

Programming

프로그래밍。โปรแกรมมิ่ง。

Advanced
ก้าวสูง(ก้าว)

Java...



JAVA
ภาษา

SCCS 209 : ADVANCED PROGRAMMING

กรุ๊ป: ๓๖

「Ta」
Sec. 1 5188018

Aj. Ananta Srisuphab

Quizz & Assignments 15%

Midterm 35%

Final 50%

Program IDE : EditPlus

ขั้นตอน

compile ➤ Menu text: Compile Java ~~หรือ install~~

Command : ... \ Java\ jdk1.6.0_03\ bin\ javac.exe

Argument : \$(FileName)

Initial directory: \$(FileDir)

* Capture output ออก
กันที่หน้าจอในรูปแบบรัน Dos

Run ➤ Menu text: Run Java

Command : ... \ Java\ jdk1.6.0_03\ bin\ java.exe

Argument : \$(FileNameNo Ext)

Initial directory: \$(FileDir)

ดาวน์โหลด JDK (Java Development Kit) ที่ <http://java.sun.com>

แนะนำ IDE สำหรับ Java ได้แก่ NetBeans และ Eclipse ก็ได้

Note: ชื่อเดิมของ Java คือ Oak แต่ลักษณะเปลี่ยนไปเป็น Java

เปลี่ยนมาเป็น Java ชื่อเดิมซึ่งอยู่ในภาษาโปรแกรม

* แนะนำ Eclipse มากกว่า เนื่องจากเจ้า Eclipse ไม่ต้อง install แต่ต้องเพียงแค่ดาวน์โหลดแล้วใช้งานได้เลย แต่ NetBeans ใช้งานได้ก็ต่อเมื่อต้องติดตั้ง Java บนเครื่อง

Java Platform

↳ แบบ platform คือ ความสามารถรองรับต่างๆ

J2SE

standard Edition

ใช้สำหรับโปรแกรมทั่วไป เช่น Java application

X. เริ่มต้นต้องติดตั้ง

J2EE

Enterprise Edition

ใช้สำหรับการสร้างเว็บไซต์ เช่น applet

Multitiered (?)

ใช้กับองค์กรใหญ่

J2ME

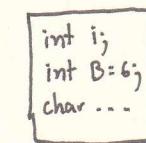
Micro Edition

ใช้สำหรับการสร้างนิสัย กับ อยู่ในเครื่องโทรศัพท์มือถือ

ข้อดีของ Java

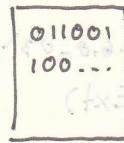
- ใช้ง่าย syntax เหมือน C
- ใช้หน่วยเดียว รันได้บนทุกรายงาน (มีจุด : ... นำเข้าอีเมล์ jvm ก็รันได้)
- ตัวที่รักษา (ตัว memory) อัตโนมัติ
- ลีบ class ได้ช่วยลด:
- ฟรี ! !!

การ Compile



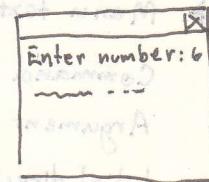
javac →
Compiler

ไฟล์ .java
Source Code



java →
Interpreter

ไฟล์ .class
Byte Code



เรียบร้อย!

ซึ่งไม่ใช่ compiler
(ไม่สามารถตัดสินใจ)

java virtual Machine (JVM)
(ทั่วไปเรียกว่า OS, รันโปรแกรม file.class
แบบ file.class: นอร์มัล)

Java

```

1 class Test
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello World!");
6     }
7 }

```

T2/5/11 กรณีจัดตั้งเดิม

- 1) Java จะมองว่า ทุกอย่างที่เราเขียนเป็น class, ชื่อ Test (เราจะ save หรือ save ให้เป็นไฟล์ กับชื่อที่ได้ไว้ตามต้อง .java)
- 3) method ชื่อ main (เหมือน func. void main()... ใน C)
- 5) method "println" ใจไว้แล้ว หัดความบันหนัน ๆ

println(); => printf("\n"); ← มีไส้ในตัวชี้ไปยังบรรทัดใหม่ต่อไปนี้
 print(); => printf(); // ไม่ต้องมีข้อความที่ต้องพิมพ์

↑ In C คือ function ของมาจากการ print line

Java จะมองว่า method "println" ถูกเก็บอยู่ใน class System จะเรียกมันใช้ต้องเลือกชื่อหน้า (คล้ายๆ รึว่า struct ใน C ที่มีชื่อ ".") ตัวนั้น

Class Pro Java จะชี้ชัดเจนมาก ! (ลองรัน Edit Plus ดูจะพบว่าต้องเลือกจากในหน้า class นั้นๆ ก่อนจะเขียนได้)



Primitive Data Type

↳ In Java มีตัวไปร์ 2 แบบ คือ Primitive และ reference

ลูงเขยฯ นั่นก็ แล้ว กดอย่างนี้ ↑
สีฟ้าเงินอ:<--> ตัวไปร์ชนิดพื้นฐาน
ใช้อิงก้าหนังรักๆ

↑ ตัวไปร์ primitive]

- boolean* เก็บตัว true, false เท่านั้น ← ลูงเขยฯ เรื่อง flag ที่ 0. เชบวันนี้
- char* เก็บ 'อักษร' 1 อัก
- byte -128 ถึง 127 (ไม่ถึง 128 เพราะมี 0 อีกต่อ)
- short -32768 ถึง 32767
- int* 1byte ! ← เก็บได้ 1byte กว่า int ใน C
- long มากกว่า !!
- float กรณีบลล.
- double* กรณีบลล. กว่า float หรือใน java ไม่สามารถ double ไม่เขียนต่อ

* การ declaration ตัวไปร์
กำหนดชื่อใน C เท่านั้น!

ไม่ต้องรังสรรค์ พวກ byte short long , float
ต้องจาก default ของ Java จะมองว่า อั้นๆ เดิมๆ ว่า เป็น int (และ double)

byte a = 5; float b = 5.6;

out error! 7 ต่อ เพราะ 5 ถูกมองเป็น int 101.2 ถูก byte ไม่ได้
5.6 ถูกมองเป็น double 9 ถูก float ไม่ได้

* ต้อง compile
จะวอร์นิ่งตัวเดียว!

byte a1 = 6b;

↑ กรณีตัว byte 9 ถูก 6 บวกตัวเลข, long 9 ถูก 1 , float 9 ถูก f

ต้องเปลี่ยนให้ได้

byte a1 = 6b;

byte a2 = 3b;

byte a3 = a1 + a2;

} หมายเหตุ ก็ยัง error!

แล้ว + เล็กๆ 1: ถูกมองเป็น int ด้วย!

ทางแก้

byte a3 = (byte)(a1 + a2);

+ , - , * , / , % และ อั่งๆ เนื่องใน C

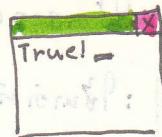
ยกเว้น % ที่ mod กรณีบลล. ที่ 4.7 % 2 จะได้ 0.7 !

if-else, switch, Loop

เนื้อง C ทุกอย่าง! จะเพิ่มมาต่อ boolean และจะเข้าเรื่องนี้ใน

เช่น boolean A = true;

```
if( A ) System.out.println("True!");
else System.out.println("False!");
```



ค่าเดือนใน if เป็น true บันทึกใน เรื่องนี้

* ปกติ พวงเรื่องในของ if (รวม while ก็อยู่) จะเซ็ตตัวเป็น true หรือ false ก่อนอยู่แล้ว

int B = 5;

if(B < 6) ~~~

→ $5 < 6 \rightarrow$ จริง = true ที่ไหน เรื่องนี้ใน if

String

* String ใน Java ไม่ใช่คลาส เป็น Array of char แล้ว
* String สร้างด้วย S หรือชี้ทุกครั้งแรก: เป็น class

String ถูกหั่นเป็น Class หนึ่ง von Java แบบที่ primitive data type (เช่น int หรือ double)

▶ การสร้าง ตัวแปร ให้เป็น String (ต้องจากมาจาก class ต้องไปรับที่คลาสไว้ จึงถูกเรียกว่า

"Object" หมายความว่า)

String str = new String("janvac~");
 หรือ String str2 = "something like that!";

→ Object str หรือ str2 ถูกสร้างโดยมีความซับรวมกัน

String ทุกอย่าง

* การแปลง String อาจดูเหมือน ประกาศข้อผิดพลาดที่นี่ดูแล้ว แต่ลับๆ ใจว่า!

กรณี int a; กำหนด a = 5, 5 ถูกเก็บที่ a

กรณี String b; กำหนด b = "กบขค..." ถูกเก็บที่ในช่องที่ดูแล้วใน memory

ส่วนที่ ๒ จะเก็บแต่ว่า ได้ตัวผู้พูดวันนี้อยู่ที่ไหน

" อะไรแล้ว จะใช้มัน? ... รู้สังข์ บันทึก: จำแล้วจะสร้าง String รึเปล่า "

- 6 -

String (ภาษาอังกฤษ)

↑ แต่ตัว height ห้ามเรียก

method ที่ใช้กับ String

→ บันทึกหาน: เนื้อหา: กดพิเศษหน้าจอกล่อง (เราก็ต้อง)

length()

: ฟังก์ชันตรวจสอบว่า String (เมื่อ int strlen ของ C) return int

charAt()

: ใช้หาว่าตัวอักษรใดในตำแหน่งนั้น (เรียกว่าหนึ่งจุดตัวอักษร หรือ character) ดังนี้ A[0] = A[15]; = A.charAt(15); return char

indexOf()

: ฟังก์ชัน search หรือตัวอักษรใน String เช่น หาตัว e จากคำว่า "Sec.1"

วิธี: return 1 ก็จะมีในนี้ ✖ ถ้าไม่มีจะ return -1 มาก

0	1	2	3	4
S	e	c	.	1

return int

substring()

↓ method ที่ใช้

หัวใจ

: ใช้ตัด String ให้ได้ต้องการ อย่างที่ 2 นี้

เช่น ให้ String test = "ABCDEFG"

0	1	2	3	4	5	6
A	B	C	D	E	F	G

- test.substring(1, 3);

- test.substring(2, 5);

↑ จุดเริ่ม
↑ หยุด!

(ต้องไปหัวเรื่อง
ตัวอักษร)

BC

CDE

หรือ

- test.substring(5);

- test.substring(3);

↑ จุดเริ่ม
(ต้องไปหัวเรื่อง
ตัวอักษรที่หาง)

FG

DEFG

return String

equals()

: ใช้เปรียบเทียบว่า String 2 ตัวเท่ากันหรือไม่ สิ่งที่ return คือบวก

return boolean

* ไส้รวมๆ กด String 2 ตัวมา A == B ไม่ใช่ Object ของ String

ใช้ reference ต้องเน้นอย่างเดียว ไม่ใช่ที่ 1 ตัวซึ่งเป็นตัวเดียว ก็พอ

จะ ? .. รันมายังไง

compareTo()

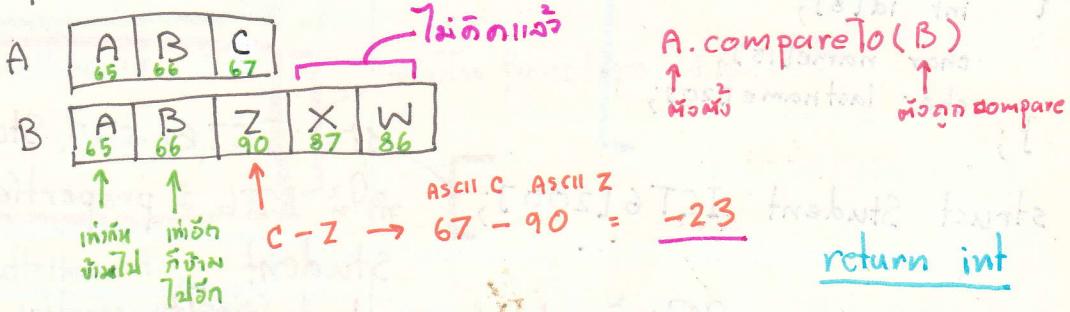
เช่น

String A = "ABC";
String B = "ABZXW";

A.compareTo(B);

-23

▶ compareTo คือ หาตัวชี้รวมไปเรียงพื้นที่ของตัว (ตัวอักษร ASCII)



concat()

เช่น

String name = "Steve";
String lastName = "Smith";
String fullName = name.concat(lastName);

Steve Smith

* concat ไม่ดองได้ใช่เมื่อ: ใน Java 1.01 String + String 7 ตัว

เช่น

fullName = name + lastName;

ดำเนินการ

เช่น

fullName = name + " " + lastName;

Steve Smith,

เช่น

name += lastName;

note a += b ดั่ง a = a + b

เช่น int a = 8; a += 3; a จะมีค่า = 11

return

toUpperCase() toLowerCase()

: ใช้เปลี่ยน String ให้กับรายชื่อตัวใหญ่ทั้งหมด (หรืออักษรทั้งหมด)

replace()

: ใช้เปลี่ยนตัวอักษร (นั่งปู: โน๊ก) ให้ String เป็นตัวอักษร

เช่น String A = "abcdefghijklm";

System.out.println(A.replace("cddefg", "#%*@"));

ab#%*@hijk

ส่วนที่ต้องการเปลี่ยน

ส่วนการเปลี่ยนเป็น
อะไรก็ได้ทัน

クラス

Class

Review struct vs C

```
struct Student
{
    int id;
    char name[15];
    char lastname[20];
};
```

```
struct Student ICT6[200];
```

สร้าง struct ชื่อ Student ที่มี properties 3 อย่างคือ id, name, lastname

สร้าง ICT6 ทาง Student
ทำให้ ICT6 มี properties หนึ่งตัว叫做 student

* ICT6 สามารถเข้าถึง id, name, lastname ได้

In Java สร้างการเขียนโปรแกรม

```
class Student
{
    int id;
    String name;
    String lastname;
}
```

ภายใน main

```
Student ICT6 = new Student();
ICT6.id = 5188018;
ICT6.name = "Ta";
```

สร้าง class ชื่อ Student
มี properties 3 อย่าง

สร้าง Object ชื่อ ICT6 ทาง
class Student

*** ก็จะเป็นแบบนี้ได้ ไม่ได้ใช้ Var เท่าเดิมกัน!

ส่วนมากสิ่งที่เขียนใน class ลักษณะ method (คือ function หรือ
จะสั่งให้ object ทำงานให้ต่อไปยัง)

ສ່ວນປະກອບນັກຂອງ Class

class & subclass

ເພີ້ມເຊື້ອດ້ວຍ ທົກໄຟກູ

```
{
    constructor (...) {
        statement 1;
    }
}

method 1 (...) {
    statement;
}

attribute 1;
    (ນີ້ແມ່ນ instance variable)
}
```

constructor ດັ່ງນັ້ນກ່າວນັດຕໍ່ເຮັດວຽກ
ກ່າວໃຫຍ່ object (ໄລຍະກົດ)

ເມີນ function ສູນເຮັດວຽກ

attribute ດັ່ງນັ້ນເປັນ variable

ກ່າວໃຫຍ່ class
note style o. ອົບນົມ ອະລິດ attribute ໄວດ້ວຍກູ
ແກ່ຮ່ວມໃຈກູ ແນວດກູ (ກ່າວ -
constructor)

Class & Object = new Class();

ນັ້ນຈາກທີ່ເຮົານີ້ກ່າວ object ແລ້ວ ເຮົາ ສູນ ໃນ object ລວມທີ່ກໍ່ມີ method ອີເໂວີໃດທີ່ສູນ class
ເຊັ່ນ ສູນ object A ຈາກ class Rectangle ... ກ່າວ A.getArea() ໂດຍ

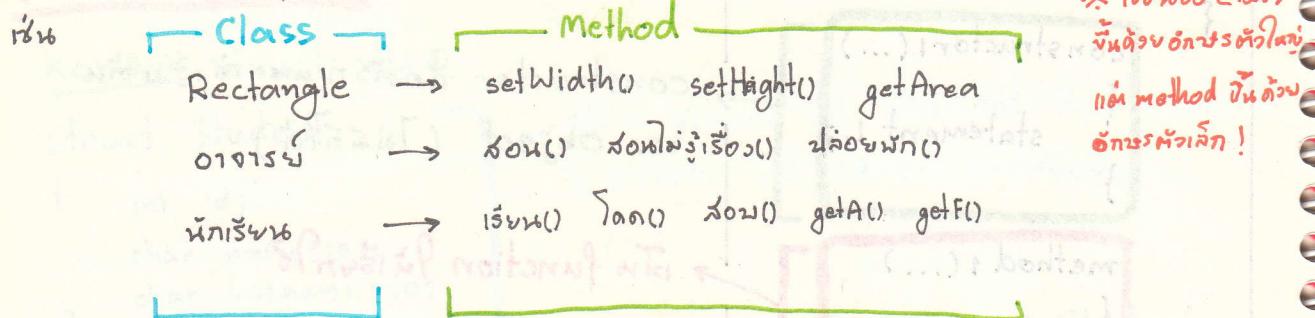


ເມີນ method ສູນ class Rectangle

(ຈະເຮັດວຽກໃຫຍ່ກ່າວໃຫຍ່ class Rectangle ມີວິທີ
ທີ່ສູນ)

Method... メットド

↳ = function, อยู่ใน object ที่สร้างขึ้นมา ทำให้ใช้งานง่าย



Ex. เมื่อยก class Rectangle รับความกว้าง + ยาว จะได้ด้านล่างนี้

ผู้ Class Rectangle

```
Class Rectangle
{
    public void setWidth(double a)
    {
        width = a;
    }

    public void setHeight(double b)
    {
        height = b;
    }

    public double getArea()
    {
        return width * height;
    }

    private double width;
    private double height;
}
```

① method setWidth เป็นการ set กว้าง

รุ่งใน object (ในห้อง A), ใช้แบบ void
เพรา: เนื่องจากเก็บค่าต่องเดียว ไม่ต้องส่งกลับ
รับค่า a จะได้ width = a

② เมื่อหันเข้ามา จะเป็นการ set กว้างรูปแบบ

③ เป็น method double, เมื่อรันเร็ยก็:
return ค่า Area หรือว่าได้ (ได้ของ parameter)
เพรา: เอามาสั่ง ก็จะมีอย่างเดียว)

④ declare ต้องไป width กับ height อยู่ใน Class เป็นตัวแปรที่
ต้อง method ต่างๆ เรียกใช้

ผู้ Main

```
Class ...
{
    public static void main(String[] args)
    {
        Rectangle A = new Rectangle();
        A.setWidth(10); ②
        A.setHeight(5.5); ③
        System.out.println(A.getArea()); ④
    }
}
```

ก็ต้องระบุตัว
แบบ .0 ไม่
เพรา: return ค่า
double

55.0

① สร้าง Object ชื่อ A ชนชั้น Class
Rectangle

② กันหนักความกว้างให้ A = 10

③ กันหนักความสูงให้ A = 5.5

④ บันทึกให้ A (กันรัน ออกจาก class
แล้ว return กลับมา)

หัวห้องใน()

แบบ a กับ b ที่ ① กับ ②

* ข้อ 11 ปกติการห้อง public กับ private (แนวร่องรอยของ class ④)

public : ใช้มีมต้องการให้ class อื่นเรียกใช้ได้ แบบ main เรียกใช้
A.setWidth(10) 75

private : ส่วนใจใน class อื่นห้าม 접근ได้ ก็ซึ่ง, private ที่มี type double ชื่อ width กับ height เพื่อใจใน class main สามารถเข้าถึงได้โดยตรง
ก็ไม่ใช่ private 75, บน main สามารถเข้าถึง A.width = 10 75
||| กรณีไร้การ: การห้องใจในเวลาสั่ง set ก็จะผ่าน method setWidth() อย่างเดียว

Note!

▶ method ก็จะชื่อเช่นเดียวกับ get...() สำหรับการห้องครัวดัง

```
public int get...() ← ห้องการ return 0; ให้ก่อนไปใน main ที่ type ชื่อตามนี้ □
{
    return ...;
}
▶ ก้านนี้ set...() ดังนี้
public void set... (int a) ← เมื่อ void แล้ว method ก็จะ set ให้ 0 ช่วงเดียว ให้ห้องการ
{
    A = a;
    ...
}
```

A เป็นตัวแปรห้องที่ declare (ห้องตัวแปร) ชื่อตามชื่อ class
* เมื่อ a 75 ที่ห้องต้องช่วยห้อง A แล้ว A 75 ที่ห้อง method
ต่อไป

↑ ห้องในช่วง (..)

Overloaded Method

↳ ดังการห้อง method ก็จะชื่อเหมือนกัน แต่จะ parameter ต่างกัน

เช่น...

(เปลี่ยนชื่อจาก class Rec. ให้เป็น rectangle)

```
public double getArea()
{
    return width * height;
}          10 * 5.5
over loaded public double getArea (double a, double b)
{
    width = a; 3
    height = b; 4
    return width * height;
}          3 * 4
```

In main ก็จะ... แบบนี้

Rectangle A = new Rectangle();
A.setWidth(10);
A.setHeight(5.5);

System.out.println(A.getArea());
System.out.println(A.getArea(3, 4));

55.0
12.0
↑ parameter
↓ value

* เวลาเรียกใช้ method บันทึกห้อง เหตุการณ์ห้องในบันทึกห้องต่อ

→ overloaded ใช้กับ constructor ได้ด้วย

→ ใช้ overloaded method บันทึกห้อง แต่ต้องมี parameter ต่างกันหมด

Constructor

コンストラクター

↳ เป็น method พิเศษที่มีชื่อเท่ากับชื่อ class 1 ตัว ก่อนการเรียกใช้งาน ระบุ

ต่อจาก Rectangle ต่อไปนี้:

เพิ่ม constructor ใหม่ๆ **constructor**

class Rectangle

```
{
    public Rectangle(double a, double b)
    {
        width = a;
        height = b;
    }
}
```

แบบ type ในหน้า!

10.0 5.5

width = a;

height = b;

5.5

Main

Rectangle A = new Rectangle();

A.setWidth(10);

A.setHeight(5.5);

System.out.println(A.getArea());

55.0

ผลลัพธ์
เป็น

Rectangle B = new Rectangle(10, 5.5);

System.out.println(B.getArea());

55.0

* หมายเหตุ คำว่า Rectangle ที่ไม่ construct ต้อง set ให้ก่อนทั้ง width และ height ของ constructor ที่ set ให้ใน main การตั้งค่า

* เหตุผล: set ใน class constructor แล้ว จะ set ให้กันทั้งหมด

Rectangle C = new Rectangle(1, 2);
C.setWidth(4);

width 1	height 2
4	2

หากใน class ไม่มี constructor อุปกรณ์ compiler จะใช้ default constructor ที่มีอยู่

โดย default คือ ไม่ระบุ value ให้ เน้น ค่าเริ่มต้น หรือตั้งค่าเริ่มต้น

* สำคัญมาก ! : ก้ามการสร้าง constructor ใหม่เพิ่มเติม, compiler จะใช้ default ที่มีอยู่ ไม่ระบุ value ให้ เน้น ค่าเริ่มต้น หรือตั้งค่าเริ่มต้น

เช่น ใน Class Rectangle ของการสร้าง constructor นาต้องรับค่า double 2 ตัว จึงสร้าง object ของ Rectangle ต้องระบุ value ที่ต้องตั้งค่า

Rectangle A = new Rectangle();

หมายเหตุ: ถ้า object ไม่ระบุ value ต้องรับค่า value ที่ต้องตั้งค่า ไม่ใช่ overloaded constructor

Class Rectangle

```
{
    public Rectangle()
    {
    }
```

ก้ามตัวตั้งค่าด้วย

```
{
    width = 0;
    height = 0;
}
```

public Rectangle(double a, double b)

```
{
    width = a;
    height = b;
}
```

default constructor

ที่บันทึกไว้

ที่ต้อง main ที่สร้าง object และ set ให้ก่อน หรือ set ก็ต้องมี value!

