+

50 - 53

D

46 - 49

D-
MÖNK

0 - 45

F
LITTLE DOG

Introduction to Data Structures

- ความหมายของ ข้อมูล และ โครงสร้างข้อมูล
- ชนิดของโครงสร้างข้อมูล
- ประโยชน์ของโครงสร้างข้อมูล



Introduction to Data Structures

- ความหมายของ ข้อมูล และ โครงสร้าง
- ชนิดของโครงสร้างข้อมูล
- ประโยชน์ของโครงสร้างข้อมูล

Integer
Array
Stack
Queue
Linked List
Tree



ข้อมูล (Data)

ข้อเท็จจริง หรือสิ่งที่ถือหรือยอมรับว่าเป็นข้อเท็จจริง สำหรับใช้เป็นหลัก
อนุมานหาความจริงหรือการคำนวณ

- ราชบัณฑิตยสถาน. (2546). พจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ. 2542. กรุงเทพฯ: นานมีบุ๊คส์
พับลิเคชั่น.

Data is a set of values of qualitative or quantitative variables.

Data are individual pieces of information.

Data in computing (or data processing) are represented in a structure that is often tabular, a tree, or a graph.

Data are typically the results of measurements and can be visualised using graphs or images.

- Wikipedia The Free Encyclopedia. (2014). Article. Retrieved January 10, 2014, from <http://en.wikipedia.org/wiki/Data>

ข้อมูล (Data)

ข้อเท็จจริง(Fact) เกี่ยวกับสิ่งใดสิ่งหนึ่ง ที่ได้มาจากการวัดหรือ
ประเมินด้วยวิธีต่างๆ ซึ่งอาจมีค่าในลักษณะเชิงปริมาณ
(Quantitative) หรือเชิงคุณภาพ(Quanlitative) ก็ได้



โครงสร้างข้อมูล (Data Structure)

Data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently.

- Wikipedia The Free Encyclopedia. (2014). Article. Retrieved January 10, 2014, from <http://en.wikipedia.org/wiki/Data>

Data structure, Way in which data are stored for efficient search and retrieval.

- Encyclopedia Britannica. (2014). Data structure. Retrieved January 10, 2014, from <http://global.britannica.com/EBchecked/topic/152190/data-structure>

โครงสร้างข้อมูล (Data Structure)

การดำเนินการจัดกลุ่มและจัดรูปแบบข้อมูล เพื่อให้สามารถจัดเก็บ
ปรับปรุงแก้ไข ค้นหาและนำข้อมูลที่ต้องการออกมาใช้งานได้
อย่างมีประสิทธิภาพ

Type of Data Structures

- โครงสร้างข้อมูลทางกายภาพ (Physical Data Structure)

มีรูปแบบการจัดระเบียบข้อมูลและการดำเนินการต่างๆตามลักษณะโครงสร้างทางกายภาพพื้นฐานของระบบคอมพิวเตอร์ โดยมุ่งเน้นในเรื่องประสิทธิภาพ (efficiently) และประสิทธิผล (effectively) ในการทำงานเป็นหลัก

- โครงสร้างข้อมูลทางตรรกะ (Logical Data Structure)

มีรูปแบบการจัดระเบียบข้อมูลและการดำเนินการต่างๆตามมุมมองของผู้ใช้เป็นหลัก



โครงสร้างข้อมูลทางกายภาพ (Physical Data Structure)

- **Primitive Data Types** (รูปแบบข้อมูลพื้นฐาน)

เป็นรูปแบบการดำเนินการของข้อมูลหน่วยย่อยพื้นฐาน ตามลักษณะโครงสร้างข้อมูลทางกายภาพของระบบคอมพิวเตอร์

- **Byte, Short, Integer, Long** : เลขจำนวนเต็ม
- **Float, Double** : เลขทศนิยม
- **Char** : ตัวอักษร
- **Boolean** : ค่าตรรกะ

- **Structure Data Types** (รูปแบบข้อมูลมีโครงสร้าง)