Attribute នៃទីតាំង (instance variable) ឬឯឈាម

→ ត្រូវបានក្លាយជាទីតាំងនៃ class

ឬគាត់ចុចិត្តនៃ main នៅពេលរំភោះទីតាំង declare (ស្រាវជ្រាវប័ណ្ណរាក់អី) នៃ class ក្នុងអ៊ីនីនីហើយ នៅពេលរំភោះការខ្លួន (ស្រាវជ្រាវនៃ method នេះ)

► ឬគាត់ស្រាវជ្រាវ private ដែលអាចធានាបានពេរខ្លួន (ទីនេះ តើអាចកិត្តក្នុងក្រុមហ៊ុន Java) នៅពេលការបង្កើតរាយការណ៍

► ក្នុងការរំភោះទីតាំង នឹងបានបញ្ជាក់ថា អំពីរួមចំណាំនៃ method នៅនេះ (method ដែលមានរូបរាង public ដែលនឹង class ដែលត្រូវកិត្តក្នុង)

► ដើម្បីចិត្តក្នុង ពេលរំភោះទីតាំង parameter នៃ method នៃ class ទាំងអស់ នឹងចុចិត្តនៃ public នៃទីតាំង (method ដែលត្រូវកិត្តក្នុង)

(ទីនេះ តើអាចធានាបានពេរខ្លួនចំណាំនៃ method នេះ)

ឧបាឌ

តាមលក្ខាត់:

```
private int count;
public void setCount(int count)
{
    count = count;
}
```

ឱកាស:
↓

```
private int count;
public void setCount(int a)
{
    count = a;
}
```

នឹងការបង្កើតរាយការណ៍
មានការបង្កើតរាយការណ៍

```
private int count;
public void setCount(int count)
{
    this.count = count;
}
```

គោលគម្រោងតួន:

គោលគម្រោងនៃ count នឹងការបង្កើតរាយការណ៍
នឹងការបង្កើតរាយការណ៍ នៅពេលរំភោះទីតាំង
នឹងការបង្កើតរាយការណ៍

.. និងស្រាវជ្រាវ count = count; ទីនេះតើ
នឹងការបង្កើតរាយការណ៍
.. count ក្នុងរាយការណ៍ នឹងត្រូវបានកិត្តក្នុង

នឹងការបង្កើតរាយការណ៍
ពេរខ្លួនដែលបានបង្កើតរាយការណ៍

* នឹងការបង្កើតរាយការណ៍ នឹងការបង្កើតរាយការណ៍
នឹងការបង្កើតរាយការណ៍ នឹងការបង្កើតរាយការណ៍
(តើ a នៅពេលរំភោះទីតាំង នឹងការបង្កើតរាយការណ៍
នឹងការបង្កើតរាយការណ៍ នឹងការបង្កើតរាយការណ៍)

តារាងនៃ លើកទី១ ពេលរំភោះទីតាំង

ពេលរំភោះទីតាំង នឹងការបង្កើតរាយការណ៍
នឹងការបង្កើតរាយការណ៍ នឹងការបង្កើតរាយការណ៍

Scanner スケーナー

→ เป็นชั้นที่ให้วิธีรับ input จาก user

ต้อง `import java.util.Scanner;`

- สืរวว Object ของ class Scanner โดยมี parameter คือ System.in

`Scanner in = new Scanner(System.in);`

ชื่อตัวแปรที่ใช้บันทึกได้

→ จากนั้นจะรับค่า 0: ให้เกิด object ที่สร้างไป เช่น ก็อปปี้ไปรับค่า

`System.out.print("Enter the number: ");`

`int x = in.nextInt();`

method ที่ได้จาก Scanner

► `int A = in.nextInt();` - ใช้รับค่า int เท่านั้น

► `double B = in.nextDouble();` - ใช้รับค่า double เท่านั้น } ถ้า เข้าพวกรหัส อักซ์ ก็ error!

► `String C = in.next();` หรือ `in.nextLine();`

↳ ตัดคำที่

"Space"

(ออกช่องว่างบวกกับตัดเลข)

↳ ตัดคำที่การ Enter

* 例: เข้าค่า 12345 → เลข 123

`int number = Integer.parseInt("0183...");`

ระวัง! ตัวอักษร เลข 0

หรือ 0x (ฐานนี้เป็น 16)

มีหน่วยอยู่ว่า เม็ดกิโล

ฐาน 8 กับ 16

Random ランドム

เมื่อ 2 วิธี - ใช้จาก class Random

- ใช้จาก Math.random();

ใช้จาก class Random.

`import java.util.Random;`

- สืรัว Object แล้ว ตัว

`Random r = new Random();`

- random เป็น interface

`int x = r.nextInt(10);`

ยกตัว int คือ เลข 0-9

(รวม 10 ตัว)

- ตัวราก เลข ให้ช่วง a ถึง b ก็...

ใช้จาก Math.random();

- ไม่ต้อง import

- เลขที่ random 00 กว่า ชาติ

ตัวที่ 0 ถึง <1

(ตัว 0.00 ถึง 0.9999... 0:)

ข้อเก็บจดก็จำ!
ข้อมูลตัดสูตร
จะไม่ทำงานแล้ว

► `int x = r.nextInt(b-a+1) + a;`

Part 6-1 Array

0 จุดเริ่ม... ก็ถือ 0 จุดเริ่ม + 0 จุดเริ่ม 0:15

รูปแบบการตั้งแปงปกติ
 ↳ int x;

รูปแบบการตั้งแปง Array
 ↳ int[] x;

↑ เช่นเดียวกันที่ต้อง Array นั่นเอง!

ส่วนต่อไป...

int[] x = new int[3];
 x[0] = 1;
 x[1] = 10;
 x[2] = 100;

มัด

เก็บ

↑

↑ ตัวอย่างของ

int[] x = new int[5]; 1-D

int[][] y = new int[5][3]; 2-D

นี้คือ...



↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

↑

▶ 101 class น่าจะร่างเป็น Array ที่ดี...

```
String[] list = new String[10];
String[] list2 = {"ชื่อใบเบี้ยง", "ชื่อใบเตา", "หัวใจผู้มี"};
```

Ex. 9: sum ผลรวมทั้งหมดของ Array นั้น

```
int[] x = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int sum = 0;
for(int i = 0; i < x.length; i++)
{
    sum += x[i];
}
System.out.println(sum);
```

↑ length ต้องห้าม
↑ ความยาวของ Array

(55)

↑ แปลง成繩子的總和

```
for(int y : x)
{
    sum += y;
}
```

↑ สำหรับตัวที่จะไปห้าม
↑ แปลง成繩子的總和
↑ ไม่ใช่ 10 ค่า แต่ใช้ 9

↑ * การเขียนแบบนี้เรียกว่า

for each Loop

Sort

การเรียงซึ่งกันใน Array วิธีหนึ่งๆ ต้อง เรียก method ชื่อ sort นั่นเอง

- import java.util.Arrays;

int[] num = {10, 9, 1, 7, 5, 3, 2};

String[] name = {"Ta", "Book", "Yoke", "Lee", "Than", "Kwan", "Friend", "Sherbet", "Gaikun"};

Arrays.sort(num);] method นี้ทำให้เรียงตัวเลขแล้ว!

Arrays.sort(name);

for(int A : num)

System.out.print(A);

1 2 3 5 7 9 10

for(String B : name)

System.out.print(B);

Book Friend Kwan Lee Sherbet Ta Than Yoke Gaikun

(ต้องลบ 2)

length & Index レンチとインデックス

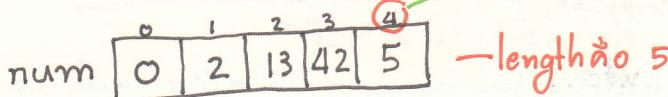
length - คือจำนวนของ Array

- สำหรับ for Loop ส่วนเงื่อนไข

```
int[] num = {0, 2, 13, 42, 5};
for(i=0; i<num.length; i++)
    System.out.println(num[i]);
```

* length คือ () น.

ส่วน () ที่อยู่ในรูปตรวจสอบความถูกต้องของ String



index - บอกตำแหน่งของที่อยู่ใน Array

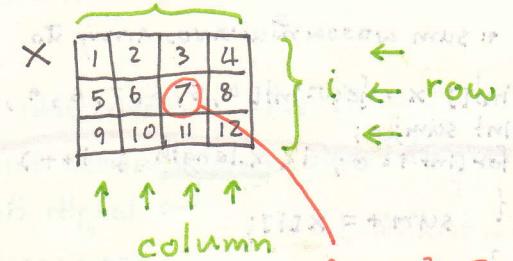
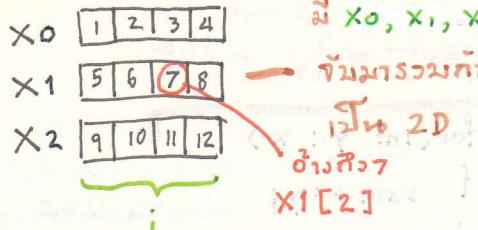
- เริ่มที่ [0] เป็น 0

- index ต้องสุดท้ายของ Array ดีอ [Array.length - 1]

Array 2D & 3D アーリー二、三 ダイメンション

2D

- Array 2 มิติ ก็คือ Array สองมิติ คล้ายๆ ผ้าม่านตั้งกัน



- ส่วนมากถ้าเขียน 1D ใช้วิธี Loop ที่ใช้เพียง i เท่านั้นจะไปวน

- ถ้าเขียน 2D ที่ใช้ i กับ j โดย i อยู่loopนอก, j อยู่loopใน

(วนloopตามแนวราบก่อน แล้วค่อยทำแนวตั้ง)

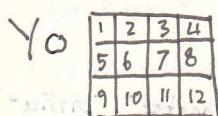
ต้องการเข้าถึง 7

ต้องหันด้าน X[1][2]

จาก X1 ก้าวเดียว
เป็น X[1]
แล้วเดิน
ก้าวเดียว
ก็เป็น!

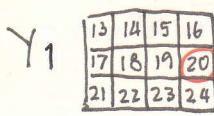
3D

- ก็คือ 3D Array 2D มาซ้อนกันหลาย ๆ แผ่น



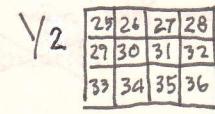
มี Y0, Y1, Y2

เริ่มต้น 3 มิติ



ด้วยตัว 20

Y1[1][3]



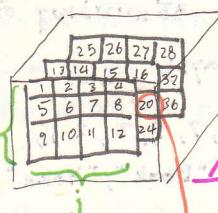
- วนloop โดย

for (k=0; ...)

for (i=0; ...)

for (j=0; ...)

ที่ให้มาไว้จะไป...



k → ดิ่งลงมา ก็คือ

แนวนอนขึ้นไป 2D ก็จะมาซ้อนกัน

ชั้นที่ 20

Y[1][1][3]

อยู่แผนที่ 1 (ชั้นที่ 2 แผนที่ 2
แท้จริงเริ่มที่ 0)

ArrayList

มันก็คือ ArrayList 1D แต่แหน่ง: แล้วถูกทำให้รองรับรายการชื่อ

- ขนาดไม่จำกัด จะใช้ไปกี่ตัวก็ได้
- ครบ + ไม่ครบ ข้อมูลยังคง (มีสิ่งที่?)

* หมายความว่า ของใช้หัวน้ำดูแล array ธรรมชาติ บันทึกการทํางาน

- ฯลฯ! ปัจจุบันจะใช้ (ส่วนบนนั้นที่ไม่ติด array อุบัติ) ฯลฯ

* ต้อง import java.util.ArrayList;

การประยุกต์ใช้งาน

สร้าง ArrayList ธรรมชาติ

String[] A = new String[3]

รูปแบบของที่ต้องเก็บ

สร้าง ArrayList

ArrayList<String> B = new ArrayList<String>()

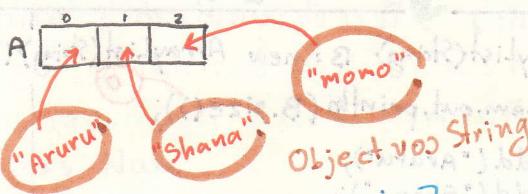
มี: รายการที่เก็บ
(ไม่ต้องนองกันมาก)
(จะใช้ class ก็ได้)

[add]

A[0] = "Aruru";
A[1] = "Shana";
A[2] = "momo";

B.add("Aruru");
B.add("Shana");
B.add("momo");

↑ กับส่วน add จะเพิ่มข้อมูล (object)



Array ธรรมชาติของ A ต้องเก็บชื่อ
ที่มีตัวอักษรตัวต่อๆ กัน

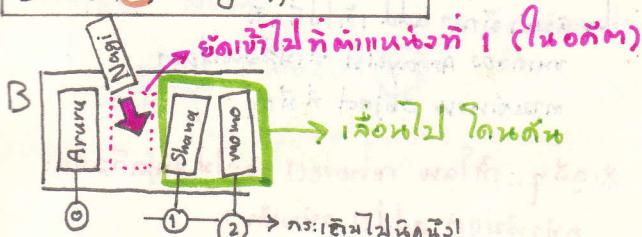
Step1 เมื่อบอกว่า สร้าง B เป็น ArrayList
ก็เหมือนสร้างห้องเก็บของ ชื่อ B ที่

น่องที่ในห้องเก็บไว้ไม่วันนนน!
(หรือหากว่า mem ต้อง)

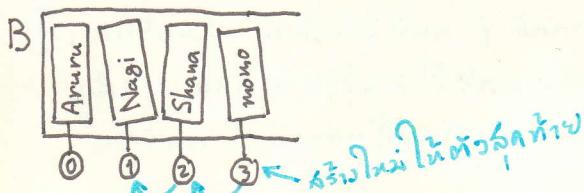
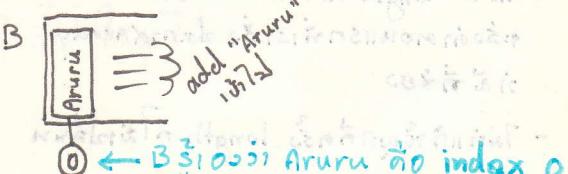
ก้าวขา เกร็งข้อมูลเข้าไปที่ต้องห้องนั้น ก็ใช้ add ให้ เช่นกัน เช่นเดียวกัน

B.add(0, "Nagi");

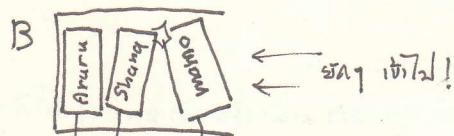
Step2 add Object ให้ไป (ไม่ต้องบอก index B จะต้อง
กันครั้งนั้นที่ใช้เท่ากับ 0)



จากนั้น B จะเปลี่ยนอันดับ index โดยอัตโนมัติ



สร้างใหม่ให้ตัวลูกห่วย

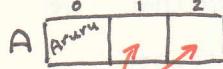


จะต้อง add เอาไปเพื่อ 4: เรียงตัวอักษร
กลับไป

การลบข้อมูล, ตัวที่ n

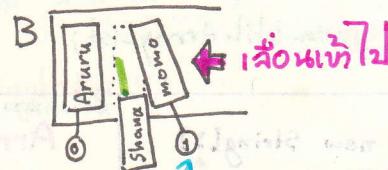
```
A[1] = null;
A[2] = null;
```

```
B.remove(1);
B.remove("momo");
```

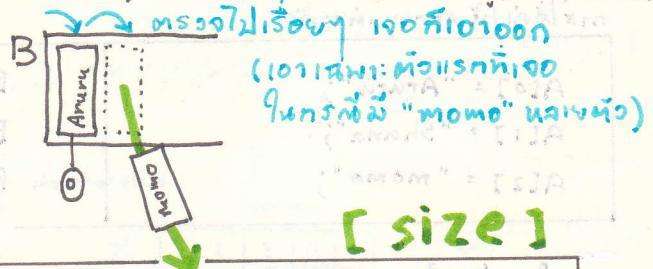


ข้อมูลนี้ไป แต่ช่องเก็บของบังอยู่

B.remove(1);
↓ ลบ object ตัวที่สองที่ 1 ทิ้งไป



B.remove("momo");
↓ ลบ object ตัวที่สองที่ 2 ทิ้งไป (ถ้าไม่มีมันอยู่แล้ว ก็ไม่ลบอะไร)



[size]

รัตตภานิยาน Array

```
String[] A = new String[3];
System.out.println(A.length); ③
A[0] = "Aruru";
A[1] = "Shana";
A[2] = "momo"; ③
System.out.println(A.length);
```

- การใช้ length ตรวจสอบความยาว
- สร้างตัวแอบนี้ไว้ก่อนที่เราสร้าง ปลากลาง Array ไว้แล้ว เช่น
- ไม่ว่า แก้ไขข้อมูลกี่ครั้ง length ก็จะไม่เปลี่ยน

```
ArrayList<String> B = new ArrayList<String>(); ①
System.out.println(B.size()); ②
```

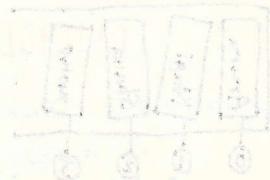
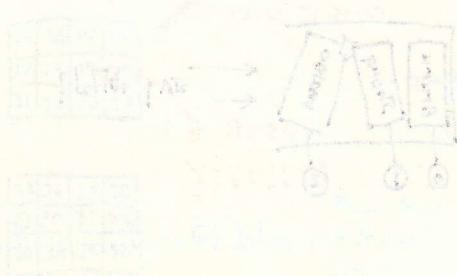
```
B.add("Aruru");
B.add("Shana");
B.add("momo"); ③
```

```
System.out.println(B.size());
```

- size() ← ผู้เชี่ยวชาญ method!

↓ ผู้เชี่ยวชาญ add เท่านี้ ก็ได้แล้ว
ขนาดของ ArrayList จะเปลี่ยนแปลง
ตามจำนวน Object ที่มีอยู่

* ดูดู... สำหรับ remove() ยังไง ขนาดก็ลดลง:
จะซ่อนแบบ add() อย่างเดียว



[get]

การดึงข้อมูล 00 กว่าๆ

```
for(int i=0; i < A.length; i++)
    System.out.println(A[i]);
```

Aruru
Shana
momo

```
for(int i=0; i < B.size(); i++)
    System.out.println(B.get(i));
```

Aruru
Shana
momo

- ใช้ คำสั่ง .get() ในการดึงข้อมูล 00 กว่าๆ

(index 0 ถึง size นับ คำสั่ง add หรือ remove object 00 กว่าๆ)

ArrayList ของ class

ArrayList สามารถ เก็บ Object หลายๆ หัว ของ class ได้ (Array ธรรมดายังทำได้ ไม่สามารถ)

ArrayList<ชื่อ class> ชื่อArrayList = new ArrayList<ชื่อ class>();

- ส่วนมากใช้ในโจทย์ มี ArrayList อยู่ หนึ่งใน <> ว่า class ชื่อ นี้มีอะไร ?

ก็จะมี ต้องรู้ 10 ว่า เขาจะให้เราสร้าง class ยังไง ขึ้นมา 10 บ้าง

บ่งชี้วิธี ()

Ex. ลืม class ชื่อ Num, มี constructor รับ parameter 1 ตัว เป็น int มี method 1 ตัว 10 บ้าง
return คุณที่รับมา กับคุณที่รับไป (กันหลังๆ ชื่อ getNum2)

- ▷ ให้สร้าง main โดยมี ArrayList ชื่อ list, add Num เท่าไหร่ อย่างน้อย 3 ตัว แล้ว print 00 กว่าๆ เป็น
- ค่าตาม
 1. ต้องใช้ int และ บวก หรือ ? (ก้าว ... ขึ้นส่วนยังไง กว่า เนี่ย + ก้าว Eng.)
 2. เพิ่นโจทย์ แล้ว สวีต Class Num ต้องมี ?

Step1 เรียนรู้การสร้าง class ชื่อ Num เราต้องทำอะไร

```
class Num { } // ต้องมี }
{
```

Step2 โจทย์กำหนด ลืม constructor 1 ตัว รับ parameter 1 ตัว เป็น int ก็ได้

(* constructor ลืมอะไร ? parameter ลืมอะไร ? กับไม่อ่านเรื่อง class ตอนนี้ (p.9-13))

```
class Num
{
    public Num(int a) { } // ชื่อ ตัวแปร 0 ตัว ก็ได้ 9 ตัว 7 ตัว ก็ได้
    { } // ปะกับ public เลยตอนนี้
}
```

!
เวลาเขียนสิ่ง ไม่ต้องรีบเขียน } ชี้ก้าวจัด เราก็ได้ไว้ไว้ จะเขียนได้เพิ่ม ต้องจบที่รัก
แล้ว ก็ จัดกัน จะ เว้นที่ไว้ ซึ่งช่วยกัน 2 บรรทัด เพื่อ เตือนว่า เนื่อง บันทึกไว้ด้วย
ก็กด Enter ที่หนึ่งบรรทัด ใหม่ว่างๆ

— 20 —

step 3 ส่อง method ชื่อ getNum2 return เลขที่รับมา กับลั่งของกลับไป

```

class Num
{
    public Num(int a)
    {
    }

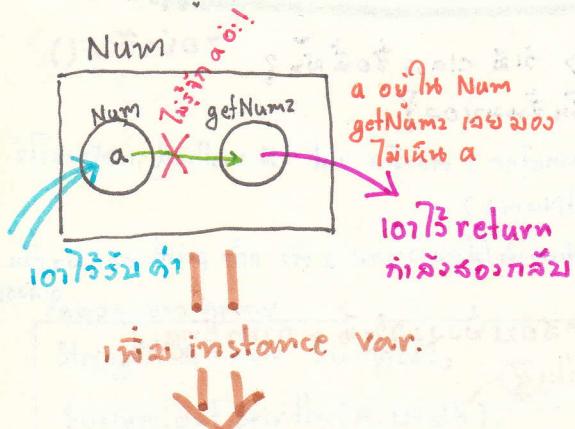
    public int getNum2()
    {
        return a*a; // ลั่งของ a^2
    }
}

```

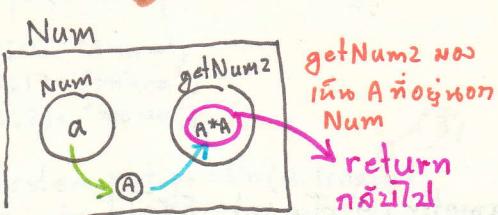
↑ ก็หนอดินี return ก็จะรับเป็น int แล้ว แต่ต้อง

step 4 ส่อง instance variable อย่าง เมนต์ต่อไปนี้ แรกกันนะว่าง method

ก้าวตามรูป ตอนนี้เราเข้า Num ไปแล้ว นี่



- ตัวที่รับมา กูกากิม เป็น a เท่ากับนั้น เองก็คือ การ return ตัว a*a ก็คือกลับไปก็จะ
- ปัญหานี้ดัง : method ที่ทำการ return ตัว getNum2 มันดันไม่รู้จัก a ที่สร้างใน Num
- อธิบาย : ส่วนตัวไปรบกวนตัว นอก method ที่รับ 1 อย่างเดียว เช่น ตัวสุ่มค่า a ที่ใน getNum2



- ส่วน A (ให้มี int ก็ได้) ซึ่งเท่ากับ a แล้ว
- getNum2 ค่าทางตัว a*A กลับสุ่งกลับ

```

class Num
{
    public Num(int a)
    {
        A=a; ← จง A เท่ากับ a
    }

    public int getNum2()
    {
        return A*A; // ลั่งของ A*A
    }
}

private int A;

```

↑ return A*A กับลั่งไป

* private int A
ทุกตัวที่เป็น instance var. ต้องมีค่า
เป็น private เพื่อไม่ให้ใน class อื่น สังตัวแปล
ผูกเข้าไปใช้

!* สัม private = โคนหักดิ้น

↳ class สร้างแล้ว ทั้งช้า สัมภัย
รุ่งไว้ จะ !

Step 5 ขั้นตอน ArrayList ที่ใช้งาน = สร้าง ArrayList จาก Num แบบ Test ชุด:

class NumArrayList ← ใจไปตั้งชื่อว่า หนึ่งชื่อไร ก็หึ่งๆ ก็ได้: (หนึ่งกันนะอย)

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

→ สิ่งที่นี่สร้างชื่อ list ก็จะรักษาตาม ไม่ต้องดูมาก

ArrayList<Num> list = new ArrayList<Num>();

:

Step 6 มองใน add Num เป็นไปอย่างนี้อย่าง 3 ตัว ก็ add เต็มไป

- มี 2 วิธี 1. สร้าง object จาก Num สาม 3 ตัว แล้วดับเบิลเพิ่ม ArrayList
2. สร้าง object มองที่บัดเด็ดเพิ่ม ArrayList เลย

สร้าง Object ก่อน

Num n1 = new Num(1);

Num n2 = new Num(2);

Num n3 = new Num(3);

list.add(n1);

list.add(n2);

list.add(n3);

สร้าง Object มองชุดแล้ว list เก็บ

list.add(new Num(1));

list.add(new Num(2));

list.add(new Num(3));

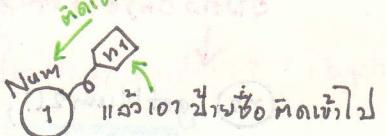
หากเรื่องนี้!

เวลาเราสร้าง Object เราจะ สร้างแบบ class ชื่อ = new Class(); โดยไม่รู้ว่า ทำในส่วนไหน ก็ได้ การสั่ง Num n1 = new Num(1);

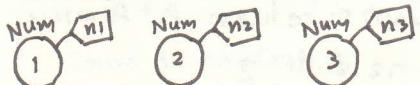
เมื่อสั่ง Num n1 ตัวนี้ก็จะรับ Num ที่เราสร้างมา

เมื่อสั่ง new Num(1); ตัวนี้ก็จะการสร้าง Object มาให้กับนี่ ก็จะนั่ง

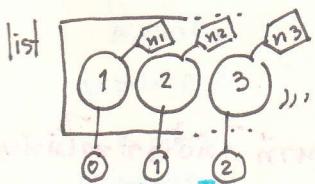
จะการนำการเรื่อง เชกุล่าบันต่อไปได้!! ↗



1. สร้าง Object ที่นี่แล้วก่อน



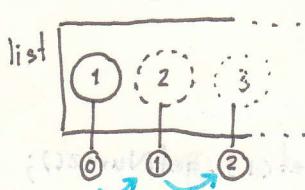
2. ใจไปพากเนย ใจบัดเด็ดกล่องชื่อ list



index ถูกตัด ใจโดยรับคืนผลลัพธ์

โดยไม่รู้ว่า Object ที่เข้ามาเป็น ชื่อชื่อติดอยู่แล้วหรือไม่

1. บัดเด็ดพร้อมๆ กัน สร้าง Object



รูป ก็จะ Object ใจล่วง แอบมายังช่องด้วย
แล้ว list ใจรัน... พากนี้เนื้อง Num นี่
เข้ามาตามลำดับด้วยกัน index นี่ชัด!

* ทั้ง 2 วิธี ใจได้เหมือนกัน (ก็ใจไทยไม่รู้จะ)
แต่ถูกกิน Object แล้วชื่อ ชื่อ ชื่อ จ: (index แล้ว
แอบมาอีกแล้ว!)

step 7 print test ណີເຫດ

: ດັວກກຳນົດກີ່ວວ

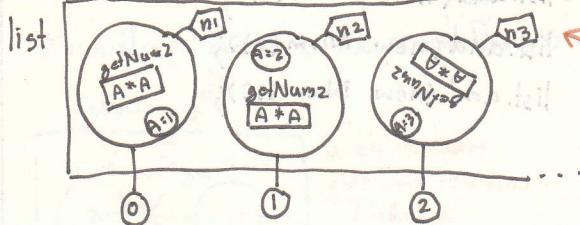
```
for (int i=0; i < list.size(); i++) {
    System.out.println(list.get(i).getNum2());
}
```

ນຽມທົດນີ້ຄົງເບື້ງໃຈ (ຢືນຢັນ?)
ນຽມທົດນີ້ແວ່ນ? (ວວ)

ນຽມການປັດເຕີກໃນດຽວດ້ວຍ
ນຽມການປັດເຕີກໃນດຽວດ້ວຍ

ກອນນີ້ເຮັດ Object 3 ຕົວໃນ list ໃຫ້: ພົມເລີ່ມ method ຂໍຂອງ getNum2 ອີ່ໆ ທີ່

(ອີ່ໆ ຖຽນໃຫ້ເຮັດຂອງສັນ)



ນິ້າຂໍ້ອນນີ້ໄມ້ມີ ຜົນມັນ
ໃນວາເຮັດກີ່ສ່ວນພານ index

list.get(i).getNum2();

ສັງເກົ່າ object ກົດນີ້ ສະໜັບກີ່ get() ສັນຈະນີ້ມີກີ່ກົດນີ້ n2

n2.getNum2();

n2 ມີກີ່ Num ຂຶ້ມີ getNum2()

ເລື່ອງໂດຍເຮັດກີ່ return A*A ນີ້
ສ່ວນ A ນີ້ n2 ພົດ 2

∴ ສຸດ list.get(1).getNum2(); ພົດ = $2 * 2 = 4$

ຈະແລ້ວ

ກຳເຫັນ ຂໍຂໍມູນນີ້ ຄວາມໃນໄດ້ໃຈ 5 ນາທີ ພ່າຍມາກ 10 ນາທີ ໄມນັ້ນກາ: ທີ່ໄວ້ກັ້ນ

Static

— ទំនើង ឱ្យ ឈុំ ចាប់បូក (រាយការ method) តិចជាដី ឬ ឈុំកសិតម្រោគ

↳ ការសែវា variable ឬ method ដើម្បី static ដែលវានាមួយគា

"សម្រាកទិន្នន័យឱ្យធូលិកអាជីវកម្មសរុប Object (Lai kou new)"

* ជាពីរការ នៃការសែវា class កិច្ចរាយ ← (ហើរការណ៍)

A ក្នុងទីនេះ ឱ្យបានចែងចាយនៃរាយ method

— ចាប់បូក static កិច្ចរាយ A នៃវានិតិចែងរាយនៃ class នេះ

Ex. ឯកសារ Class ឱ្យ ឱ្យ

```
Class Count ឱ្យបានចែងចាយ
{
    public void plus()
    {
        count++;
    }
    public int getCount()
    {
        return count;
    }
    private int count=0;
}
```

ការសែវា count ឱ្យជា static

```
Class Count ឱ្យជា static
{
    public void plus()
    {
        count++;
    }
    public int getCount()
    {
        return count;
    }
    private static int count=0;
}
```

method តិចជាដី ឬ ឱ្យជាដី !

static ទៅស្ម័គសារ ឬ private / public នៃការនៅ final

↳ ការសែវា private static final void ចំណូនឲ្យ ;
(ឬ public) ឱ្យជាដី

ចុចិត្ត
ចំណូនឲ្យ

* code ខាងក្រោម នឹងតារាង print A,B នូវ System.out.println(A.getCount() + ", " + B.getCount());

សរុប class test

class Count-test

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

ឯកសារ

```
{
    Count A = new Count();
    Count B = new Count();
```

ឱ្យបានចែងចាយ

① print A,B

A.plus();

② print A,B

B.plus();

③ print A,B

A.plus();
B.plus();

④ print A,B;

ឱ្យជាដី

① 0,0

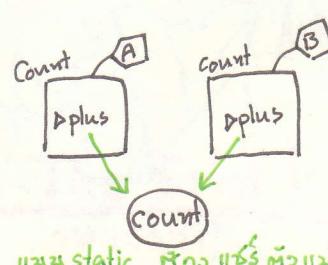
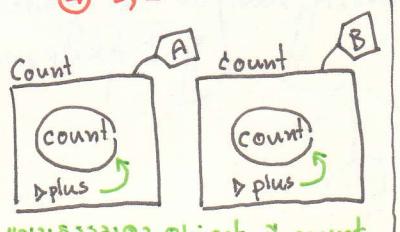
② 1,0

③ 1,1

④ 2,2 ← plus រាយ A

⑤ 2,2 ← plus រាយ B

⑥ 4,4 ← plus រាយ A+B



}

ឱ្យបានចែងចាយ ឬ count ឱ្យជាដី

ឱ្យជាដី

ឱ្យជាដី ឬ ឱ្យជាដី ឬ ឱ្យជាដី

ឱ្យជាដី

static ก็จะอนุญาตให้ใช้ในตัวแปร

- public static void main
- ของ class Math เช่น Math.sqrt() Math.round() Math.PI
- Arrays.sort(); (เรียงลำดับกิจกรรม)

1. main เป็น static ไม่ต้อง param: คลาสต้องอยู่ใน class มาร์กชั่น Object.

ใช้ร่องกิจ main เดียว (มี static method)

2. Math ใช้ class ไม่ใช้ร่องกิจ method: บวกต้องใช้ต้อง import Math มาร์กชั่น
Object ก่อน ส่วน Math.PI เป็นการขอค่าคงที่ของ PI (ที่มี method แต่เป็น static variable ... ใน class Math คือ การประกาศ)

public static final double PI = 3.14...

variable ห้ามก่อสร้าง public เพื่อให้ class อ่านร่องกิจ
ใช้คงที่นี้ได้

3. ใน method อยู่ร่อง array ใช้สั่ง array เท่าไปรับผลได้เลย

สงสัยกันวันนี้?

ลองมาดูทางนี้กันดูอีก
100=๑๙

comment: ลืม... รู้บ้าง เมื่อกำกับแล้ว
ผู้อ่านรู้ว่าต้องร้องทุกๆ ข้อความ



สรุป code ที่ผ่านมา

- ชื่อ class จะเป็นต้องตัวใหญ่ , method จะเป็นต้องตัวเล็ก
- การสร้าง Object ที่โดย class สร้าง object = new class();
ยกเว้น String ที่สร้าง
หรือ String word = new String("something");
String word = "Something";
- การใช้ method ทำให้ method ต่อ class object ที่สร้างจาก class นั้น เชื่อมต่อ .
เช่น word.substring(2, 5);
- constructor คือ method ที่มีชื่อเหมือน class โดยรับ参数 เริ่มต้น 0:7 ซึ่งอย่างที่นี่ object ไม่มี new
- parameter คือ สิ่งที่ต้องมีใน (...)
- method ที่ชื่อชื่อ set ลักษณะจะเป็น public void set... (int a, String b ...)
ที่มีใน class ตัวแปรที่รับมาส่งค่าหันไป instance var. ไม่มี return
- method ที่ชื่อชื่อ get ลักษณะจะเป็น public int get...()
จะต้อง return ค่ากลับ สิ่งที่ return เป็นอะไรก็ได้ที่ return คือ return ที่นี่ class
ที่เขียนชื่อนามของก็ได้ ไม่มี parameter
- method จะมี: การสืบสืบ public
instance var. เช่น private
- method ที่ชื่อชื่อ กันได้ ไม่ parameter ไม่ใช่หน่วยเดียวกัน
(ถ้าจดกันไว้ overloaded หรือ polymorphism ประกอบกันมีการกำหนดชื่อ method
ที่เขียนแยกกัน)
- การสร้าง array ที่ `[]` เช่น [] (อย่างไรก็มีพิธีคือเพิ่มเข้าไป)
เช่น `int[] num = new int[5];` หรือ `int[] num = {4, 10, 6, 0, -1};`
- `ArrayList<ชื่อคลาส> list = new ArrayList<ชื่อคลาส>();` บันทึก () ขึ้นหลัง!
ที่ชื่อคลาส เช่น ชื่อแบบง่าย ที่ไม่เกิดข้อข้อ เช่น Integer หรือ String
เช่น `100 Student` ประกอบ เช่น ที่ไม่เรียบร้อย class Student ชื่อคลาส (ต้องรู้ 100)
- static สำหรับ ที่ทุก object ที่สร้างจาก class เดิมกันใช้ร่วมกัน
- ถ้าใช้หนึ่งกันนั้น ชื่อ method นี้เป็น variable (ตัวแปร) และจำนวน parameter ไม่ใช้กันตาม ไม่ว่าจะ ค่าไหนๆ ที่ไม่เหลือ!

The End !
จบแล้วจ้า...

* ไม่ระบุชื่อของผู้เข้าร่วมในปีที่ 1-25 1000:

Programming part II. (for Final)

สรุปโดย คร.

「Ta」
sec. 1 5188018
#ICT

インターフェース

Interface

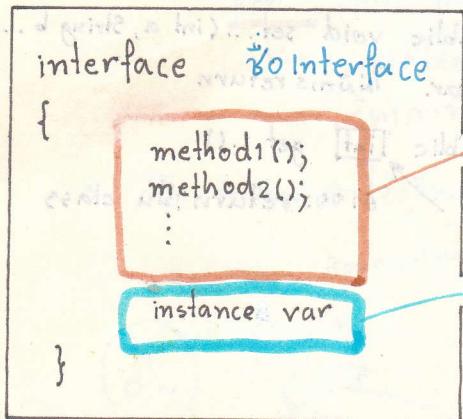
- ↳ เมื่อ class มี method ที่จะบอกได้ว่า สามารถทำอะไรได้เท่านั้น แต่ไม่แน่ชัวร์ว่า
- (มีแต่ชื่อ method แต่ไม่รู้จะใช้งานการว่าห้องอะไร ยังไง) ต้องดูคือ?

interface แค่ต้องมี interface-able

→ เรื่องนี้ๆ ก็ต้องยกมา ก็ต้องยกมา!
ก็ต้องยกมา!

จะอย่างเดียวคงจะว่าชื่อจะเด็ดขาดมาก

ส่วนประกอบของ Interface



ปกติเวลาสร้าง class เราจะใช้ตัว var class ตามด้วยชื่อ, แต่ถ้าเป็น Interface จะใช้ตัว interface แทนที่เดิม แต่ต้อง

เขียนชื่อ method ตามด้วย () ;
หมายความว่า ไม่ต้องเขียนว่า method ที่อยู่ใน

ประกาศต่อตัวที่ของ interface
ลืมไว้ ก็ได้! แต่ต้องมี instance var

แล้ว interface มีไว้ทำไม?

- ▶ เวลาสร้าง class ধันคลายๆ รัน เราอาจจะมี method ที่ทำงานเหมือนกัน (นี้เรียกว่าเหมือนกัน)
เช่น มี class เครื่องบิน และ class รถ อยู่ ทั้ง เครื่องบิน และ รถ มีความสามารถยื่นเข้า (หรือยื่นออก) มากกว่า อย่างหนึ่ง) ที่เหมือนกัน叫做 "บิน()"

↳ เมื่อมันมี "บิน()" เหมือนกัน เราจึงควรสร้าง interface สามารถบินได้ (flyable)
แล้วใช่ความสามารถบินได้ ที่ บิน กับ เครื่องบิน กับ รถ (ที่... ที่ เครื่องบิน และ รถ
ก็สามารถบินได้) แต่ บิน ขึ้นไป ที่ class เครื่องบิน กับ รถ ที่ห้องไปบินกันเอง
เราต้องรู้ว่า ตอนนี้ ใช่ 2 รันนี่ อีก บัน บินได้ แต่ นั่น

- ▶ สมมุติเราเขียน method ที่บินฟังฟ้า() รับ object เครื่องบิน รันพิมพ์แล้ว สั่ง "บิน()"

```

public void บินฟังฟ้า(เครื่องบิน ค)
{
    System.out.println(ค.บิน());
}
  
```

เราบัน เครื่องบิน 9 ล้อ แล้ว สั่ง ค.บิน() ให้

แต่เราใส่ object ที่สร้างจาก รถ ไม่ได้

เพราะ รถ ≠ เครื่องบิน ห้าม กับ บันบินได้ทั้งหมด

↪ วิธีแก้ก็คือ เขียน method บนห้องฟ้า() ใหม่... บอกว่า ให้รันต์ໄก์ที่ก็คือ สามารถบิน ก็ต้อง เราก็ต้อง เลือดบิน หรือ นก (ซึ่งต้องมีส่วนก่อหนี้ เกรื่องบินกับนก บินได้) ส่วนที่บินห้องฟ้า() ให้

↓ เป็น code Java ก็ต้อง implements

คำอ่านสิ่งที่พูดในเมื่อ ก็มานเขียนใน Java ว่า ?

```
interface Flyable
{
    public String fly();
}
```

← บอกว่า ความสามารถ "สามารถบินได้"
ปะกอนด้วย ความสามารถ "บิน"

บุกง่าย อีกที สามารถบิน(flyable) ก็ต้องบินได้(fly())

```
class Plane implements Flyable
{
    public String fly()
    {
        return "บินเร็วมาก";
    }
    // นี่ method สำคัญมาก
}
```

← บอกว่า เกรื่องบิน บินได้ ปะบันนี่

► รันความสามารถ Flyable นา แต่รุ่นเดียว บินเร็วมาก
เกรื่องบิน

```
class Bird implements Flyable
{
    public String fly()
    {
        return "บินเร็วมาก";
    }
    // นี่ method สำคัญมาก
}
```

← บอกว่า นก บินได้ ปะบันนี่

► รันความสามารถ Flyable นา แต่รุ่นเดียว บินเร็วมาก

* สูญเสียเวลาเขียน แบบ interface : เราอย่างที่ class ไหน มี ความสามารถ แบบนั้นๆ เราก็ implements ความสามารถนั้นมา (ชื่อ interface ก็จะมี suffix ท้าย ด้วย ก้าว -able = ความสามารถ)

↪ หลังจาก ดูความสามารถมา เราต้อง บอกด้วยว่า ความสามารถนั้น เมื่อมาอยู่ใน class นี้ เป็นไร โดยการเขียน method ใหม่ ที่บันลักษณะ

!
ระวัง... การ implements interface ต้องใส่ method ทุกอย่าง ใน interface นั้น
มาเขียนใหม่ด้วย เช่น ก็ interface Flyable มี method ชื่อ public String wing()
เพิ่มเข้ามาแล้ว Plane กับ Bird ตามนี้ โดยการเขียน method ใหม่ ที่บันลักษณะ

สร้าง Object จาก interface

การสร้าง object ที่จะต่างจาก การสร้าง object จาก class ... ฝึกหัดแล้ว เราจะเขียนแบบนี้

```
class Object = new class();
```

1

2

รูปแบบนี้ เมื่อ click ที่ๆ ก็ เป็นการสร้างหน้านั้น
แต่ไม่ได้สร้างขึ้นมา ภายนอกต้อง เสิร์ฟเวอร์

→ ก็ต้องมี "new"

ตรงนี้ ส่วนว่า ใช้สร้าง object มาตัวนึง

ซึ่งล้วนๆ ก็เป็น class ที่เราๆ ก็
ไป copy class ทันทีเลย

ตรงนี้ สร้างตัวไปประมวล 1 ตัว
โดยมาเก็บข้อมูลประมวล [class]
เก็บตัวนึง

3

จับตัวไปประมวล
เก็บกัน object ที่
เพิ่งสร้างเมื่อตัวนี้

เวลาเราสร้าง object จาก interface ว่า ? ... ก็สร้างแบบนี้

```
interface Object = new interface();
```

← เป็นหนึ่งไปเดียวกันแล้ว!

→ สร้างแบบนี้ ไม่ได้! เพราะ interface มีแต่ method ไม่ได้มีบล็อก method
ที่ถูกสร้างไปที่ได้ → ทางแท้: เวลา สั่ง =new ให้ new ชนชั้น class ซึ่ง implements
interface ตัวนั้นๆ ... ดังต่อไปนี้ ... ดูต้องเข้าใจว่า คืออะไร ?

(จากหนังสือแล้ว ...)