เป็นรูปแบบการดำเนินการจัดกลุ่มของข้อมูลหน่วยย่อยพื้นฐานที่มีความเกี่ยวข้องกันเข้าไว้ด้วยกัน

- **Array** : ชุดข้อมูล(แถวลำดับข้อมูล)
- **Record** : ระเบียบข้อมูล
- **File** : แฟ้มข้อมูล

โครงสร้างข้อมูลทางตรรกะ (Logical Data Structure)

- **Linear Data Structure** (โครงสร้างข้อมูลแบบเชิงเส้น)

มีลักษณะการจัดรูปแบบข้อมูลที่เรียงต่อกันเป็นแนวเส้น

- **Stack** : กองซ้อน
- **Queue** : แถวคอย
- **Linked-List** : รายการโยง

- **Non-Linear Data Structure** (โครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น)

มีลักษณะการจัดรูปแบบข้อมูลที่ไม่เรียงต่อกันเป็นแนวเส้น

- **Tree**: ต้นไม้
- **Graph**: กราฟ



ชนิดของข้อมูลใน JAVA



ชนิด	ขนาด	ช่วงค่าข้อมูล Java	ช่วงค่าข้อมูล Visual Basic
byte	8-bit	-128 ถึง 127	Byte : 0 ถึง 255, SByte : -128 ถึง 127
short	16-bit	-32768 ถึง 32767	Short : -32768 ถึง 32767
int	32-bit	-2^{31} ถึง $2^{31}-1$	Integer : -2,147,483,648 through 2,147,483,647
long	64-bit	-2^{63} ถึง $2^{63}-1$	Long : -9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
float	32-bit		Single : -3.4028235E+38 ถึง -1.401298E-45 (negative) 1.401298E-45 ถึง 3.4028235E+38 (positive)
double	64-bit		Double : -1.79769313486231570E+308 ถึง -4.94065645841246544E-324 (negative) : 4.94065645841246544E-324 ถึง 1.79769313486231570E+308 (positive)
boolean	ขึ้นอยู่กับ platform	true หรือ false	Boolean : true หรือ false
char	16-bit	'\u0000' ถึง '\uffff'	Char : 0 through 65535 (unsigned)

Array

- ความหมายของ Array
- ประเภทของ Array
 - Static Array
 - Dynamic Array
- ตัวอย่างการประยุกต์ใช้ Array



ความหมายของ Array

Array เป็นโครงสร้างข้อมูล ที่มีลักษณะการจัดเก็บข้อมูลชนิด(Type) เดียวกันเป็น แถวเรียง(linear) ต่อเนื่องกันไป และการเข้าถึงข้อมูล แต่ละตัวสามารถกระทำได้โดยระบุหมายเลขดัชนี(Index) ซึ่งเป็น หมายเลขบอกตำแหน่งของข้อมูลนั้นๆ

Index	1	2	3	4	5
Name	สมชาย	มาลี	สามารถ	วิชาญ	วันเพ็ญ
Score	85	62	77	38	59



ประเภทของ Array

Static Array (Array ที่มีขนาดคงที่)

- มีการกำหนดขนาดตั้งแต่ตอนที่สร้าง Array ขึ้นมา

Dynamic Array (Array ที่มีขนาดไม่คงที่)

- ไม่มีการกำหนดขนาดในตอนที่ยังสร้าง Array
- ขนาดของ จะแปรเปลี่ยนไปตามจำนวนของข้อมูลที่ถูกเก็บ



ประเภทของ Array

Static Array (Array ที่มีขนาดคงที่)

- มีการกำหนดขนาดตั้งแต่ตอนที่สร้าง Array ขึ้นมา

คำสั่งสร้าง Static Array

Type [][]... Obj-name = new Type [n₁][n₂].....;

อ้างถึงข้อมูล : Obj-name [i₁][i₂].....;

ขนาดของ Array : Obj-name.length

ตัวอย่าง :

```
int[] a = new int[5];  
float[][] money = new float[5][3];  
money[1][2] = 12.50;  
float x = money[2][1];  
int s = a.length;
```

ที่ถูกเก็บ



ประเภทของ Array

Static Array (Array ที่มีขนาดคงที่)

- มีการกำหนดขนาดตั้งแต่ตอนที่สร้าง Array ขึ้นมา

Dynamic Array (Array ที่มีขนาดไม่คงที่)

- ไม่มีการกำหนดขนาดในตอนที่ยังสร้าง Array
- ขนาดของ จะแปรเปลี่ยนไปตามจำนวนของข้อมูลที่ถูกเก็บ



คำสั่งสร้าง **Dynamic Array**

```
ArrayList<Type> Obj-name = new ArrayList<Type>();
```

เพิ่มข้อมูล : `Obj-name.add(data);`

อ่านข้อมูล : `Obj-name.get(index);`

ขนาดของ **Array**: `Obj-name.size()`

***หมายเหตุ ต้องมีคำสั่ง `import java.util.ArrayList;` ที่ส่วนเริ่มต้นของไฟล์โปรแกรมด้วย

ตัวอย่าง :

```
ArrayList<integer> a = new ArrayList<integer>();  
a.add(7*5);  
int dt = a.get(3)  
int s = a.size();
```

Dynamic Array (Array ที่มีขนาดไม่คงที่)

- ไม่มีการกำหนดขนาดในตอนการสร้าง **Array**
- ขนาดของ จะแปรเปลี่ยนไปตามจำนวนของข้อมูลที่ถูกเก็บ



ตัวอย่างการประยุกต์ใช้ Array

คำสั่ง **for**

```
for (Var-name = init; end-condition; inc-exp)  
{  
    Command Statements;  
}
```

ตัวอย่าง :

```
int i;  
int[] a = new int[10];  
for(i=1;i<10;i++)  
{  
    a[i] = 2*i;  
    System.out.println("Welcome to Thailand " + a[i]);  
}
```



ความหมายของ Record

Record เป็นโครงสร้างข้อมูล ที่มีลักษณะการจัดเก็บข้อมูลเกี่ยวกับสิ่งใดสิ่งหนึ่งที่เราสนใจ โดยเป็นข้อมูลชนิด(**Type**) เดียวกันหรือต่างชนิดก็ได้ และโดยทั่วไปมักมีการจัดเก็บระเบียบหลายๆระเบียบไว้ด้วยกันเป็น ไฟล์(**File**)

ชื่อ	คะแนน	เกรด
สมชาย ใจดีมาก	85	A



ความหมายของ File



File เป็นโครงสร้างข้อมูล ที่มีลักษณะการจัดเก็บระเบียบข้อมูลหลายระเบียบไว้ด้วยกัน เพื่อเป็นการรวบรวมข้อมูลเกี่ยวกับเรื่องที่เราสนใจไว้ด้วยกัน

การเข้าถึงข้อมูลแต่ละระเบียบ สามารถทำได้หลายวิธี ทั้งวิธีการใช้ค่าดัชนี (**Index**) หรือไม่ใช้ก็ได้ขึ้นอยู่กับโครงสร้างการทำงานที่ถูกกำหนดไว้ของไฟล์นั้นๆ

Record



ชื่อ	คะแนน	เกรด
สมชาย ใจดีมาก	85	A
มาลี มีมานะ	65	C
สามารถ ประสาทดี	77	B

การประยุกต์ใช้งาน โครงสร้างข้อมูล

- การจัดการข้อความในลักษณะต่างๆ เช่น การตัดข้อความ, การก๊อปปี้ข้อความ
- การดำเนินการ **Undo/Redo** ใน **Microsoft Office**
- การแก้ปัญหาต่างๆที่มีความซับซ้อน เช่น การหาทางออกจากเขาวงกต
- การประมวลผลนิพจน์คณิตศาสตร์
- การตรวจสอบความถูกต้องของภาษา