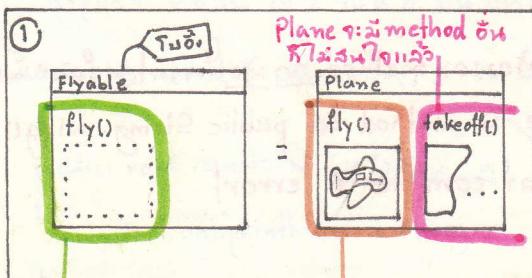
Ex. อย่าง 1 interface Flyable มาสร้าง object ซักตัว ...

step

1 Flyable โน๊ต 747

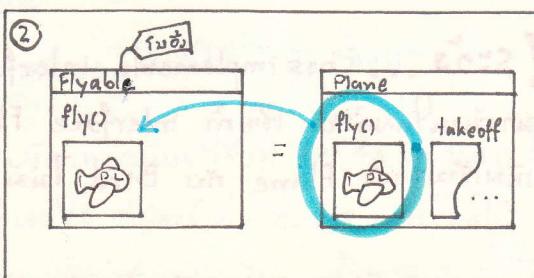
← สร้าง object ที่ "บินได้" มาตัวซึ่ง โน๊ต 747

2 Flyable โน๊ต 747 = new Plane(); ← บ่งบอกว่า object "บินได้" ตัวนี้
Plane "บินได้" บินได้บนฟลัต เกริลิงฟลิต (Plane)
เขยรับเท่ากับ โน๊ต 747



ตรงนี้ fly ตัว
ไม่ใช่บินได้

fly ของ Plane
"บิน" แบบ "เกริลิง"



Flyable ตัวนี้ fly() เดียวไปcopy มาจาก
fly() แบบ "เกริลิง"

> กดๆๆๆ Flyable แล้ว test ที่ main อะ...

```
public static void main (String[] args)
{
    Flyable P = new Plane();
    System.out.println (P.fly());
}
```

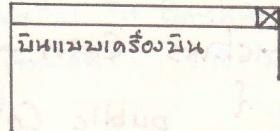
สร้าง object ที่มันได้แบบ "เครื่องบิน"

method fly() ของ เครื่องบิน

return String คือว่า "บินแบบเครื่องบิน"

(class เครื่องบิน (Plane) อายุหน้า 27)

output



แล้วถ้าจัดให้ object Flyable บินแบบเด่นชัดว่า? :

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

Flyable something = new Plane();

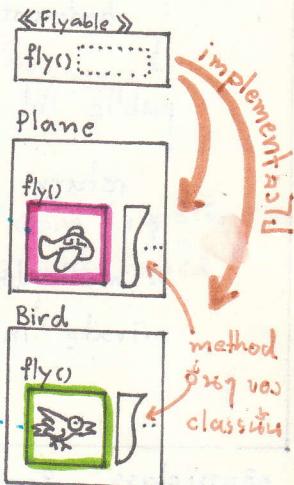
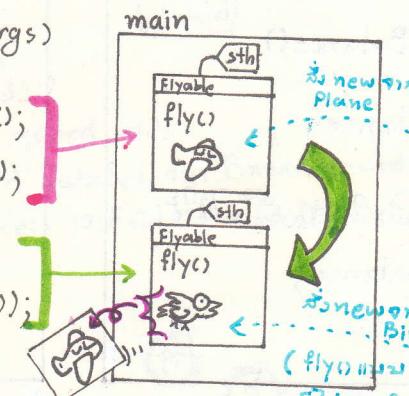
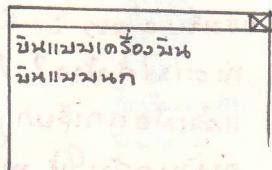
System.out.println (something.fly());

something = new Bird();

System.out.println (something.fly());

}

output



* สำหรับ something.fly() เหมือนกันแต่ output ต่างกัน
เพราะ: fly() มาจาก class และ class รัน

สรุป... จัด: สร้าง object interface ก็สร้างตามนี้

interface ZooObject = new class();

* In class นี้ๆๆๆ ต้อง implements interface ได้ในเวลตัวเอง เช่น
class Plane implements Flyable, Takeoffable
;

เช่นเดียวกับ interface นั้นๆๆๆ

โดยที่ class นั้นๆๆๆ ต้องเขียน class ที่ implements interface นั้นๆๆๆ

takeoff ให้ตรงกับ
ก็ต้องทำให้เป็น interface ด้วย

สังเกตุ: สร้าง object interface ก็สร้าง ใจ: ถูกสร้าง (new) ทาง class

||| ทำงานของ method ของ interface class

เช่น... ถ้า Plane มี ... object interface ที่มี method fly() ให้ return คำเรียก interface takeoff()

จะ error! (ผิดง่ายๆ ถ้า ตอนสร้าง interface

จะ method อะไรเข้าไป ก็จะได้แล้วแน่นอน:)

↳ แต่ถ้าสร้าง object จาก Plane แล้ว... ถ้า takeoff() ก็ไม่ใช่ function!

```
public String fly()
{
    return "บินแบบเครื่องบิน";
}
public String takeoff()
{
    return "ขึ้นฟ้าแล้ว!";
}
```

การเอา interface ไปผูกกับการรับตัวแปร parameter

↳ อนุมอนหัวใจ: คือตัวแปร BankAccount Coin แบบ DataSet ที่มี type ของตัวแปร comment: class 3 ตัวนี้ ลงเข็มกัน แล้ววีน: ... ทุกๆ กันหันด้วย สักกัน

* เพื่อคนที่เข้าเก็บข้อมูล text หา แล้ว: ต้องแน่ใจว่าบุกสิ่ง class เดียว กันจะเขียนใหม่ เลย

เพราะ: 3 class ที่มี text เดียวกัน ตัดแปลง แก้ไข ทำซ้ำ ... จุดเด่น บันทึกภาษา!

- ระบุชื่อความต่างๆ ของ int กันหมด!

class BankAccount :

```
{
    public BankAccount (int inAmount)
    {
        balance = inAmount;
    }
    public int getBalance()
    {
        return balance;
    }
    // method ที่ต้องมีใน Class เรื่องหัวใจเดิม!
    private int balance;
}
```

อธิบาย class: BankAccount เอาไว้เก็บยอดเงิน
และเก็บตัวแปร balance ตัวหนึ่ง: โดยมีเมื่อเวลาเรียก
getBalance() ก็จะ return ยอดเงิน บนหน้าจอหนึ่ง
ไปที่ๆ main

class Coin

```
{
    public Coin (int inValue)
    {
        value = inValue;
    }
    public int getValue()
    {
        return value;
    }
    // method ที่ไม่ใช่หัวใจเดิม:
    private int value;
}
```

อธิบาย class: Coin ตัวนี้เป็น บันทึก
หัวใจเดิม? ก็รับตัวเงิน (value) มา
แล้วเมื่อถูกเรียก getValue() ก็ return
ตัวเงินกันไป main

class DataSet // บันทึกกับ BankAccount ให้เข้า:

```
{
    public DataSet()
    {
        count = 0;
        sum = 0;
        max = null;
    }
    public void add(BankAccount b)
    {
        sum += b.getBalance();
        if (count == 0 || max.getBalance()
            < b.getBalance())
        {
            max = b;
        }
        count++;
    }
}
```

เพื่อให้ ตัว add สามารถ หักไว้
ตัว max หักไว้ตัวที่ add เท่านั้น → เป็นตัว max!

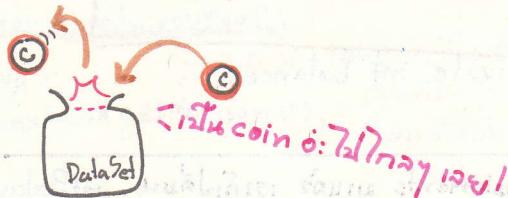
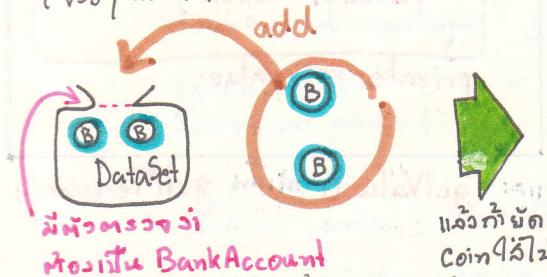
```

    public BankAccount getMax()
    {
        return max;
    }
    public int getAverage()
    {
        return sum / count;
    }
    private int count, sum;
    private BankAccount max;
}
```

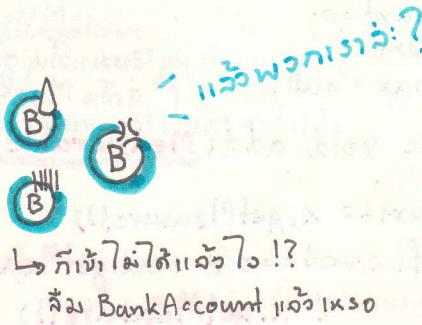
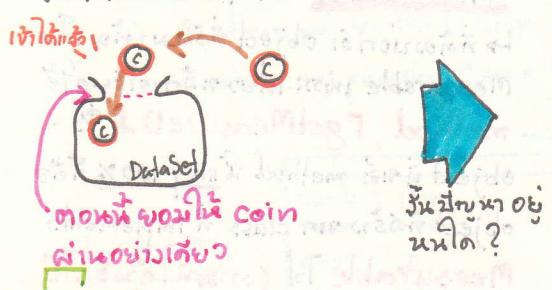
อธิบาย class อยู่หน้าโน๊ต

DataSet ใจกว้างเกินขีดจำกัด เรา add object เท่าไหร่ก็ได้เรื่องๆ โดย DataSet จะใช้หน้าต่าง max เก็บเอาไว้ 9 หน่วยแบบ object BankAccount และถ้า sum ใจกว้างทุกรอบที่มีการ add() ก็ทางฝั่ง main เรียก getMax() ก็จะ return ตัวที่ max ก่อนไป 9 หน่วยแบบ object, getAverage() จะ return ค่าเฉลี่ยแบบ int ก่อนไป (ต่อจากนั้น mega 9 หน่วย แต่เราต้องการความ 정ยันต์ !)

มาดูตอนนี้ เราได้ DataSet กับ BankAccount 7 ตัว อย่างนี้ชี๊บนา ! ... อื้ว !? แล้ว กี่ เรายังคงใช้ได้ นี่ ก็ยังคงใช้ได้ กันไปอีก !!!



▶ 乍วากัน คงจะสับ แล้วลิหะ: ว่า ถ้าเรา เก็บ DataSet จาก BankAccount → Coin ก็จะนัวรำ!
แล้ว กี่ เนี้ยยยยย → getValue() เรายังคง coin 9 ตัว DataSet ได้... แต่ซึ่ง ก็ทำได้!
งานที่รอบ 2 ถูกนั้นแน่!



[?] สังเกตดู: ว่า BankAccount กับ Coin มี method ที่ ก็จะย่า กันดีๆ getBalance() กับ getValue() (เนื้องนัยว่า... ก็ต้องด่า หลักของ class ให้ดีกันไป)
ก็ผ่านมา เราได้ Plane กับ Bird เป็น Flyable เพราะพ่วงมัน fly() ได้
↳ รู้ เรา ก็ สามารถ นำ BankAccount กับ Coin ดันตัวได้ เมื่อันกัน แล้ว ก็ interface ได้!

```
interface Measurable
{
    public int getMeasure();
}
```

* จากการถูก class A นั้น measure = วัดตัวได้, ประมาณ

→ ผิดๆ : สิ่งที่ "สามารถวัดตัวได้" getMeasure()
ก็สามารถ "ตัวตัว ()" อะไรบางอย่าง
กลับมาได้

จากนั้น ที ใจ... ก็ไปเพิ่มน BankAccount กับ Coin, และ DataSet เพิ่ม
↳ implements

↳ ไปเปลี่ยน parameter
กัน max

เขียนห้องการบดบังซักน้ำเก็บเงินได้: T^T

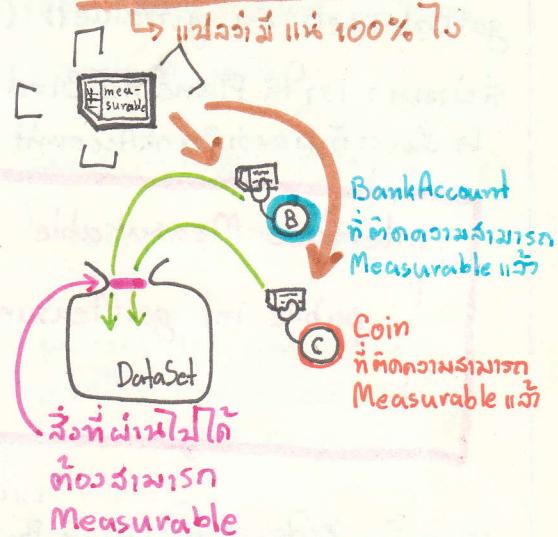
```
class BankAccount implements Measurable
{
    public BankAccount(int inAmount)
    {
        balance = inAmount;
    }
    public int getMeasure()
    {
        return balance;
    }
    private int balance;
}
```

```
class Coin implements Measurable
{
    public Coin(int inValue)
    {
        value = inValue;
    }
    public int getMeasure()
    {
        return value;
    }
    private int value;
}
```

ห้องที่ implements อะไรแล้ว เราก็เปลี่ยน `getBalance()` // ให้เป็น `getMeasure()`
แนว: method ที่ 2 ต้องทำให้เป็น Measurable

```
class DataSet // version 2
{
    public DataSet()
    {
        count = 0;
        sum = 0;
        max = null; // รับมาเป็น object
    }
    public void add(Measurable x)
    {
        sum += x.getMeasure();
        if(count == 0 || max.getMeasure() < x.getMeasure())
        {
            max = x;
        }
        count++;
    }
    public Measurable getMax()
    {
        return max;
    }
    public int getAverage()
    {
        return sum / count;
    }
    private int count, sum;
    private Measurable max; // max เป็น object
}
```

หากฝึกที่การรับตัว parameter รับได้ 1 อย่างเดียว (ต้องเลือก: ห้อง Bank หรือ Coin)
ตอนนี้ชื่อง่าย รับ object 0: ใจคิดว่า "สามารถวัดได้" มาก (รับเป็น Measurable)
→ ที่ต้องบอกว่า object ที่รับมาต้องเป็น Measurable เพราะ: เดียวจะมีการเรียกใช้ method `getMeasure()` ซึ่ง object ที่รับ method นี้ ไม่ได้ 100% ก็ต้อง object ที่สร้างจาก class ที่ implements Measurable แล้ว (ตามที่ได้ลงไว้)
ก่อนที่ 2: 0: ถ้า implements ไม่แล้ว ต้องเขียน method (ให้ตรงไปวัน error!)

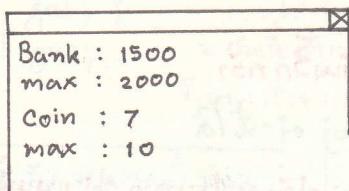


เขียน ตัว Tester เพื่อความเที่ยงตรงกันนั้น (ว่า: นั้น..!?)

```

1 class Tester
2 {
3     public static void main (String[] args)
4     {
5         DataSet data = new DataSet(); ← สร้าง object
6         data.add (new BankAccount(1000));
7         data.add (new BankAccount(2000)); } งตรงนี้ กวนไปถูกหน้า 21, 28
8         int avg = data.getAverage(); ← ข้อดี: คลับ (ให้ทันเมื่อตอน Bank q )
9         BankAccount maxBank = (BankAccount) data.getMax(); เมื่อจาก สนับ return เป็น Measurable
10        System.out.println ("Bank : " + avg);
11        System.out.println ("max : " + maxBank.getMeasure()); เลยต้องเปลี่ยนเป็น Bank q
12        DataSet data2 = new DataSet(); ← สร้าง object อีกหนึ่ง
13        data2.add (new Coin(5));
14        data2.add (new Coin(10)); } ก่อจังหวะอยู่ตาก็ไปถูกหน้า 21 กับ 28 ต่อ:
15        int avg2 = data2.getAverage(); ← ข้อดี: คลับ (ให้ทันเมื่อตอน Coin)
16        Coin maxCoin = (Coin) data2.getMax(); * ก่อใจให้ใจ ในการเพื่อน
17        System.out.println ("Coin : " + avg2); เนื่องด้วยเหตุผลว่าบันทุกอย่างเป็น double
18        System.out.println ("max : " + maxCoin.getMeasure()); เป็นลักษณะ Coin
19
20    }

```



! ตาม sense แล้ว ต้องลับของ coin
ควรเป็น 7.5 แต่ว่าใช้ int เลยเหลือแต่ 7
▷ อย่างให้ถูกใจแก้ไขให้เป็น double ช.



สังเกตดู: ว่า บรรทัดที่ 9 หลังจาก เรียก data.getMax() เราจะลืมแม้แต่

กันก่อเมื่อ BankAccount 7 ตัว (โดยวิธีเดียวที่นักการเปลี่ยน double → int [int x=(int)5.5;]
อะไรมากันนั้น) โดยการเปลี่ยน แนวโน้มเรารอต้อง ซัพ ว่า object Measurable กันหนึ่งเมื่อ
BankAccount (ก่อนดัน เมื่อ Measurable แบบ Coin ผิด: error!) ... ภัยที่นี้เรา
รู้ว่า มันเมื่อ "แบบ BankAccount," ไม่เหมือน ... ก็ add 7 ตัว BankAccount ที่ บันทุก
คงมาเมื่อ Coin 7 ตัวนอง --"
ส่วนใน บรรทัดที่ 16 ก็ต้องเนื่องกัน แต่เมื่อ "แบบ Coin" แทน
วิธีแบบนี้เรียกว่า การ Cast

class Object

↳ class Object (ชั้นต่ำที่สุด) สำหรับเรียกหน้าที่ "ควร" ผูกกับซากเล็กน้อย ก่อนที่จะ inheritance

↳ class Object (ชั้นต่ำที่สุด) สำหรับเรียกหน้าที่ "ควร" ผูกกับซากเล็กน้อย ก่อนที่จะ inheritance

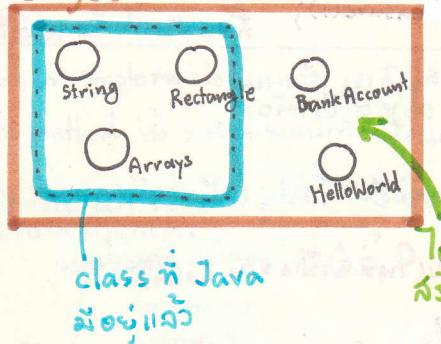
矩形, ArrayList (หรือแม้กระทั่ง Integer, Double และ Boolean) เป็น class ที่มีอยู่ใน Java

มี ...!? และบ่อยครั้งที่มันมีรูปแบบที่ไม่ใช่ rectangle

▶ เมื่อพูดถึง Object เป็น "class ใด" "ของทุกๆ class" หมายความว่า BankAccount หรือ Coin ที่เราสร้างขึ้นมาเป็น class ที่อยู่ใน Object เป็น subset ของ Object

class ทุกๆ class \subset Object ; ทุก class เป็น subset ของ Object

Object



▶ หมายความว่า Object คือ ทุกๆ วัสดุ Object ที่

สร้างจาก Object นี่ เก็บ (object) ลง class
ดังนั้น Object ที่สร้างกันเอง!

▶ หมายความว่า Object ที่สร้างกันเอง

▶ Object obj1 = new String("o-b-j-e-c-t...");

↳ Object เก็บ String ไว้

▶ String str = obj1;

↳ String เก็บ Object ได้! つまり: ชนิดเดียวกัน

? วันนี้: ห้าวใจ str สามารถเก็บ obj ตัวเดียวได้

▶ String str = (String) obj1; ← การเปลี่ยนแปลง obj ใหม่เรียกว่า "Cast"

↳ บอกให้เปลี่ยน obj1 ไปเป็น String
→ str เก็บ String ได้แล้ว

* object ที่สร้างจาก Object เก็บได้ ทุกอย่าง และ:

class ทุก class ที่อยู่ใน Java นั้น มี relation extends Object
ที่แน่นอน แต่ไม่ชัดเจน เช่น class (compiler จัดการให้!)

Inheritance

インハーリタンス

ภาษาไทยใช้ต่อว่า "การสืบทอด"

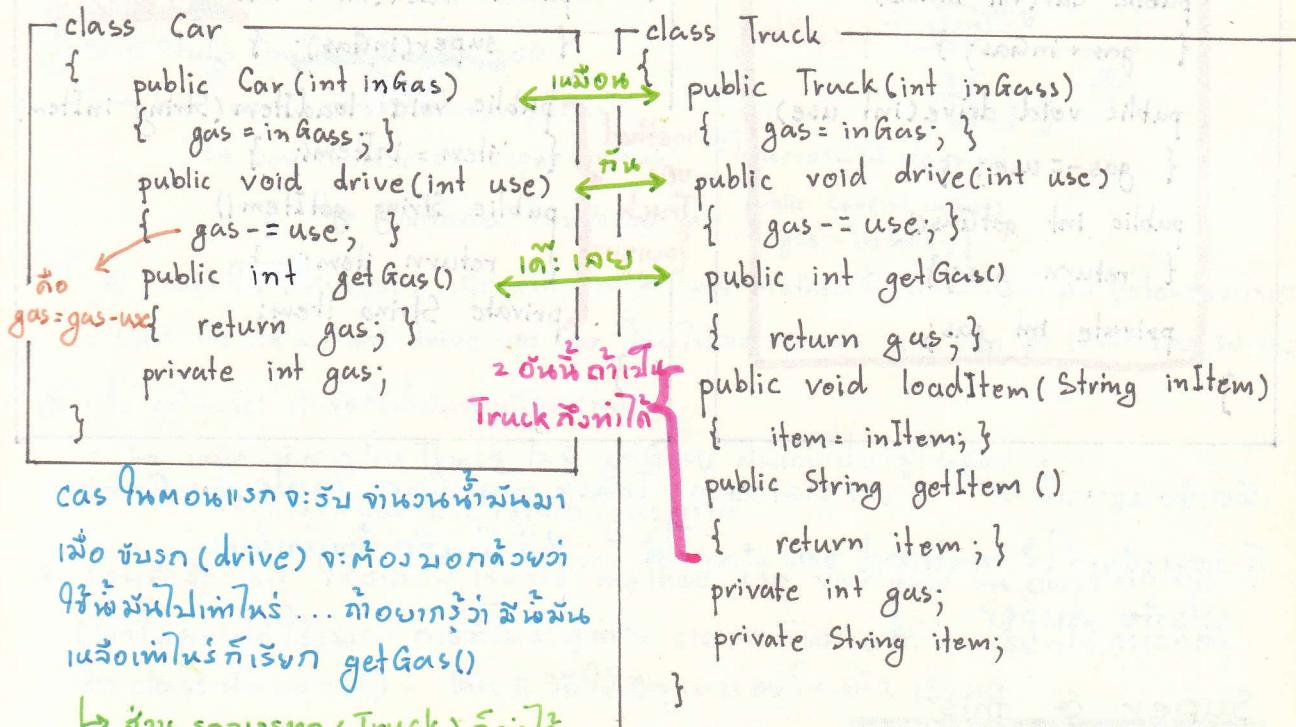
↳ Inheritance คือการ สืบทอด หรือ สืบทอดภาระ ของ class หนึ่งไปสู่ class อีก class หนึ่ง โดย class ที่รับการสืบทอดจะมีลักษณะเหมือน (หรือมากกว่า) class ที่ตน [[[[[]]]]]

* Inheritance มันจะคล้ายๆ กับ Interface แต่กันตรงที่

- Inheritance มี instance variable (จะมาตัวประกอบ)
- class ที่ตน [[[[[]]]]] ให้ method ที่เป็นแบบ public (public void setValue() { ... })
- ใน interface ไม่มี method ที่เป็นแบบ public (public void setValue();)
- ในต้องเขียน method ทั้งหมดเดิม, ถ้า class ที่ตน [[[[[]]]]] method อยู่แล้ว ก็เรียกได้เลย

แล้วเมื่อไหร่ ถึงจะใช้ Inheritance

Ex. ถ้ามี class Car และ Truck



car ที่มีอยู่แรก จะรับ จำนวนห้ามนำ

แล้วขับรถ (drive) จะหักบớtกี่ด้วย

ที่เหลือนำไปใหม่ ... ก็คงยกเว้น ว่ามีเงื่อนไข
เมื่อเท่านั้น ก็รีบกัน getGas()

↳ ส่วน รถบรรทุก (Truck) ก็จะได้

เมื่อ car [[[[[]]]]] รถบรรทุก สำหรับ

บรรทุกของเพิ่มได้ด้วย

* จากข้างบน จะเห็นว่า Truck ล้วน method 3 ตัวที่ Car มี ยกเว้น `{...}` ซึ่ง
เนื่องจากต้อง → ผูกง่าย ถ้า Truck รีบไป Car เสียด้วยความน่าจะดี
Car ถูกอย่าง (รวมทั้ง copy หาที่แนบ เช่น)

▶ เมื่อปีนี้ บันทึกไว้: สั่งให้ Truck extends Car !

copy ความสามารถที่ class ดึง คือ Extends

อนุรักษ์ Interface ไว้: implements เมื่อ class นั้น ลืมความสามารถ แบบ interface นั้น

อนุรักษ์ Inheritance ไว้: ใช้คำว่า "extends" แทน โดยจะ extends เมื่อ class ที่นั่น

"เป็น" class ต้นไป

Ex. รถบรรทุก เป็น รถ: รถนั้น → class Truck extends Car

class subclass extends superclass

Note

คลาสแม่

= Parent หรือ Base class

คลาสลูก

= Derived หรือ Child class

例 Car กับ Truck จาระยนนิ่ง

class Car

```
{
    public Car(int inGas)
    {
        gas = inGas;
    }

    public void drive(int use)
    {
        gas -= use;
    }

    public int getGas()
    {
        return gas;
    }

    private int gas;
}
```

class Truck extends Car

```
{
    public Truck(int inGas)
    {
        super(inGas);
    }

    method
    ของ
    Truck
    ที่อยู่ใน:
    {
        public void loadItem(String inItem)
        {
            item = inItem;
        }

        public String getItem()
        {
            return item;
        }

        private String item;
    }
}
```

เมื่อบอกนี้ : เราจะบอกว่า Truck เป็น Car อนุรักษ์เป็น Car แล้ว ก็ยังคงเรียกใช้ method ของ Car ได้ ... ที่นี่ ก็จะนับถือเป็นหน้ากึ่ง อะไรก็ได้ super...?

super & this

โครงสร้าง extend มันไม่เหมือน implements ที่เราต้องเขียน method ใหม่อีก เช่น อนุรักษ์ Truck สามารถเรียกใช้ drive(...) กับ getGas() ได้ (เรียก gas ที่เป็น variable ไว้ได้ เพราะ private ไม่ถูก) โดยที่ Truck เอง ก็ไม่มี method พวกนั้น ...

- แล้วเราจะบอก Truck ยังไง เวลาจะใช้ method จาก "คลาสแม่"

▶ เราต้องระบุ super, โดยการเรียก super ที่หมายถึง การหันไปถูกคลาสแม่ ของมัน ส่วน super(inGas) ก็หมายถึง หันไปถูกตัว inGas ที่เป็น constructor ของ class และ (super เป็นที่ หมายถึง constructor)

- เราอาจเรียก super.drive(...) หรือ super.getGas() หากใน Truck ที่ได้] super.ชื่омethod หรือ super.ชื่อvariable ก็ได้ (ที่ gas ที่มีชื่อ private ไว้เรียก super.gas จะใช้ลง ก็ได้)

อุตสาหกรรมเพื่อทดสอบ Tester

```

1 class Tester {
2 {
3     public static void main(String[] args) {
4     {
5         Truck t = new Truck(200);
6         t.loadItem("anime");
7         t.drive(50);
8         System.out.println("รถบรรทุก " + t.getItem());
9         System.out.println("เหลือที่มั่น " + t.getGas());
10    }
11 }

```

Output

รถบรรทุก anime
เหลือที่มั่น 150

อธิบาย Code

⑤ สร้าง Truck โดยให้ที่มั่น เป็น 200

↳ 200 เป็น constructor ของ Truck

↳ constructor สั่ง super(inGas) ก็จะไปที่ class นั้น

↳ มีตัวแปร gas ของตัว 200 ใน public Car(int inGas)
{ gas = inGas; }

⑥ 从 Truck ที่ drive(50) เดินทางจาก Truck แล้ว method นี้ บันทึกว่า มัน extends มาจาก Car → 93 method drive ของ Car ลดลงต่อไป gas จะเป็น 50 (เหลือ 200 - 50 = 150)

⑦ 从 getGas() ต้องการที่มั่นที่เหลืออยู่

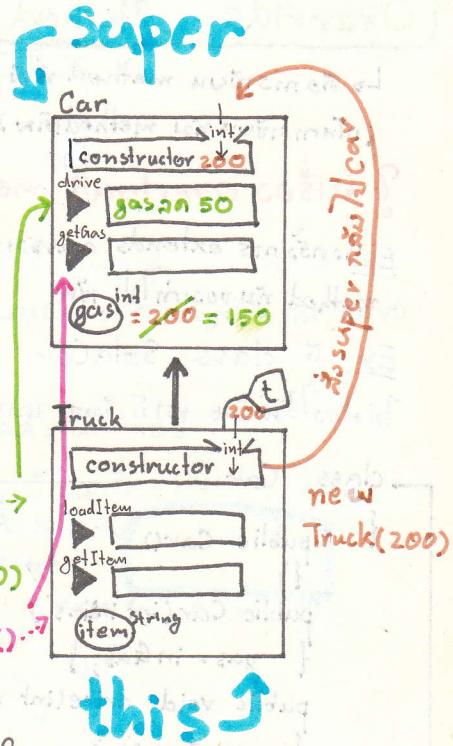
↳ เหลือเท่าไหร่ใน Truck ไม่ใช่ getGas() บันทึกไว้ในตัว Car

↳ ไปที่ Car มัน return gas ที่นี่

▶ สรุปคือ ชื่อ this ใช้ส่วนรับ เรียกว่า 93 method หรือ variable ของ class ตัวเอง (ถ้าไม่คิดว่าใช้เพื่อ: การที่มีรูปแบบต่างๆ super ก็สามารถเรียกการเรียกได้โดยท่องใน class นั้นอยู่แล้ว) this ลักษณะเดียวกันกับ 13 นี่:

สรุป

- 93 super เนื่องเรียก 93 method และ variable ของ class นั้น (ถ้าไม่คิดว่า private 73) จะใช้ใน class นั้น
- 93 this เมื่อ ระบุ 93 เนื่องเด่นว่า เรากำลังเรียก 93 method และ variable ของ class ที่เรียกเพื่อนั้น



this ↑

Override Method

↳ ต้องการเขียน method ที่มีชื่อ == parameter เหมือนกันทุกอย่าง แต่ต้อง ต่าง class กัน
(เมื่อกำรเขียน ก็จะ method นั้น)

? จึงเรียกว่า overload method 7 ตัวมือ: (ข้อมูลที่อยู่ในห้อง 11)

บางครั้ง กำร extends class บางตัว จะมี บ้าง method ที่ มันไม่ได้รับ class ลูก เรา ก็จะ override method กัน ของตัวใจ ใช่

Ex. มี class SolaCar (รถพลังงานแสงอาทิตย์) เมื่อขับรถ พลังงานแสงอาทิตย์ เดย
ให้ต้องใช้ผู้คน ไร้กีดขวาง แบบนี้ได้

class Car

```
{ public Car()
  {}
  public Car(int inGas)
  {
    gas = inGas;
  }
  public void drive(int use)
  {
    gas -= use;
  }
  ① public int getGas()
  {
    return gas;
  }
  private int gas;
```

class SolaCar extends Car

```
{ public SolaCar()
  {}
  ② public int getGas()
  {
    return 0;
  }
```

↑ สืบทอด

* เรียกว่า default constructor (มันต้อง 0: ใจท่องที่ 12.) มาเพิ่มใน Car เพื่อให้

SolaCar สามารถ new SolaCar()
แบบไม่ต้องมีส่วนร่วมหน้าต่าง เรียบร้อยได้

เพราะว่า SolaCar ไม่ต้องใช้ผู้คน, method getGas() จึงไม่ควรรัน
ประกอบ SolaCar จึงเรียกว่า **Override** getGas() และส่วนหนึ่น return 0
แทน

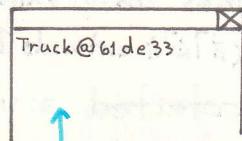
ถ้าเรา สั่ง ให้ SolaCar getGas() บน Java ตาม ② ไม่หนีบ่อน Truck ที่ใน
class ของตนเอง ไม่ใช่ getGas() ลับมาอยู่ห้องๆ กับ getGas() รัน ①

toString()

↳ toString เป็น method ที่จะถูกเรียกเมื่อเรา เรียกชื่อ object ไปกับ ไฟล์ไฟล์ .method()
ต้องหิบ, toString เป็น method ที่อยู่ใน Object ผู้ปะบันดี...

```
public String toString()
{
  return "ข้อความที่อยู่กัน return";
}
```

สมมุติว่า 101 object ของ Truck (นั่นคือ class 0: ใช้ getClass() แล้ว toString() แล้ว print)
 (Human)
 :
 Truck t = new Truck(100);
 System.out.println(t);
 :
เขียนมาใน code
การหาตัวตน (ที่อยู่ มันสื่อ address ของ mem n:)



→ ที่นี่ เราอย่างกันนี้ บัน print 0: ให้ออกมา ชื่อ กับ ส่วนบนเรียกว่า ชื่อ object อย่างเดียว
 โดยไม่มี .method() 0: ใจ เลย

สมมุติ ต้องการให้ 00 ก็ว่า Truck[Gas = 150, load = anime]

ก็ไปเพิ่ง method นี้ ให้มันใน Truck (ที่เปลี่ยนเป็นชื่อนี้เดิม เวลา !)

```
public String toString()
{
    return "Truck[ Gas = " + gas + ", load = " + item + "]";
}
```

↑ 00 ห้องประ (variable)

เมื่อที่ method นี้ แล้ว ก็เราเข้า code ว่า 00 ก็ว่า !

แบบข้างบน ลักษณะนี้ output จะ 00 ก็ว่า ...

Truck[Gas = 150, load = anime] แล้ว Truck@61de33

Access Modifier 00 protected

(ประจักษ์ การเข้าใช้ variable แบบ protected)

Ex: มี class (สินค้าพัสดุ) ชื่อ "Value" และ มี class ชื่อ "Amount" ที่ extends Value

```
class Value
{
    public void setValue(int v)
    {
        value = v;
    }
    public int getValue()
    {
        return value;
    }
    private int value;
}
```

```
class Amount extends Value
{
    public void setAmount (int a)
    {
        super.setValue(a+1);
    }
}
```

class Value สามารถรับ int มาจากพัสดุ แล้ว return ก้อน ส่วน Amount
 กากลับไป กัน แต่ รับตัวมา แล้วตอน return ตัว 0: เพิ่มขึ้น 1

→ แต่ในที่นี่ เราต้องการต่อ set ตัว กัน +1 แล้วให้ value ใหม่นั้น 00 ... ต้อง สั่ง super.setValue(...)

ก็ยังไง 0: สั่ง super.value = a+1; ก็ได้ แต่ value เป็น private

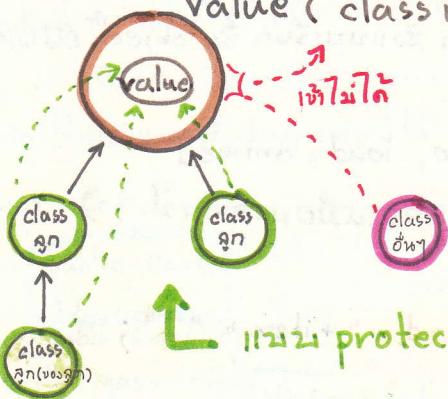
ก็ประกูลเข้า public ก็ส่ง ↑ ได้ แต่ class อันนี้ ทั้งหมด ก็ value ได้ด้วย !

ดูแล้ว กัน private 00: public ใจ work สำหรับงานนี้ ใจ ก็ส่ง แบบนี้กัน

protected int value;

การนองกว่า value เป็น protected หมายความว่า... ต้องมีการต่อสืบทอด class ให้เป็น public ได้
สำหรับ class อื่นๆ ทุกๆ class ห้ามใช้ value แบบ public ได้
แต่สำหรับ class ด้านใน (ที่ไม่ได้ extends ใดๆ) จะใช้ช่วงเวลาที่ value ได้ เป็น private
สรุป: ต้องไปรับแบบ protected จะสามารถใช้ได้ class ด้าน外 ไม่: class อื่นๆ ห้ามใช้
ที่มีสีชมพู

Value (class ภายนอก)



* ก้าวเป็น public
class ด้าน外 ห้ามใช้ value ได้

* ก้าวเป็น private
ไม่: class ด้านใน (รวมถึง class อื่นๆ) ห้ามใช้

||| หมายความ protected : class อื่นๆ ห้ามใช้

!* ระวัง... 1. กรณี extends สามารถ extend อะไรได้ แต่ class ด้าน外 ไม่สามารถ implement interface ที่ต้องการได้!

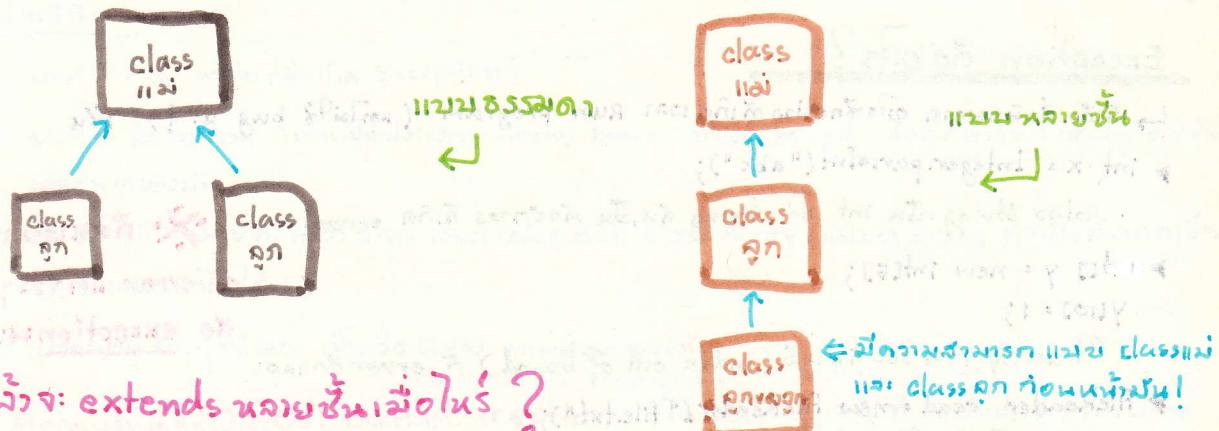
2. กรณี extends, constructor จะไม่ต้องไปด้วย (ไม่มี method และ variable เอง) ดังนั้นที่ class ใหม่ ช่อง构造 สร้าง default constructor ไว้เพื่อต้องการด้วย

3. ก้าว variable ที่ class ใหม่ เป็น private นำมายัง super.variable
ให้ลองหาดูว่า ลักษณะ method ในชั้นเดียวกันจะเปลี่ยนเป็น var. ผู้คนนี้ได้ชั่ง ให้มีส่วน
super.method(...) แทน



การ Extends หมายความว่า

ต้องเข้าที่ผ่านมา เราเอา class Car เป็น class และ extends ให้เป็น Truck และ SolarCar แต่การ extends เรากำหนดค่าชั้นได้ดังนี้



แล้วจะ extends หมายความว่าในรูป?

กรณี 1 : ง่ายกันสุด... ใจหายสับ ! ๘

กรณี 2 : เมื่อ class ลูก ยังคงมีฟังก์ชันเดิมๆ ได้อีก เช่น มี class เครื่องเสียง , ปักก้า และ ปักก้าลูกสั้น

↳ ปักก้าลูกสั้น เป็น ปักก้า , ปักก้า เป็น เครื่องเสียง

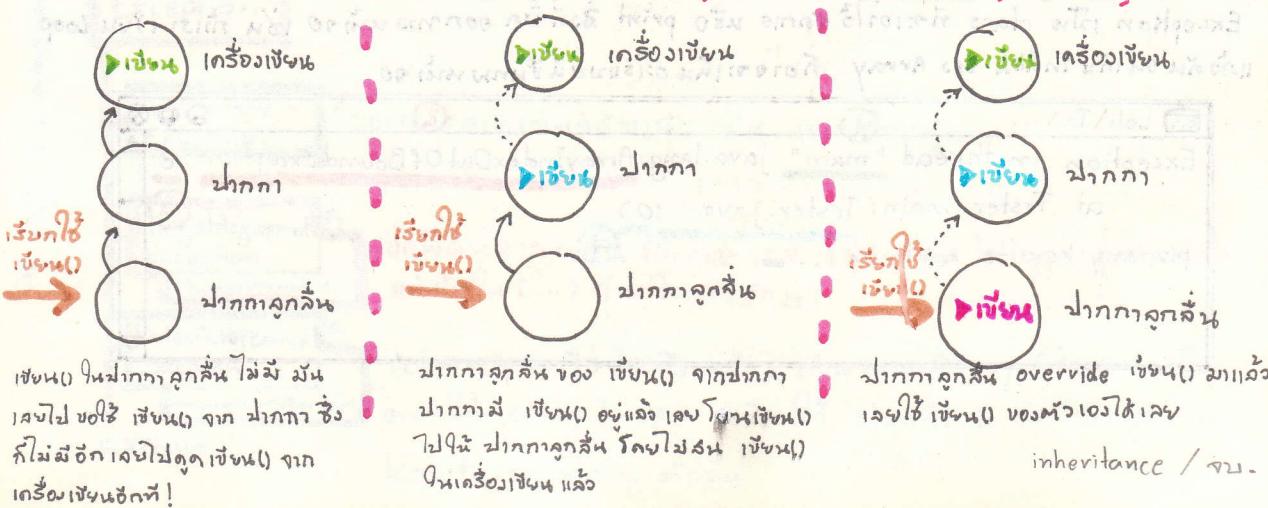
class เครื่องเสียง { };	class ปักก้า <u>extends</u> เครื่องเสียง { };	class ปักก้าลูกสั้น <u>extends</u> ปักก้า { };
-------------------------------	---	--

สำคัญ: เรียกว่า method ในเครื่องเสียง ของ ปักก้าลูกสั้น เราสามารถใช้ super.method()
ได้ต่อเนื่องติด (* แต่ถ้า ปักก้า ฝึก override method จาก เครื่องเสียง มา ก่อน จะ ...)

ปักก้าลูกสั้น จะทำ method นั้น แบบ ปักก้า ไม่ใช่จาก เครื่องเสียง !

Ex. ยกตัว เครื่องเสียง ล้วน method [เช่น ()] แล้ว เราก็ สั่ง [ปักก้าลูกสั้น] ให้ เช่น ()

แบบ 1 เช่น() อยู่ที่
เครื่องเสียง บังคับ
แบบ 2 อยู่ปักก้า ก็แล้ว
เช่น() (override น้ำ)



Exception

エクセプション

- การป้องกัน exception (ห้ามผิดพลาด error!)

- throws (ห้าม throw)

- try-catch (ห้าม finally)

Exception ต้องอะไร?

↳ ผู้เขียน สิ่งใดก็ตาม 0: ให้ซื้อของ ที่เกิด เวลา Run Program (หา bug หรือ error ให้ !) ใช่ๆ

▶ int x = Integer.parseInt("abc");

แปลว่า String เป็น int และ String คืนเป็น ผิดพลาด ก็เกิด error!

▶ int[] y = new int[5];

y[10] = 1;

9: array เก็บ ช่อง กี่ช่อง (index out of bound) ก็ error ดีๆ แบบ

▶ FileReader read = new FileReader("file.txt");

คืนมันไปร์ file.txt ไม่เจอก็ error!

→ 7: error พวกนี้ หน่วย ดู Exception!

ตรวจสอบ Exception ด้วย throws.

รูปแบบ คำสั่ง จ

```
class ชื่อ
{
    public void method(...) throws Exception
    {
        ...
    }
}
```

throws อยู่ในชื่อ method
ตามด้วย class [Exception]
ช้าๆๆๆ

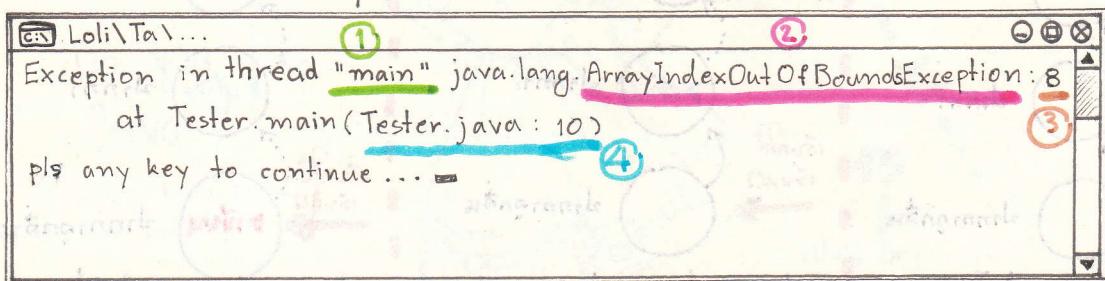
เขียนอย่างนี้ มีความหมายว่า... ถ้า method ทำงาน แล้วมี error เกิดขึ้น ให้ โยนไปที่ class Exception

ต้องการ → throws Exception]

การใช้คำสั่ง throws เป็นการตรวจสอบข้อผิดพลาด แบบง่ายที่สุด (อะ: มั่น:) เพราะ: เวลาไม่ต้องบอก โปรแกรมว่า จะ ดูตรงไหน แค่บอกว่า ถ้า ล้มเหลวเกิด ก็ โยนต่อไป class Exception จัดการเลย

แล้ว class Exception ต้องอะไร?

Exception ร่วม class ที่จะสามารถ ห้าม print สิ่งที่ผิด ออกทางหน้าจอ เช่น ถ้าเรา เช่น Loop แล้วดัน วนเกิน index ของ Array ก็ อาจจะ เกิด error แบบนี้ ขึ้นทางหน้าจอ



Exception ນົດກອນການວ່າ ເກີດໂຕຮູບບັນທຶກທີ່ ດີດ ແລ້ວ ຮັນການທຳງານໂປຣແກຣມ



ກໍານົດ ແລ້ວ ເກີດ ໂຕຮູບບັນທຶກ ລວມທຳນຽດກົດນະກາມດາວັນກົດດີນ:

ເພຣ: Exception ນົດກຳຈະເສັ່ນດູ່ ແລ້ວວ່າ ພິກດ້ວຍ...

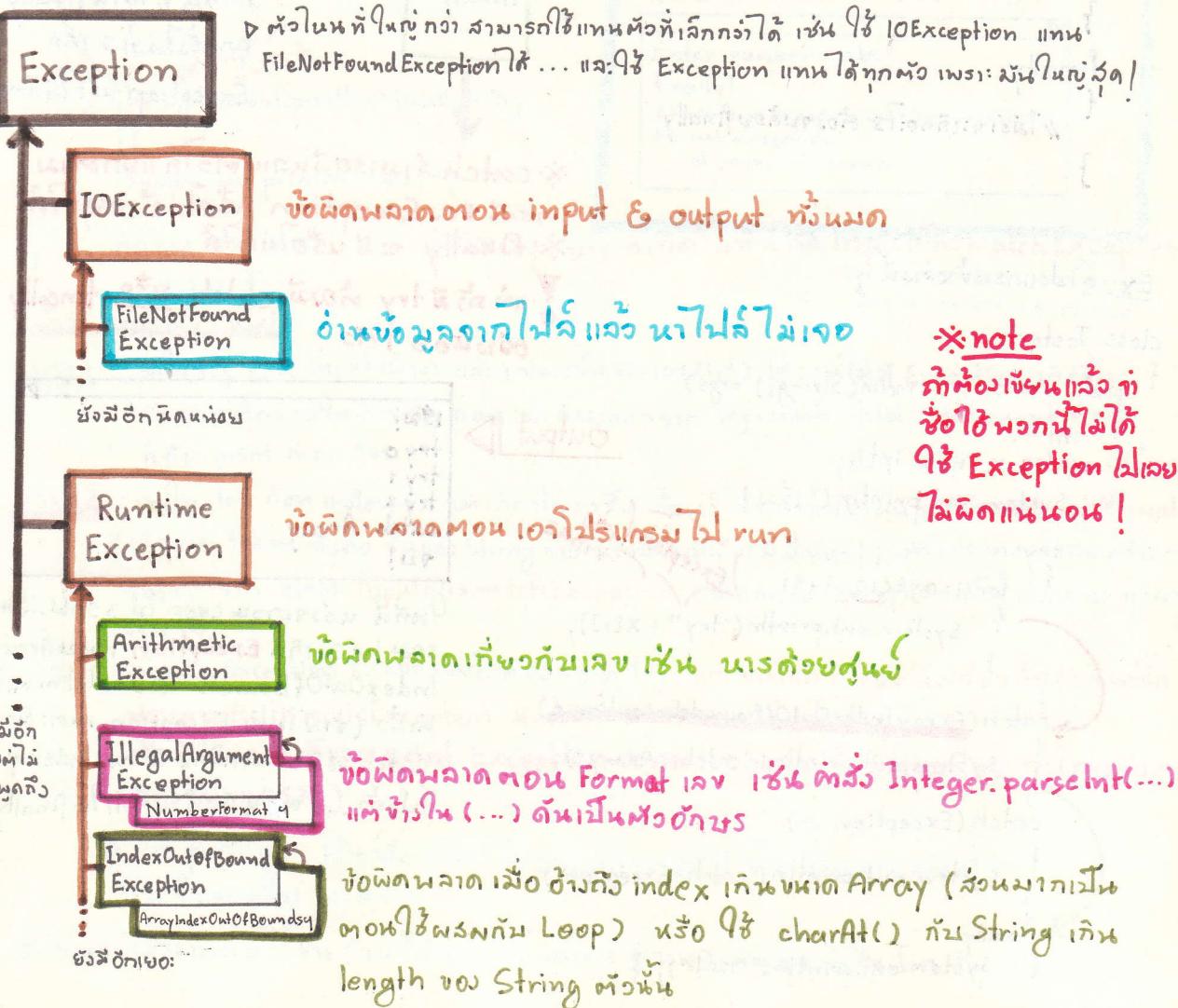
ແນ່ນໃນທີ່ນີ້...

- ① ນົດກວ່າ run main ແລ້ວເກີດ Exception!
- ② ນົດກວ່າ Exception ກໍາເກີດເມື່ອປະເການ Array Index out of bound ສິ່ງດີກໍາເຫຼັກ Array ອີເຊັກຫຸກ
ເກີດຂາດຂອນນີ້
- ③ ເຄີຍ ຖີ່ນີ້ໜ້າຍດາວັນວ່າ ສັກການ ເຮັດ index ຕົວທີ່ 8 ຂອງ Array (ແນວດີວ່າ Array ມັນຫຸ້ນພັນຫາດເລັກ
ກົງ 8 ເຊຍເຮັດໄວ້ໄສ)
- ④ ນົດກັບນີ້ ຂອງຈຸດທີ່ ດີກ ເປັນຊື່ class ທາມດັບຍ ພຣທັດ
→ ເບີຍວ່າ ເວັ້ນເນື້ນ ຂົດລັດຮອຍນີ້... ວັດພົນົດ ພັນເນື້ນສີ່ຄວາ!

* **ArrayIndexOutOfBoundsException** ເມື່ອ class ຈີ ນີ້ທີ່ extends ມາຈາກ class

Exception ໂດຍຈະກໍານົດທີ່ ຕຽບຈັນເຄພະກົດ ເຮັດ index ເກີດຂອນເບຕ

ແລ້ວ Exception ແມ່ນວ່າງ ລວມໄດ້ຢືນ?



ประจําการบํารุง คําถาม ผิดพลาด (Exception)

1. แนวโน้มที่ต้องตรวจสอบ: ส่วนมากมักจะเป็น Runtime Exception โปรแกรมจะไม่ช่วงดับให้ตกร้าว
Exception เพราะ: จะเกิด ความผิดพลาดใดๆ ก็ตาม run เก็บนั้น ก็จะรู้

2. แนวโน้มที่ต้องตรวจสอบ: เป็นกิจกรรมที่ทำงานกัน ณ ไชซ์กอนหน้างานของโปรแกรม เช่น
การเรียกอ่านไฟล์ แนวโน้มที่ต้องตรวจสอบ เช่น

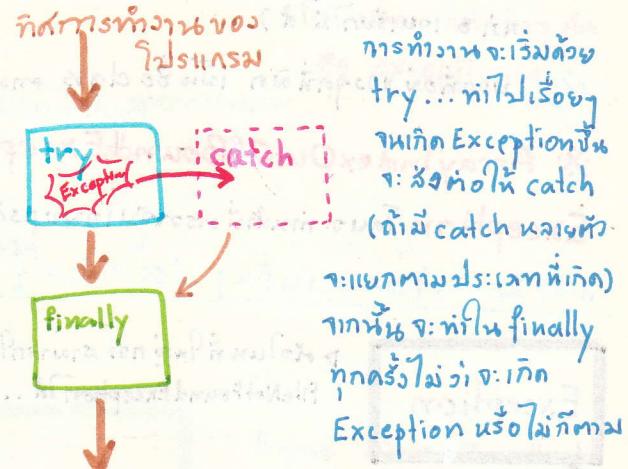
try... catch

เป็นคำสั่งที่ใช้ใน java เวลาผิดพลาด เช่นนี้ เว้นไว้ แนวโน้ม
มันเป็นชุดคำสั่งเดียว if else; หรือ Loop

```
try
{
    // ชุดคำสั่งที่อาจเกิด Exception
}
catch (Exception e)
{
    // ถ้าเกิด Exception ให้ไปที่นี่
}
finally
{
    // ไม่ว่าจะเกิดอะไร ก็ต้อง 실행 finally
}
```

Ex. จําลองโปรแกรมข้างล่างนี้

```
class Tester
{
    public static void main(String[] args)
    {
        int i;
        int[] x = {0, 1, 2};
        System.out.println("เริ่ม!");
        try
        {
            for(i=0; i<10; i++)
            {
                System.out.println("try " + x[i]);
            }
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("catch - Array");
        }
        catch(Exception e)
        {
            System.out.println("catch - Exception");
        }
        finally
        {
            System.out.println("จบ!");
        }
    }
}
```



* catch สามารถจับเฉพาะได้ แยกตาม
กรณี ของ Exception ที่ต้องจับไว้ เจย์ก็ได้
* finally จ: มันหรือไม่ก็ได้

!! แต่ ถ้ามี try ต้องมี catch หรือ finally
อย่างน้อย 1 ทัว

Output

เริ่ม!
try 0
try 1
try 2
catch - Array
จบ!

ให้ที่นี้ หลังจากจบ Loop ที่ 3 ตอนที่ 9
ตอน i = 3 เกิด Exception หมาย Array
IndexOutOfBoundsException เจย์จะไปที่ 9
ก็สั่ง (ทางไม่ที่ Exception ใจ: ไม่มี
catch หรือกันตาก็คง ArrayIndex q
ไปแล้ว...) ตอนนี้สุดท้ายก็ที่ใน finally

finally ដ้วยก็ไม่เกิดขึ้นเมื่อพอนลูดจาก try-catch ก็ต้องทำคำสั่งที่อยู่ต่อจาก finally อยู่ดี *แต่สั่งต่อไปน้ำหนักก็ต้องสั่งทุกประการ เช่น if-else loop หรือพาก $x=y+3$

↳ finally จะไม่แล้วเมื่อเราดักจับ Exception ได้ถูกต้อง... ไม่ว่าการสั่งของหน้าที่แล้ว เราจะ print System.out.println("จบ!"); ขึ้นหนอ แล้วก็จะใน finally ที่ไว output ก็ยังเหมือนเดิม

▶ รันลอง code กันดู 2

```
import java.util.Scanner;
class Tester
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number: ");
        try
        {
            int r = in.nextInt();
        }
        catch (ArithmeticException e)
        {
            System.out.println("ເກີດException!");
        }
        finally
        {
            System.out.println("finally!");
        }
        System.out.println("จบ!");
    }
}
```

(หมายเหตุ: กรณีการรันโปรแกรมจะมีสีเขียว)

output

แบบ 1 : user 9 วันตัวลงมาดิน

แบบ 2: user 9 วันตัวสักช่วงมา

ก็สามารถทำได้

*Exception ที่เกิดขึ้นนี้คือ **InputMismatchException**

ໄປໄວ່ໃຫຍ່ໃນເຊັ່ນ 43 ໂມຣະເກີດ
ຕົ້ນໂດຍໃເກີດເນື້ອໃນ ດຳເນີນຈຳລັງ

॥ຂໍ້ວັນຄົດດໍາໄວ້ນີ້?

॥ຂໍ້ວັນ 1 : 假設ใน try ก้อน input เบิกນາ แล้ว ແປລວມເວັນຕົວເລບ (int) ໄດ້ລົບໄວ່ສັບສົນException เกิดขึ้น
ກອງຈົນສິ່ງທີ່ໃນ finally กອນ แล้วດໍາຍັດຕາຈາກ try-catch ໄປດໍາ statement ທີ່ຕົວ
ກັດ print ຕ່າງ "จบ!"

॥ຂໍ້ວັນ 2 : ກ່າວໃນ try กອນ ແນວດນັກນີ້ ແລ້ວເກີດປົກກົງນາຫຼືກົງ... ປັບງານ ແນວດໃຫຍ່ວ່າ ກ່າວນີ້ **InputMismatchException**
(ເກີດຈາກ token ສັງເກີດ ຊິ້ນຂອງ String ທີ່ເອານວາແປລວມເວັນເລບນີ້ປົກກົງນາ!) ຕົວໂປຣແກຣມ ຂຍ ສຮັກ -
object ຈາກ class **InputMismatchException** (ນີ້ບັນທຶກໃໝ່ Exception ເນື້ອ class ນີ້ ເອກວາ
ສຮັກ obj. ຖີ່)

ເນື່ອເກີດ Exception ສັນແດງ ເລບໂດດອອກຈາກ try ແລ້ວ catch ທີ່ລົບບໍ່ຕົວເຕັ້ງວ່ານີ້ແດ່: ດັນເນີນ
ປະເທກ ArithmeticException ມັນໄດ້ຕ່ອງກັບວ່າ... ແລ້ວເວັນໄວ້?

▷ ຜັກໂປຣແກຣມກີ່ເລີຍ print Exception ອອກມາຮັງເລຍ! (ຕາມສົງລະບຽບ ຈົນການກ່າວງານ
ຂອງໂປຣແກຣມ)

▷ ແລ້ວ ຜັກດອນ! ມັນຍັງ print ພົດຕານ Exception ໄວໃດເພີ້ມເລີນຕ່ອງກຳນົດກົດ
ໃນ finally ດີກດອນ

ຈະເບີນວ່າໄລ້ວ່າເກີດດໍາໄວ້ນີ້ (ແລ້ວໂປຣແກຣມນີ້ການກ່າວງານ) ມັນກີ່ຈະກ່າວໃນ finally ພູມດີ!

printStackTrace នឹងប្រកបដូចជា print ទៅធនធាន 100.

↪ មែន method នៃវត្ថុ class Exception.

ម៉ោង...

```
try
{
    // ចាកចេញវាទំនាក់...
}
catch (Exception e)
{
    e.printStackTrace();
}
```

ការកែតិចង្វារ Exception ក៏ដែលបានបង្ហាញ

```
java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:840)
    at java.util.Scanner.next(Scanner.java:1461)
    at java.util.Scanner.nextInt(Scanner.java:2091)
    at java.util.Scanner.nextInt(Scanner.java:2050)
    at Tester.main (Tester.java:15)
```

នៅលើលទ្ធផល e នា print ឡើង (ខ្លួនជាមិន method toString)

```
try
{
    // ចាកចេញវាទំនាក់...
}
catch (Exception e)
{
    System.out.println(e);
}
```

↑
សង្ឃ printStackTrace នៃការបង្ហាញបញ្ជាក់ថាបានបង្ហាញឡើង

```
java.util.InputMismatchException
```

* សង្ឃភាព តូចរាល់ វិវាទៗ ថា 「Exception in thread "main"」 ដំណឹងពេលវេលាបានជាមួយវារាយការ តាក់ចុង Exception វិញ។

តីវ៉ាការហើយ catch ឬចំណាំតែងទៅ!

នៅលើលទ្ធផល catch ពេលឱកវិនិមីនៅលើលទ្ធផល 10 នៅពេលកែតិចង្វារ Exception បាន មឺនខាងមុខរបស់ខ្លួន...

```
try
{
    int[] x = {0, 1, 2};
    x[100] = 20/0; // ចាកចេញវាទំនាក់...
}
catch (ArrayIndexOutOfBoundsException e)
{
    // ចាកចេញវាទំនាក់...
}
catch (ArithmaticException e)
{
    // ចាកចេញវាទំនាក់...
}
```

ជីវ៉ាការ?

តើចុចរាល់ មុនខាងមុខ ត្រូវបានបង្ហាញឡើងហើយនៅលើលទ្ធផល?

ជីវ៉ាការ!

បាន Exception ដែលទី 2 បាននូវ Arithmatic

? កំណត់វា... អាមេរិកមិនមែនវានៅលើលទ្ធផល

note : គោលទៅ សម្រាប់ក្រុមហ៊ុន កំណត់ពីការបង្ហាញ

និងការបង្ហាញបញ្ជាក់ពេលវេលាបានបង្ហាញឡើង

បានការបង្ហាញលទ្ធផល 20/0 កំណត់ក្នុង x[100]

និង divide by zero ក្នុង Exception

បាននូវរាយការណ៍ទៅ

! ការកែតិចង្វារ តែងតាំង

ការរៀបចំ Exception ។

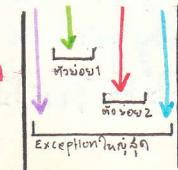
ដែលក្នុងការកែតិចង្វារ ត្រូវបានរៀបចំ

មុនខាងមុខដូចជា Exception ។

ក្នុងការកែតិចង្វារ ត្រូវបានរៀបចំជាអ្នកប្រើប្រាស់

នៅលើលទ្ធផល ក្នុងការកែតិចង្វារ នៅលើលទ្ធផល

តាក់ចុង ... ចាកចេញវាទំនាក់ មុនខាងមុខ: compile វិញ!



• តិចការបង្ហាញ រាយការណ៍ទៅ obj exception
តាក់ចុង នៅលើលទ្ធផល ក្នុងការកែតិចង្វារ នៅលើលទ្ធផល
ទៅក្នុងការកែតិចង្វារ នៅលើលទ្ធផល
តិចការបង្ហាញ នៅលើលទ្ធផល

สร้างขึ้นผ่านคลาสแบบ throws ด้วย throw ชุดคำสั่ง throws น:

→ ถ้าใช้คำสั่ง throw (ที่มี ' throws') อย่างไรจะสร้าง object ที่ทำให้โปรแกรมเกิด exception ขึ้น
Ex. เช่น ลองการป้อน input เข้ามาเป็นเลข(+) เท่านั้น ถ้าไม่ใช่ สิ่งที่ป้อนให้เกิด exception ลง

```
Scanner in = new Scanner(System.in);
Exception exc = new Exception("error!");
int x;
try
{
    x = in.nextInt();
    if(x < 0) throw exc;
}
catch(Exception e)
{
    System.out.println(e.getMessage());
}
```

throw objException;

รูปแบบ
ดำเนิน

ก็จะ exception เกิด ณ print [error!] ทางหน้าจอ

ตอนแรก สร้าง obj. ของ Exception (ชื่อของ exc) เตรียมไว้ก่อน ...

หากนี่น ถ้า x เป็นลบ ให้ throw exc → คือบุญ กับว่า บอกไปร่างระบุว่าเมื่อ exception เกิดขึ้นจริงๆ (แต่ตอนนี้ยังไม่ได้รัน code)

โดยเมื่อ throw แล้ว ลักษณะของ exception → นี่คือ try ไปที่ catch method getMessage() อย่าง print ข้อความ จาก obj Exception

* ถ้า user ใช้เงินหัวรักษาเข้ามา จะถูกจาก try เหลือบกัน แต่ไม่ได้ถูกตัวของ exception ที่เราเขียนคนนั้นร่าง ... เมื่อไหร่สักวัน throw และ สั่ง getMessage() มันไม่ได้เขียน exception หากการ throw ของเรามี print [null] เพราะถือว่าไม่มีการ set ไว้



สรุป Exception

- Exception เป็นความผิดกฎหมายของโปรแกรม ... โปรแกรมจะนำ exception ที่สร้าง obj ตามประสาท exception ที่เกิด ถ้าเขียน try
- ▶ ผู้คนง่ายๆ ก็ถือ คงทำ { ... } ถ้ามี obj. Exception เกิด ให้ catch , รับ obj. นั้นตามประสาทที่ได้ต้องไว้ ...
- ▶ ถ้าเขียน throws ถือว่าถ้ามี obj. Exception เกิด ให้สร้างใน class ที่ถูก " โฆษณา " นั้น
- ▶ ถ้าเขียน throw ... ต้องไม่ได้รับ exception ที่มีแต่เขียนหัวทำให้เกิด exception ที่ต้อง try-catch หรือ throws มาตรวจสอบให้ได้ !

Miku Says:
Java is my
LIFE >[<"
เช่นไร?



Hatsune Miku from Vocaloid 2 ..

Illust by: ~Mudmeen~

นิตย์พัฒนา
เรื่องนี้ถูกไปเมื่อ 0_07

อ่าน เขียน File

読と書 ファイル

การอ่านข้อมูล / เขียน File

↳ การอ่านข้อมูล + เขียน File จะทำผ่าน obj. ตัว class **FileReader** และ **PrintWriter**
 ※ ก็ตติกาใน `java.io` จะใช้ดังนี้ `import java.io.FileReader; import java.io.PrintWriter;`

FileReader ชื่อ `obj = new FileReader("ชื่อไฟล์ที่ต้องการอ่าน.นามสกุล");`
PrintWriter ชื่อ `obj = new PrintWriter("ชื่อไฟล์ที่ต้องการเขียน.นามสกุล");`

FileReader

เวลาเราเขียนบันทึกไว้ หรือนำไฟล์จากที่ไหน โดยทั่วไปใช้ร่วมกับ **Scanner** ปกติแล้วเวลาใช้ **Scanner** จะใช้แบบ

`Scanner in = new Scanner(System.in);`

สร้าง **Scanner** ผ่านช่องทางดังนี้
 ให้ไปป้อนข้อมูลจากที่ไหน

เป็น obj. ตัวนี้ที่บอกให้ **Scanner** ดึง buffer (user input มาจาก keyboard)

ที่นี่ เราจะสั่งให้อ่านจากไฟล์ในเครื่อง เราแก้ obj. **FileReader** ยัดเข้าไปแทน parameter ของ **Scanner** แทน **System.in** ...

`FileReader reader = new FileReader("sth.txt");`

`Scanner in = new Scanner(reader);`



เมื่อเราสร้างจาก obj. **FileReader** ให้ตั้งชื่อ (เวลาจะยัดให้ **Scanner** เพื่ออ่าน **Scanner** ให้ไปอ่าน (scan) ที่ไหนก็ตามที่หัวที่แล้ว รันสั่งบนบล็อกๆ กัน)

`Scanner in = new Scanner(new FileReader("sth.txt"));`



note file directory ที่ต้องการใช้ แนะนำตัวอย่างไปเลยก็ได้ กรณีที่ไม่

อยู่บน local ที่กันตัวโปรแกรม เช่น "D:\\Aruru\\Test\\sth.txt"

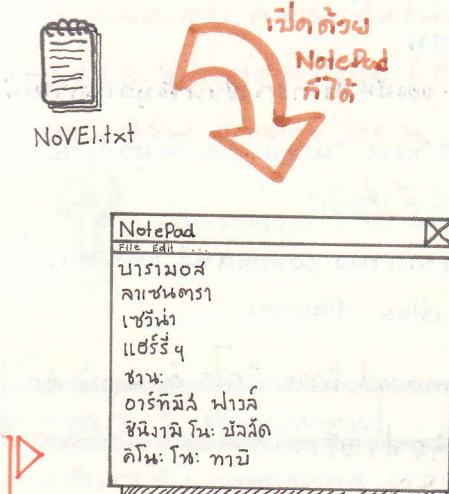
แล้วหัวใจสำคัญอีก 1 ตัว? : การใช้ '\\' หนึ่งตัวความ (เรียกว่า escape char)

เวลาเราสั่งตัวสัมภาระ เช่น \t ดังเช่นบรรทัดใหม่ \t หรือ tab

ส่วน \| จะหมายถึง \\

❗ ซึ่ง+ทั้งบุคคล มนพ์ตัวในสูตรแล้วก็จะต่างกัน ไม่เหมือนซึ่งส่วนมาก

เริ่มต้นไฟล์



output ก็จะออกมาเหมือนกับที่เราตอนนี้
จาก notepad แต่เป็น 90 DOS ซึ่งดีๆ

สมมุติว่าเราต้องการจะให้โปรแกรม
อ่านข้อมูลทั้งหมดใน text file ที่ชื่อว่า
"Novel.txt" และ print ลงหน้าจอ

```
import java.io.FileReader;
import java.util.Scanner;
class FileTest
{
    public static void main(String[] args) throws Exception
    {
        Scanner in = new Scanner(new FileReader("novel.txt"));
        while(in.hasNextLine())
        {
            String x = in.nextLine();
            System.out.println(x);
        }
        in.close();
    }
}
```

รับรู้ไฟล์

ตัวเล็กๆ ใช้รับตัวเข้ากัน
เมื่อต้องการ ผูกพัน addem
ของ My computer ห้อง
ทำงาน หรือตั้งค่าไฟล์

method [hasNextLine()] เป็นฟังก์ชันที่ตรวจสอบว่าไฟล์ยังมีข้อมูล剩餘ให้อ่านหรือเปล่า,
剩餘ในที่นี้ หมายถึง เหลืออีกหนึ่งบรรทัด ใช้งานมาใช้กับ Loop while หมายความว่า "เมื่อเหลือ¹
ข้อมูลให้อ่าน ให้อ่านต่อไป"

method [nextLine()] เนื้องอกการ scan จาก keyboard แต่ใช้กับ scan จาก
file แทน (ใช้ nextLine เนื่องจาก อ่านที่ห้องบรรทัด = ตัดคำที่การ **[enter]** ไม่ใช่ตัดที่ space)

method [close()] ปิดไฟล์เมื่อต้องจบแล้ว (ก่อนอ่าน จดจำ ไว้ต้องปิดก็ได้)

* Exception ไฟล์ข้ามรักแก้ไข!

Java บังคับว่า เมื่อมีการ อ่าน/เขียนไฟล์ ต้องมีการ ป้องกัน Exception ด้วย
(กลับไปดูหน้า 44 ได้!) จะใช้ throws หรือ try...catch ที่ใช้ไปก็ได้ เสียงซักถาม

โดย Exception แบบเจ้าๆ ของหัวหน้าไฟล์คือ **[FileNotFoundException]** แต่ถ้า:
บุคคลที่ไม่รู้ว่า ต้อง **Exception** ลับก็ได้

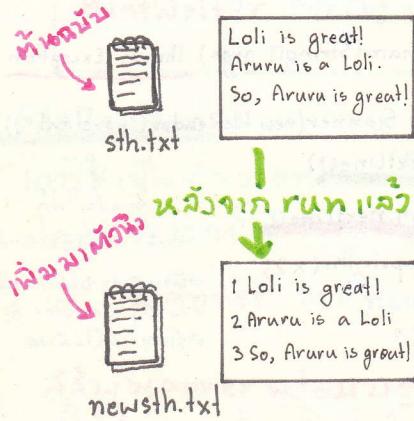
nextLine() method นี้เมื่อ 93% แล้ว (หมายความว่า อ่าน ข้อมูลที่บรรทัดแล้ว)
โปรแกรมจะ ส่งให้ ข้อมูลในบรรทัดต่อไปหนึ่ง (แต่ข้อมูลไม่ต้องอ่าน: เตรียม
รอไว้ก่อน)... เนื่องจาก 93% ของ Loop เดียว ก็สามารถ scan ตัวยังดีสัก
เดียว ก็ได้ อาจใช้ **String** ดูตามดูจะ: คำกันได้!

เขียนไฟล์ด้วย PrintWriter

การใช้ PrintWriter จะคล้ายกับการใช้ System.out.print แต่จะใช้เมื่อ output ออกทางหน้าจอ แต่จะไปบนไฟล์แทนการเขียนที่ไฟล์แทน

การเขียนไฟล์ ไม่ต้องยุ่งกับ Scanner เนื่องจากตอนอ่าน obj. ของมันสามารถเขียนข้อมูลเองได้

Ex. เช่นนี้โปรแกรม copy ไฟล์ โดยไฟล์ที่ถูก copy จะมีชื่อ "new-" นำหน้า (เช่น ข้อมูลชั้วโมง: เนื้อหาเดิมที่บล็อกบอร์ดที่เนื่องมาช่วงนี้...) ก็คงจะมีสิ่ง?



- โดยใช้ไฟล์นี้รับมาจากการ command line แบบนี้

java CopyFile.class filename

? วงเล็บ ... command line ต้องใส่ชื่อไฟล์
ก็ได้รึเปล่าแล้ว ต่างๆ กันอย่าง...

[] จุดนี้ไม่ได้บอกว่า class น้ำใจจริงแต่บัง
บังเด้งๆ มากใน command line ว่าจะ run
โปรแกรมชื่อ Copyfile ๆ!

```
import java.io.FileReader;
import java.io.PrintWriter;
import java.io.FileNotFoundException;
import java.util.Scanner;
```

แม้ว่าจะ try-catch อยู่แล้ว ก็ต้องมี throws เพราะ: เวลาสร้าง
obj. PrintWriter กับ Scanner(FileReader) ชื่อ main อยู่ใน try
▶ หากผิดพลาดไม่สามารถ try ไม่ได้ดี ต้องเตือนผู้ใช้
หากใน try หรือใน finally จะใช้รัก obj. พวกนี้!!

```
class Copyfile
{ public static void main(String[] args) throws FileNotFoundException
{
```

PrintWriter file = new PrintWriter ("new" + args[0] + ".txt");

Scanner in = new Scanner (new FileReader (args[0] + ".txt")); หมายเหตุ มากินน้ำ กดปุ่ม

int number = 1;

```
try { while (in.hasNextLine())
{ file.println (number + in.nextLine());
number++;
}
```

ได้รับเพิ่มเติมมา 100 (เรียบร้อย)

รอดูความที่อ่านมาจากไฟล์ต้น

เขียนໄດ້ສໍາເລັດແລ້ວ

copy เสร็จแล้ว

ก็ໄວ່ເປັນ print ໄດ້

```
} catch (Exception e)
{ System.out.println ("copy ไม่ผ่าน อ: ");
```

}

finally

```
{ in.close();
file.close(); }
```

จบวันนี้ ก็ปิดไฟล์ต่อง!



Alert กันที่หัวร่าง `obj.getWriter()`, file เป็นชื่อกลางสร้างขึ้นมา
รองให้ได้เลย... สำหรับนั้นมีอยู่แล้ว (ชื่อตัวอักษร) มันคือส่วนของ
ที่บันดาลใจ กล่าว เช่น `FileWriter` เป็น... เพราะวันนี้ ก็จะเป็น `File` แต่ก็ต้องใช้ชื่อเดียวกัน
เพียงลง `FileWriter` ตัวเดียวทัน (ด้านบนและเขียนลงในไฟล์เดียว) จะเห็นว่าด้านในนี้มีร่องก่อนแล้ว
ต้องสร้าง `PrintWriter` นั่นเอง!



แล้วต่อมาไว้ก่อนแล้วจะเด้อไปเก็บที่ไหน

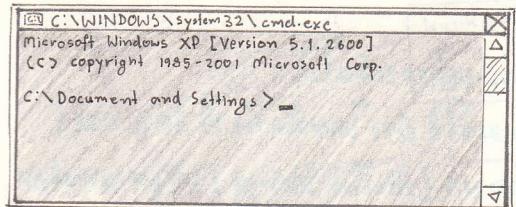
ก็ต่อ `Array` ของ `String` มาเก็บไว้ แล้วตอนเขียนนี้ก็ต่อ `Array` ตัวเนี้ย
มาวน Loop เขียน!
ต่อ

cmd & Argument

การทำงานกับ command prompt กับ args ของโปรแกรม

- > การ run program ที่เขียนด้วย `java` ต้องเนื้องอกในโปรแกรมที่เขียนน้ำหนัก .EXE กับ run ได้เลย... เราต้องเรียงกัน `java` กับ class ของเราเขียนให้ (ส่วน `javac` เอามาให้ compile) ถูกต้องหน้า 2
- > การ `compile .java` ให้ `.class` แล้ว run ปกติ เราจะไม่ต้องรู้ เพราะ `EditPlus` ทำให้เรา
- > ถ้าต้องการ command prompt จะทำอย่างไร?

↳ ตามห้องเรียน กด + R ชื่อ (ตัวเริ่ม Run...) แล้วพิมพ์ cmd →



ตอนนี้ก็ต้องติดๆ เมื่อนะ: cmd



อื้... แล้วไงเดี๋ยวจะเขียนไว้, เรื่อง ก็เข้าไปในไฟล์เดียวกันที่เก็บไว้ก่อน

สมมุติว่าเก็บที่

C:\TA-Java\Test.java

ก็ต้องกลับไปที่ TA-Java ก่อนทุกอย่างนั้น > cd.. กลับ

C:\> Document and Settings> cd.. 9ดูเพื่อบอกกลับ(Back)

c:\>> cd TA-Java cd หมายความ ชื่อโฟลเดอร์ เป็นการเข้าไปในโฟลเดอร์นั้น

c:\> TA-Java> javac Test.java เรียก javac 9ดู compile Test

c:\> TA-Java> java Test.class sht1 sht2 sht3 ...

เรียก java 9ดู run 9ดูพากันนี่นะ: args

note ต้องไฟล์เดียวกัน space
เขียน My Document 9ดู

> cd My Document 9ดูติดๆ

9ดู My * 9ดู

▶ ตัวที่อยู่ต่อหัวเป็น .class ต้องมีตัวอักษรเป็น `args[0]`, `args[1]`, `args[2]` 7ดูเรื่องๆ

ภาษาโปรแกรมเราสั่ง...

```
for(String a: args)
{ System.out.println(a); }
```

9ดู Loop for 9ดูนี่ติดๆ
0ดู นี่ 15
(foreach
Loop)

output ▶ sth1
sth2
sth3

ในกรณีแบบตัวอย่างนี้ ถ้าเราพิมพ์
`args[3]` จะเกิด Exception เพราะมี
การตั้ง `args` แค่ [0], [1], [2] !



Comment: ว่าดายังลาก
ร้าวมาก!?

Illust by: ~Mudmeen~TWT

cmd, args / คำศัพด์ 72.

リーカーシャン Recursion

— ทฤษฎี Logic —

↳ Recursion (นั้น Recursive) เมื่อ ซึ่งมีการเรียก method ที่เรียกว่า วนลูป ... ซึ่งมันก็วนซ้ำ
จะต้องใช้ sense ของคนเขียนมาพยายาม สิ่งที่เขียนยอดกันได้

ข้อสังเกตของ method ที่มีการเรียก Recursion

- ▷ ทำงานเป็นลำดับ
- ▷ เมื่อการ sum 0: ໄฉซักกบบ่งเพิ่มเรื่อยๆ
- ▷ ต้องการใช้ชี้สู่ว่าบุรุษรอบที่แล้ว

ตัวอย่างเช่น การหาตัว factorial หรือ revert String แก้: การ solve "Tower of Hanoi"

```
public static int ชื่омethod(int n)
{
    I. if(n==...) { ...return 0: ໄฉซักกบบ่ง... }
    else
        II. return ชื่омethod(n-1);
```

◀ ดูการเรียกที่โปรแกรม จะไม่มีรูปแบบ
ตายตัว แต่ ออกแบบที่พอดีรูปไว้
ของ method recursion เป็น
แบบนี้

Step ในการที่ Recursion

- I. หากที่ที่เล็กที่สุด และ return ค่าทุกที่ 0: ໄฉน้ำดองกลับไป
- II. ถ้าไม่ใช่กรณี เล็กที่สุดให้ เรียก method นั้นมาใช้ อัง กดยังส่วนต่อไป วนไป
(ก้าวเป็นตัวเลข) หรือ substring บ้าง ส่วน 0 ถูกมา (ก้าวเป็น String) เพื่อ parameter

?? ... !? ไปถูกตัวบ่งชี้ ?

Ex! เรียน method "fac" ที่น้ำดอง factorial แบบ recursion

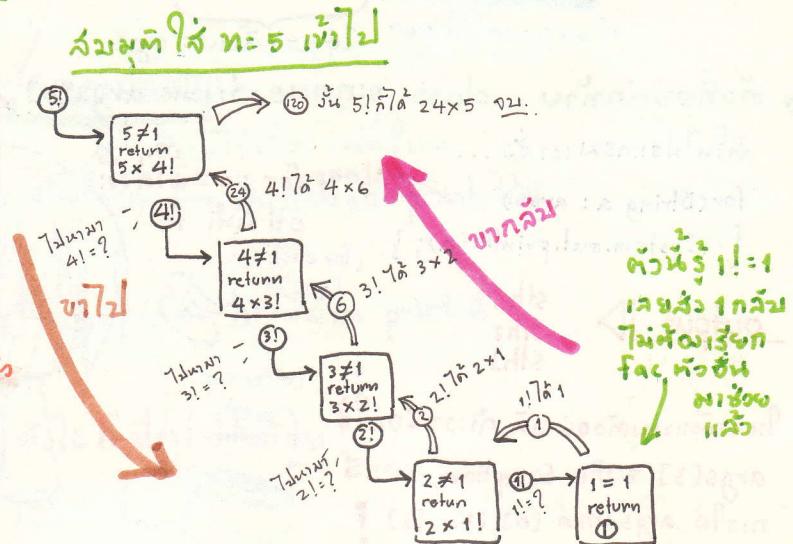
I. กรณี เล็กที่สุดของ fac (!) คือ $1! = 1$ ใช้แทนค่าทุกที่ ที่ 0: ໄฉน้ำดอง return (ต้องให้เข้ามายัง 0!)

II. $n! = n \times (n-1) \times (n-2) \dots \times 1 = n \times (n-1)!$ หมายความ $(n-1)! \rightarrow (n-1) \times (n-2)!$

เน้นว่า: ว่า วนลูป 1 ลงเรื่อยๆ $\rightarrow (n-1)!$

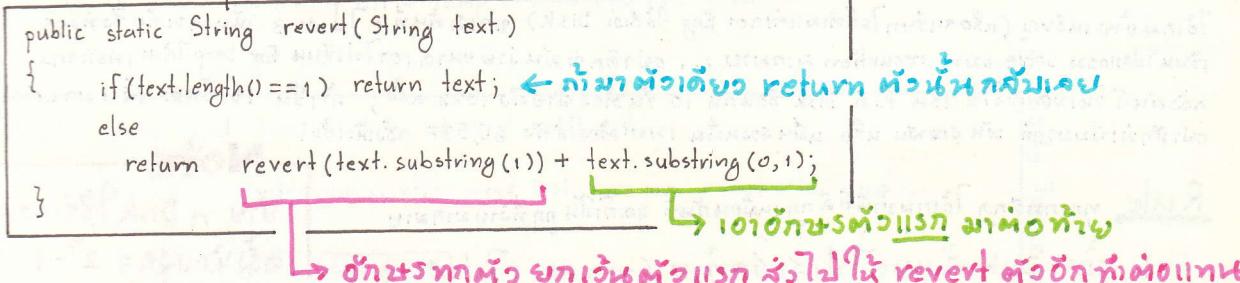
```
public static int fac(int n)
{
    if(n==1) return 1;
    else
        return n * fac(n-1);
```

ทำงานเดลล์ เช่น ลืม fac คุณอยู่แล้ว
มาเขียนกันติด

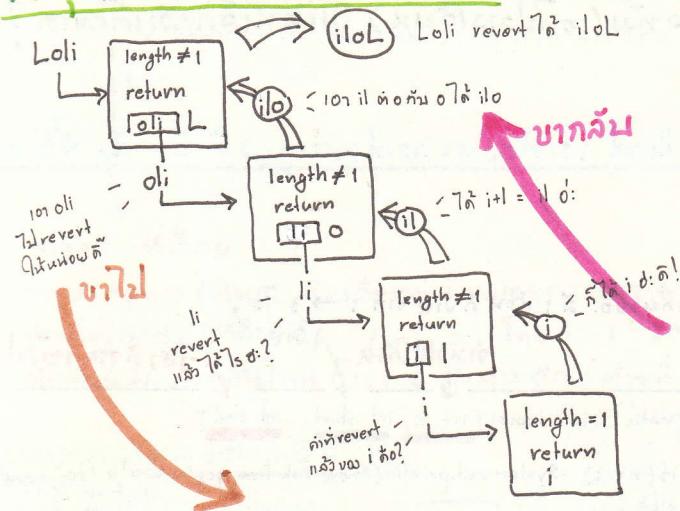


Ex. 2 เรียน method revert ดี ใช้สิ่ง "Aruru" ที่ล return เป็น "ururA"

- I. กรณีลึกสุด คือ มีตัวอักษรหนึ่งตัวเดียว ถ้า revert เลิกก็ต้อง บากซ์หัวต้นนั้นแทน: return ตัวเดิมกลับไปเลย
- II. กรณีตัวมากกว่า เช่น abcdef → ให้เป็น a █ : ถ้า revert แล้วก็ต้อง █ a ให้มีชัย รันเวก์ มา revert ก็ใน █ ตัวที่ 2, bcdef → ให้เป็น b █ : ถ้า revert แล้วก็ต้อง █ b ต่อไป... วันนี้ ก็ทำต่อไปเรื่อยๆ นี่叫做: recursion



สมมุติว่า สิ่งที่ว่า "Loli" เข้าไป



เมื่อไร... ผ่านมา 2 ข้อ รู้สึกว่างานยัง? (ยังไม่เห็น นี่ช่องมีได้ดี ... ใจ Recursion
มีความน้อยๆ อยู่!)



method ที่สามารถเรียก recursion ได้สองมุม ๑: เขียนให้หาตัวเองโดย
วิธีธรรมชาติ เช่น ใช้ for Loop ๒: ... ช่วงมันว่าอยู่กว่า 1 รอบ: ถ้า ๓: เขียน
แบบ for

- ถ้าโจทย์ໄลก์กำหนดว่าต้องใช้ recursion ก็ติดแบบ for ไปก็ได้จะดี



: ข้อเสียของ Recursion คือ มันต้องหาก โจภารผิด เมื่อ: ทำงานเวลา-
ที่慢 เนื่องจากมีลักษณะ mem. (เพราะต้องเรียก method หลายครั้ง) รับมันแล้วต้อง
รับมันเข้าไป ?

► เขียนแบบวน erte ๆ , บางที่ recursive method ก็มีประโยชน์ส่วนหนึ่ง
โจทย์นิวยอร์ก (โจภารเรื่องตั้งติดกัน) เช่น Tower of Hanoi เป็นต้น

10 วิ่ง... ในที่สุดก็มาถึง Tower of Hanoi ห้ามลืมหูหิ้วที่สุด เนื่องจาก: แปลงตัวจาก 1 ชั้น 1 ด้าน เป็น 10 ชั้น 10 ด้าน ซึ่งจำเป็นมาก (แต่เราใช้หน่วย 55+)

タワー of Hanoi ハノイ

ภาษาไทยใช้ได้: ปริศนาหอกอง แห่ง ฮานอย (ค้น Google)

ได้เกณฑ์ชี้วิธีเรียงกู (หรือจะเรียกว่า Disk) หากเลือกต้นที่ 1 ไป剩 3 เนื่องจาก: ต้องสับหัว ก่อนโน้มร่างรอม ลงบน วนน์ recursion มากน้อย ... อย่างติดต่อ วนน์ง่ายๆ หาด 10 ไปซึ่งนี่ for 100 ได้: ทราบ: นาน ตอบสนอง กว่า เกินไป เช่น ถ้า Disk ชั้นก้อน 10 ชั้นต้อง ใช้ชั้น 1023 ครั้ง, ถ้าชั้น 16 ชั้นจะ: เนื่องมา วนน์ อย่างนี้ กว่า เพิ่มน้ำด้วย พื้น ส่องพัน หรือ หมื่น ส่องพัน 朋: ต้องใช้ก้อน 65,535 ครั้งเชียว!

Rule ทุกคนฝึก ได้เกณฑ์ชี้วิธีเรียงกู เหล่านี้ กัน และ ก็ฝึก กูที่ง่ายมากน้อย

- basic 1. ย้าย Disk กับห่วงต่อไปเรื่อยๆ ต่อไป (จาก เสา 1 ไป เสา 3)
มาก 2. ย้ายได้ครั้งละ Disk ต่อเนื่อง

Note

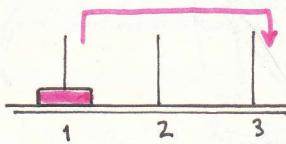
ย้าย ท Disk 9 ชั้น วนน์
ครั้นน์บวก 1 = $2^n - 1$

→ 3. Disk ที่ใหญ่กว่า น้ำเงิน 10 ไปวางทับ (10 ไปวางไว้ข้างหน้า) Disk กับเล็กกว่า เด้งๆ ๆ !

จัดชั้นเรียง:
ฐานที่ดิน

▶ มาเข้าเรื่องกัน สำหรับ recursive method เราต้องหา กฎกติกา 2 ข้อ I. II. กันหน้า ชีวิต...!? รันกราฟ I. ดีอ...

I. กติกา กฎ กติกา ของ Disk อยู่ดังเดิมไว้



ที่นี่เรา ก็ได้ code ห้องแรกมา แล้ว: → ก้าวแรกแล้ว ให้เรียง สำหรับ เก็บไว้ ชี้ยังจาก start ไป end

< basic มากน้อย: มี 1 ก้อน ก็ย้าย จาก 1 → 3 ใจ,

จึงมี:

ตัวหนัง Disk / ใจที่เริ่ม

เริ่มที่: ชี้ยังไป

```
public static void hanoi (int n, int start, int end)
{
    if (n<=1) System.out.println("move Disk from peg " + start + " to " + end);
    else
    {
        ← เดิมที่อยู่
    }
}
```

* 9 ชั้น void ไม่รู้ใน method print
จะ solve 100

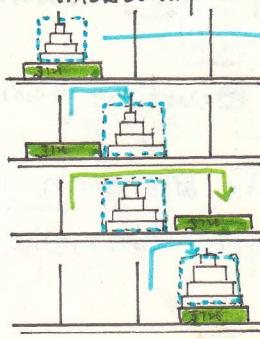
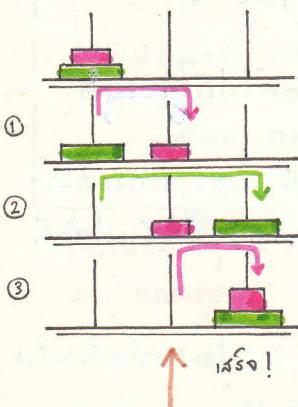
? แล้ว วนน์เป็น recursion ตรงไหนกัน

ในรูป II. ต้องมาว่าเดือน Tower ที่กันซักหน่อย

- ถ้า: ย้าย 2 Disk ทำไง?

2 ชั้น ทำได้ใน 3 บันทุ: , 4 บันทุ: , 10 บันทุ: ?

↳ ถ้าเริ่มชี้ช่องๆ ลองมาดูว่า วนน์ Recursion กัน สำหรับ ท ชั้นก้อน



ถ้ามี 2 ชั้น สำหรับ $(n-1)$ ชั้น:

▶ มองว่า เรายังคงย้าย ได้ $n-1$ ก้อนชั้นนี้ ไปวางที่ เสา 2 ก่อน แล้ว ก่อนย้าย ฐาน 1 ก้อนไป เสา 3 แล้วด้วย 1 ครั้ง ก่อนที่ ย้าย ชั้น 2 ย้ายตามมาต่อไป:

ปัญหาเดียว ได้ $n-1$ ก้อนจะ: ย้ายชั้น 2 กลับไป

↳ ก็เรียง method ตัวเองมา ช่วย แก้ ปัญหานี้!
แก้ ปัญหานี้!



สรุป: วิธีการ การย้าย n Disk ต้องทำไงบ้าง?

- step1:** ย้าย $n-1$ ก้อน ชิ้นบนไปไว้ที่ temp (หมาดกึ่งใส่ที่ไม่ได้ใช้ ให้กับที่เดินทาง 2 ... ก็จะ จาก 1 \rightarrow 3 อะ)
- step2:** ย้าย ก้อนใหญ่สุด จาก start \rightarrow end (ให้กับที่เดินทาง 1 \rightarrow 3) } อยู่รูปแบบนี้จะช่วยให้เข้าใจง่ายมาก!
- step3:** ย้าย $n-1$ ก้อน ชิ้นบน temp (2) กลับ end (3) ... จบ!

源代码 II. 7 ไปเขียน code เต็มที่ๆ กัน...

```
public static void hanoi(int n, int start, int end)
{
    if(n<=1) System.out.println("move Disk from peg " + start + "to " + end);
    else
    {
        int temp = 6 - start - end;
        hanoi(n-1, start, temp); // ย้าย n-1 ก้อน จาก start  $\rightarrow$  temp
        hanoi(1, start, end); // ย้าย 1 ก้อน (ฐาน) จาก start  $\rightarrow$  end
        hanoi(n-1, temp, end); // ย้าย n-1 ก้อน ที่ใน temp  $\rightarrow$  start
    }
}
```

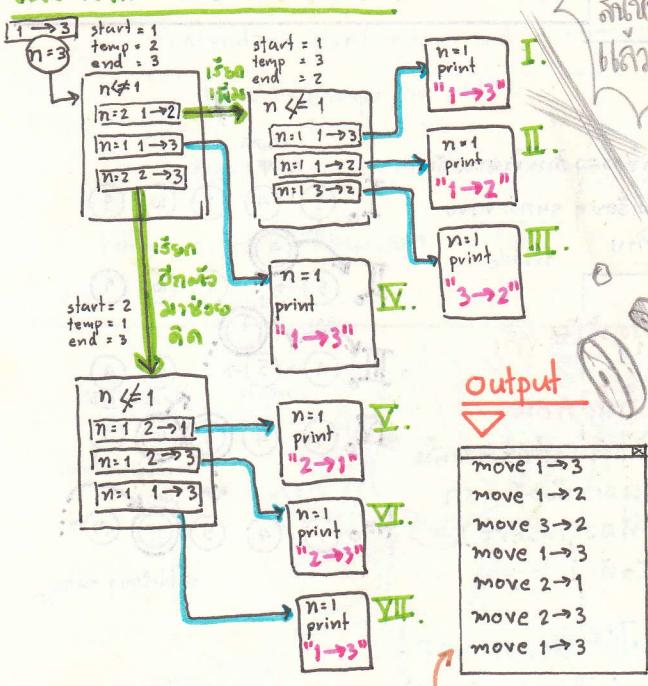
Code Tower of Hanoi หมายความว่า...

บรรทัดเดียวกันๆ กัน?

กรณี temp (ต้นเส้นที่เปลี่ยนไป) นำไปยังการ solve 6 จานบน start n: end

\rightarrow มีต้นเส้น 3 หัวสี 0, 1, 2, 3 โจทย์ $1+2+3=6 \therefore$ ถ้ามีเส้น 0 ไป 2 หัวบนยกครึ่ง
หัวที่เปลี่ยนต้องหันไปทาง ก้าว 6 หัวลง ยก 2 หัวที่ 5 หัวไป!

แผนภูมิการ solve 3 Disk



Output

```

move 1->3
move 1->2
move 3->2
move 1->3
move 2->1
move 2->3
move 1->3

```

output ของภาษา C สำหรับการ
เรียกฟังก์ชัน method

Hanói ผัง กะ ล่าย!

Sayonara Setsubou Java-Tan T.T

comment: ข้อควรระวังในการพิจารณาเรื่องนี้คือเมื่อ:

Sort & Search

並と探

- หมวด Logic -

→ การ sort (เรียง) และ search (ค้นหา) เป็น operation ที่ใช้กับ Array

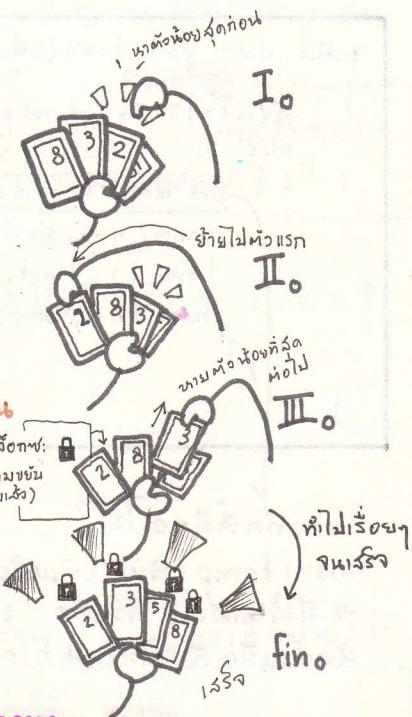
Sort 並

→ การเรียงข้อมูลกันง่ายๆ ใน Array จากน้อยไปมาก เมื่อ 2 วิธี (เรียก ก็เรียน) สืบ...

1. Selection Sort
2. Insertion Sort

Selection Sort

- การ sort แบบนี้ ให้นึกถึง เวลาเรียงไพ่ บนมือ เราจะนำตัวที่ห้ามสุด แล้วข้างหลัง 0 บู๊ กางดำเนินเข้ามาสุด แล้วหันจากนั้น เราจะใช้วิธี กับตัวแรกซึ่ง (ตัวเดียว ก็ลังกุนแข็ง มันดี) แล้วก็นำตัวห้องสุด (ที่แล้ว) ต่อ... พอกด ก็ตามไปได้ พลิกเข้ามาสุด ต่อจาก ช่วงที่เรา sort ไปแล้ว (สังข์ช่วงที่โคน ก็ไม่แล้ว) ทำอย่างไรไปเรื่อยๆ ตัวที่โคนจะถูกกีดขวางชั่วขณะ จนครบทุกตัว ส่วน sort แล้วจะมี!

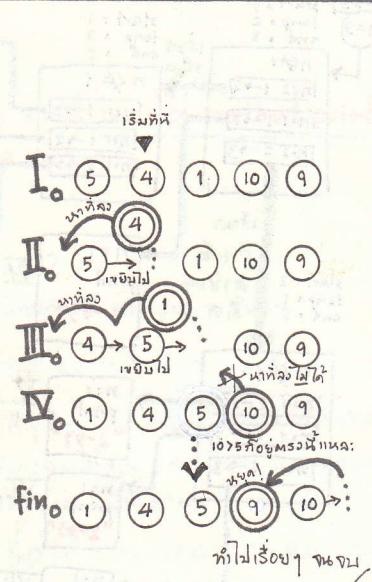


```
public static void selectionSort(int[] a)
{
    for(int i=0; i < a.length-1; i++)
        {
            int min = i;
            for(int j = i+1; j < a.length; j++)
                {
                    if(a[j] < a[min]) min = j;
                }
            int temp = a[min];
            a[min] = a[i];
            a[i] = temp;
        }
}
```

※ ใช้การ swap กับเพื่อ: ไม่วันเดียว ค่าที่เก็บใน a[จ้งหาย] เอาไว้ทางที่เก็บใหม่ให้มัน

Insertion Sort

- การ Sort แบบนี้จะเริ่มที่ตัวที่ 2 ($index = 1$) และให้ลืมว่าตัวนี้ช่วยช่วยของมัน (เรื่องนี้ถึงถ้วนจะยังห้องกว่า 0 บู๊ ก็จะเขียน index ลดลงไปเรื่อยๆ จนกว่าจะ 0 ซึ่งห้องไว้) และทำแบบนี้ไปเรื่อยๆ จนกว่าจะถูกห้องทีเดียว



```
public static void insertionSort(int[] a)
{
    for(int i=1; i < a.length; i++)
        {
            int temp = a[i];
            int j=i;
            while(j>0 && a[j-1] > temp)
                {
                    a[j] = a[j-1];
                    j--;
                }
            a[j] = temp;
        }
}
```

→ เริ่มที่ 1 จะที่ตัวสุดท้าย
→ เก็บตัวที่ index นั้นไว้ก่อน
→ นำตัวที่ index นั้นไปห้อง
→ ขยับกลับมาถึงตัวแรก
→ เริ่มน้ำที่ลง (insert)
→ ใช้ตัวที่ j
→ 1 ถ้าตัวที่ i ที่เก็บไว้ตอนแรกนั้น
→ นั้นมากกว่า "insertion"

Search 探

> การหาข้อมูลที่เก็บอยู่ใน Array ว่า data ตัวนั้นเก็บอยู่ index ที่เท่าไร
ก็ search ก็หาไม่เจอ จะ return -1

▶ ปกติ ก่อเรื่อง Search เราก็จะ วนหาตั้งแต่หัวไปจนกว่าจะเจอ หรือไม่เจอ

```
public static int search(int[] a, int value)
```

```
{ int i = 0;
```

```
while(i < a.length && a[i] != value)
```

```
{ i++;
```

```
if(i >= a.length) return -1;
```

```
return i;
```

```
}
```

return ตัวนั้นคือ

หัวใจ

→ แล้วก็ Array ตัวนี้ sort อยู่แล้ววะ? หมายความว่าต้องเสียเวลา จึงมาใช้ binary search เกิดขึ้น

▶ Binary Search

↳ ถ้าหาก search สำหรับ Array ที่ sort มาแล้ว เราเนี่ย! ... binary search :: search แบบแบ่งครึ่ง naïve search

จะดีกว่า จ: 170 สำหรับตัวที่ไม่sort ข้อบัญญัติหนึ่งๆ จึงต้องตัดทิ้งไปเลย

ไม่ต้องสนใจอีก

```
public static int binarySearch(int[] a, int value)
```

```
{ int low = 0; ] ขอนำเข้าในการวนloop
```

```
int high = a.length - 1; ] ตัวเดียวเริ่มสิ้นหัวสุดท้าย
```

```
while(low <= high)
```

```
{ int mid = (low + high) / 2; → หาจุดกึ่งกลาง (middle index)
```

```
if(a[mid] == value) return mid;
```

```
else if(a[mid] < value) low = mid + 1;
```

```
else
```

```
high = mid - 1; // เมื่อ a[mid] > value
```

```
}
```

return -1; → ตัวนี้ไม่เจอก็

```
}
```

Search for: 4

4 < 6 อยู่ช่องไหน

→ 4 > 3 อยู่ช่องไหน 7 > 6 อยู่ช่องไหน ไม่เจอก็

I. 1 2 3 4 5 | 6 7 8 9 10

II. 1 2 | 3 4 5 | 6 7 8 9 10

fin. 1 2 3 4 5 6 7 8 9 10

* ประเมินเวลาตามกoeff.

ห้าม 3 รอบก็เจอก็แล้ว!

return 3

ก็ ทำให้แน่นอนว่าหัวใจ

value ที่ return index ตอนนั้นนั้นก็เป็นกลาง

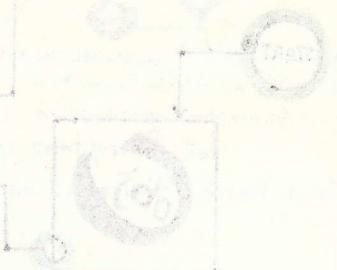
→ ลองเก็ง ข้อมูลการวน

ครึ่งที่ไม่ลับ ข้อบัญญัติหนึ่งๆ ดังนี้ กูกตัดหัวใจ

ไม่นานในช่วงนั้นอีกแล้ว



เงื่อนไขอีก 1?



LinkedList

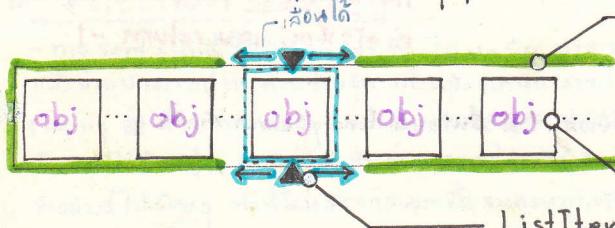
LinkedList เป็นเรื่องของ Data Structure (ลักษณะ: คล้าย กับ ArrayList (หน้า 17) ต่อ 1 ปี "list" 10 วิวิคก์ บุคคล object หรือก็คือ Data อีกแบบ...)

▷ ล 2 class ที่ใช้ในเรื่องนี้ดัง

LinkedList กับ **list** เก็บ Data ไม่เรียงลำดับ:

ListIterator เป็นตัวคล้าย pointer ซึ่งบอกว่าเรากำลังจัดการกับ Data ตัวไหนใน LinkedList อยู่

Linked List: เป็น class ที่เอาไว้เก็บ Data (คล้าย ArrayList)



Object: Data ที่เก็บใน Linked list

obj ของ ListIterator เป็นแต่ pointer ที่เก็บหัว Data ของหัวหน้าของ LinkedList มาก่อน

ListIterator: เป็นตัวชิ้นเดียว (pointer) ว่าเรากำลังจัดการกับ obj. 什么呢 9 ณ List อยู่... ติดลบฯ ซึ่งเป็นเวลาระบบที่ Word หรือ NotePad อีก หัวหน้า ก็จะถูกเปลี่ยนเป็นหัวหน้า Data, เก็บช่อง ก็จะเป็น pointer 15 หัวหน้า เก็บช่อง ไปซ้าย-ขวาได้เพื่อจัดการ Data หัวหน้าเรื่อย

LinkedList<class> list = new **LinkedList<class>()**;

< ที่เก็บ Linked list ไม่: ArrayList
มันก็ extend มาจาก **List** ที่
ที่รุ่น form การประทุม ก็จะมี method
กัน แต่ method พวกตัวบน

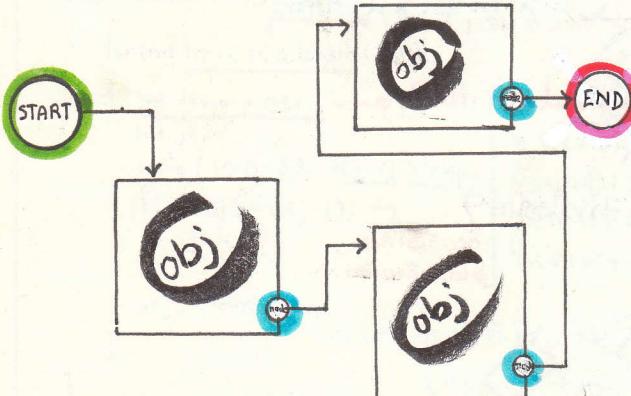
method ของ LinkedList

มันก็คล้าย กับ ArrayList นะ: แต่ LinkedList บันทึก index แบบ array เวลาจะ add, remove หรือ get ให้คืนมา: ที่นั่น กับหัว list ทำหน้าที่!

addFirst(...)	, addLast(...)	: add Data (obj.) ที่หัว หรือ ท้าย
getFirst()	, getLast()	: หัว Data ก็จะเป็นตัวแรก หรือ หัวสุดท้าย
removeFirst()	, removeLast()	: remove Data ที่เก็บอยู่ตัวแรก หรือ หัวที่เพิ่งถูก

{ บันทึก method
เมื่อเวลาความ
สามารถ
ก็ได้แล้ว

โครงสร้างของ LinkedList



↑ data หัวในล ทอยู่ที่ไหนก็ได้ (หนาแน่นใน memory)

tail ต้อง array ที่จะเรียงกับตาม index

▷ หัวเก็บ obj. ของ linkedlist หัวหึ่ง ตัว: ภูมิ: ก่อนหัว 2 ส่วนยังคง
ต่อ ส่วนที่ 10 ไว้เก็บ Data ของ obj. และ อีกส่วนหัวหึ่ง
หัวหึ่ง ก็จะเป็น data หัว ตัวที่ 2 อยู่ที่ไหน เรียกว่า "Node"
โดย จะมี หัว START ซึ่งเป็น node ที่ ซึ่งมี data แรก
ของรายการ (list) ส่วนหัวหึ่ง หัวหึ่ง: ซึ่งมี data END
หัวหึ่ง หรือ "null" (ไม่มี อะไร) ก็แปลว่า ไม่มีตัวต่อ
ตามนั้น อ:

ListIterator<class> `listIterator = objlist.listIterator();`

< บังคับไปแล้วว่า iterator ไม่ได้เป็น pointer บันทุณต้องสร้างจาก list

obj. ที่สร้างจาก class LinkedList จะมี method สำคัญเช่น listIterator() 107 ชิ้น new iterator หรือในแบบง่ายๆ iterator คือตัวไปร่วมกับ pointer ของ list ด้านใน ส่วน method ของมันทำได้ดูในต่อไปนี้

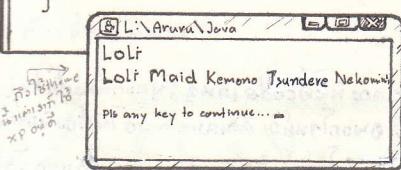
```
import java.util.LinkedList;
import java.util.ListIterator;
```

Code

```
class Tester
{
    public static void main (String [] args)
    {
        LinkedList<String> list = new LinkedList<String>(); ①
        list.addLast ("Kemono");
        list.addLast ("Neko mimi"); ②
        list.addFirst ("Yuri & Yaoi");
        list.addFirst ("Loli"); ③

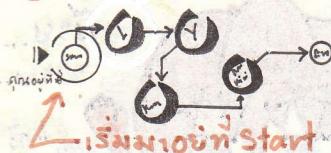
        ListIterator<String> pointer = list.listIterator(); ④
        String temp = pointer.next(); ⑤
        pointer.next(); ⑥ (อย่า Yeo ก็ได้!)
        pointer.remove();
        pointer.add ("Maid"); ⑦
        pointer.next();
        pointer.add ("Tsundere");

        System.out.println (temp);
        for (String name : list)
        {
            System.out.print (name + " ");
        } ⑧
    }
}
```

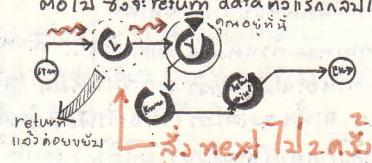


Output (version theme W.Z.)

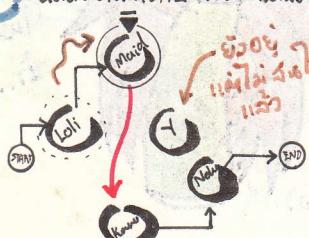
4 สร้าง pointer ขึ้นมาใหม่ให้ list หันด้วย



5 สั่ง next กิลังเลื่อน pointer ไปที่ตำแหน่งต่อไป เช่น: return data หรือแรกจะป้ามีล่อง



7 add ตัวใหม่เข้าไป ตรงตำแหน่งนั้น



1 ความหมายการเรียกนี้เรื่อง Linked List ที่หัน
ก่อน แบบง่ายๆ ว่า data

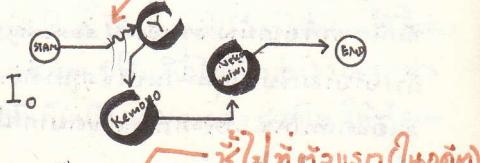


2 addLast กิลังใส่ data ที่ไปที่ตำแหน่ง,
สุดท้าย ฝ่ายล่อง node ไปบุ๊ก Kemono

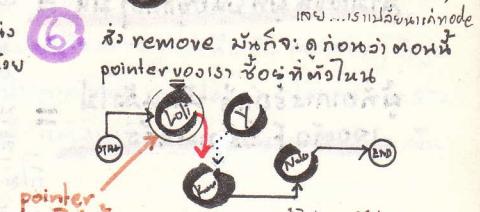


III. ฝ่ายล่อง node
ที่บุ๊ก Neko mimi
เพื่อต่อตัวเรียก Kemono

3 ใช้ชิ้นบังคับ addFirst ไว้ติดหนังสือรวม
เดียวแทน Kemono นั้น แต่จะเปลี่ยน node ของ
START ไปล่องเพื่อเรียก



IV. ฝ่ายล่อง node
ที่บุ๊ก Maid (แรกที่สุด)
เพียง ... รากจะล่อง node ที่ต่อไป



pointer กลับไปรับ Loli
> Loli ฝ่ายล่อง node ที่ซึ่งบุ๊ก ไปทางต่อ
จากนั้น (ไม่มี node ในซึ่งบุ๊ก ก็ต้อง
ร่วมกันลบ remove ออกจาก list).

8 ใช้ for each loop print ทุกตัวใน list

ชิ้น data เอกสารจาก list นั้น ที่มี pointer (iterator)

ไม่เก็บต่อไป...

บางทีต้องลบ ทากดซีก คิมเบอร์ เรือง...

● เวลา คือ pointer หรือ iterator ให้ใช้ก็ได้เวลา ผิดมั้ย อะไร才叫做 จุด word

abcde < ห้องรักษาตัวหนึ่งจะแทน data ทั้ง ส่วนเกินของ ห้อง iterator

↑ เก็บช่องกับมันดูกระ พิรุณร่องๆ ก็ได้:

abcde < เราสั่ง next ครั้งหนึ่ง ก็เนื่องจาก \rightarrow บน keyboard เก็บช่อง ก็จะ return ค่าในตัวเดียว ... (หมายกรอกนี้มันขยับผ่าน a มันก็จะ return a กันไปต่อ)

aA|bcde < ถ้าเราสั่ง add (เนื่องจากน้ำเสียงพิมพ์ต้องการตัวใหม่ จำไม่เป็น) มันก็จะเพิ่มตรงตำแหน่งที่ เก็บช่องอยู่ จริงมั้ย

aAb|cde < ถ้าสั่ง next ลากก์มันขวางไว้ที่นั่น

aA|cde < มาถึงตอนนี้ ถ้าเราสั่ง remove ห้อง: เนื้อห้อง \leftarrow Backspace เก็บช่องอยู่หนึ่ง บล็อกก์กันไป อีก

a|Acde < ถ้าเราอยากรู้ว่าเก็บช่อง (pointer) ขยับกันยังไง \leftarrow แบบนี้ ก็ใช้ method Γ.previous()]

LinkedList / (10) จบ (ภาค 2)

Epilogue.

* ก่อนอื่นขอให้รู้ว่า ก่อนหน้า code 70% ของเรามันเป็น **เขียนรหัส** โดยยังไม่ได้ compile คง compile พอจะต้อง... แต่ logic ถูกกำหนด 100%

* อย่าเพิ่งลืม Java... ใช้ต่อใน data structure ปี 2

... เช่นไว้ 9 บทที่สุดจะจบกันซึ่งกันนี้: ที่กัน Java กัน 60 หน้า ตั้งแต่ ตุ่นเริ่มห้องมาถึงห้องนี้ ซึ่งก็จะผลักกันไป กัน เนื่องจากที่ ย่างกันไป ง่ายขึ้น (ส่วนใหญ่จะ: บล็อกว่า ย่างกันมากกว่า หรือ ย่างกันนิดๆ) แล้วก็จะ เนื่องจากนั้นไป... ซึ่งเรา ก็พยายามจะเขียนง่ายขึ้น ไม่ใช้ภาษาที่ง่ายๆ ที่สุดแล้ว (ขอรับ ก็ไม่ว่า ต้องยังคงภาษาเดิมๆ และ logic ก็ไม่ต้องต้องมีความซับซ้อนเท่าไหร่... เพราะ: ถ้าเรา: เขียนภาษาไปมันจะ: กว้างเข้ม สรุป C (กัน))

▶ คำแนะนำสำหรับ...

คนเขียนโปรแกรมทั่วไป = สำหรับคนที่ทำงาน logic ก็ไม่ใช่ภาษา ก็ไม่ต้องดัง ลง พิมพ์ code เล่นๆ (คุณแค่พิมพ์เข้าไป)
||: อ่าน text "ผ่านๆ" ชัดเจน พอเขียน idea ดูง่ายๆ ก็พอไปทำได้ แต่ต้องมีพิมพ์ก็ได้

สาวกบุญชุน ผู้ที่เขียนบล็อกภาษา มีน = Logic ของท่านจะสูงกว่าคนที่ไม่ได้ เพราะ: ฉัน จงทำให้ชัดเจน จัดระเบียบ ทำให้แล้วก้ามีเดช ทุกโจทย์ที่เจอกัน ||: ลองพูดออกเสียง ก็จะเข้าใจง่ายๆ มาก จงรักษาความเรียบง่าย (พวกนั้นอย่างกันนี้ ตามดูแล้ว แล้วก็จะเข้าใจง่ายๆ !) สมสัย 0:15 แม้แต่หนอนๆ ก็สามารถเขียน

ผู้ที่ต้องการรับ||: นันหนังไป 170 ตัว F วันทางหลัง = บล็อกตามตรง ทำนองด้านต่อไปนี้ C+... คำแนะนำ
ต่อ ทำนองนี้ไม่ใช่ห้องเดียว สำหรับ ที่ไม่ต้อง logic
ก็จะไม่ได้ ฉัน: ฉัน จงไว้ เดียวให้เข้าใจที่เพื่อนๆ ที่ไม่เข้าใจ
มาเข้าห้องไปเล่น (ภาษาต้องบ่อก ไม่ได้ไป) ||: ใน

นั่งลง จะ ย่องเข้าไป เขียนไปล้วนๆ ชักหน่อ
อาจจะต้องตัดต่อ แต่ก็ต้องรักษาความเรียบง่าย

▶ คำขอตุก...

- จัด: ชี้ชัด บล็อก ภาษา + confirm code

- ลดข้อผิดพลาด ให้เข้าใจง่ายๆ ให้เข้าใจง่ายๆ

- ทุกท่านที่ประชุมกันต้องตั้ง บล็อก ก็จะ ตั้งนั้นชั้นนั้น... ถ้าไม่ใช่ ทุกท่านจะต้องตั้งตัวเอง

- ห้ามที่กำลังต่อเนื่องกันอยู่หัวและ: ข้อมูลที่ อุตสาห์ ต่อ นั้นจะ ต้องหันกลับ 60

(ต้องหัน 60 องศา ไว้เมื่อ ไม่ต้องสั่ง) , //

สุดท้าย... จะเป็นเจ้า: get A กันทุกดัน น้ำ 戍

comment ปีหนึ่งลับ: อาจารย์สอนเรื่องเก็บห้องของภาษา เช่นนี้... รุ่น ก็จะ A ไปได้ 80%

Java / 72 (09) 10 มีนา.