แนะนำภาษาจาวา

หัวข้อ

- ประวัติของภาษาจาวา
- ข้อดีของภาษาจาวา
- จาวาแพลตฟอร์ม
- การคอมไพล์และรันโปรแกรมภาษาจาวา
- โปรแกรม **HelloWorld** ในแบบตัวอักษรและแบบกราฟิกส์

ประวัติของภาษาจาวา

ภาษาโอ๊ค

- เริ่มต้นขึ้นในปีค.ศ. 1990
- แพททริก นอทัน (Pattrick Naughton) จะลาออกจากซัน
- เจมส์ กอสลิ่ง (James Gosling) ผู้ให้กำเนิดภาษาจาวา
- กรีนทีม
  - ยุคถัดจากคอมพิวเตอร์ส่วนบุคคล (personal computer) ก็คือคอมพิวเตอร์ในอุปกรณ์อีเล็กโทรนิกส์ (consumer electronics)
- ดุ๊ค

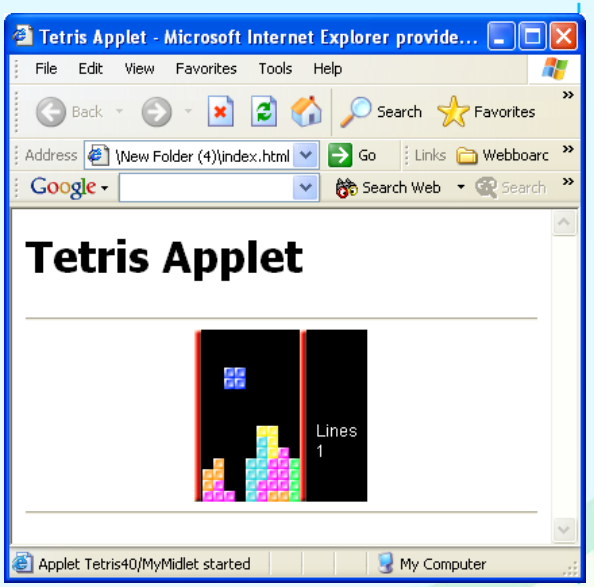
ภาษาจาวา

- มาร์ค แอนดรีสเซน กับ Mosaic และเว็บ
- บิล จอย (Bill Joy) แจกจ่ายตัวพัฒนาภาษา
- นอทันพัฒนาเว็บเบราเซอร์ที่สนับสนุนภาษาจาวา โปรแกรมดังกล่าวมีชื่อว่า HotJava
- Netscape Navigator
- 23 มกราคม ค.ศ. 1996 ซันได้ออก **JDK** 1.0

ข้อดีของภาษาจาวา

ข้อดีของภาษาจาวา

- ทำงานบนเว็บเบราเซอร์ได้
- ความปลอดภัยสูง
- สนับสนุนงานหลายระดับ



ข้อดีของภาษาจาวา

- สามารถทำงานบนเครื่องคอมพิวเตอร์ต่างระบบได้
- ภาษาจาวาเป็นภาษาเชิงวัตถุ
- ความทันสมัย
- ความเรียบง่าย

ข้อดีของภาษาจาวา

- กลไกในการคืนพื้นที่ในหน่วยความจำอัตโนมัติ (garbage collection)
- มีคลาสและอินเตอร์เฟซให้ใช้เยอะมาก
  - —794 interfaces
  - —2485 classes

จาวาแพลตฟอร์ม

แพลตฟอร์ม (platform)

- ระบบที่โปรแกรมอาศัยทำงาน
- **Hardware**
- **Software**

จาวาแพลตฟอร์ม

- **Java 2 Platform, Standard Edition (J2SE)**
  - จาวาแอพพลิเคชัน (Java application)
  - แอพเพลต (Java applet)
- **Java 2 Platform, Enterprise Edition (J2EE)**
  - โปรแกรมแบบมัลติเทียร์ (multitiered) สำหรับการพัฒนาโปรแกรมในระดับองค์กร
- **Java 2 Platform, Micro Edition (J2ME)**
  - สินค้าอีเล็กโทรนิกส์ เช่น โทรศัพท์มือถือ พีดีเอ (personal digital assistant) และกล่องเคเบิ้ลทีวี (TV set-top box

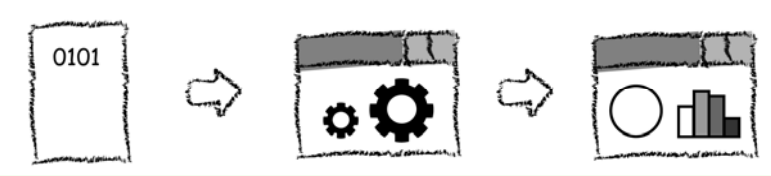
การคอมไพล์และรันโปรแกรมภาษาจาวา

ชุดพัฒนาภาษาจาวา (JDK)

- ชุดพัฒนาภาษาจาวา (Java Development Kit - JDK)
  - จาวาคอมไพเลอร์ (javac.exe)

  - สภาพแวดล้อมการรันโปรแกรมจาวา (Java Runtime Environment - JRE) (java.exe)

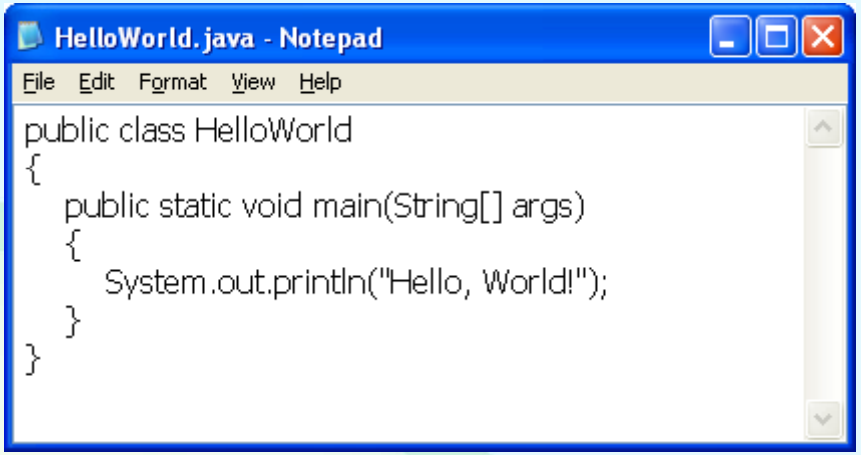
- Download http://java.sun.com



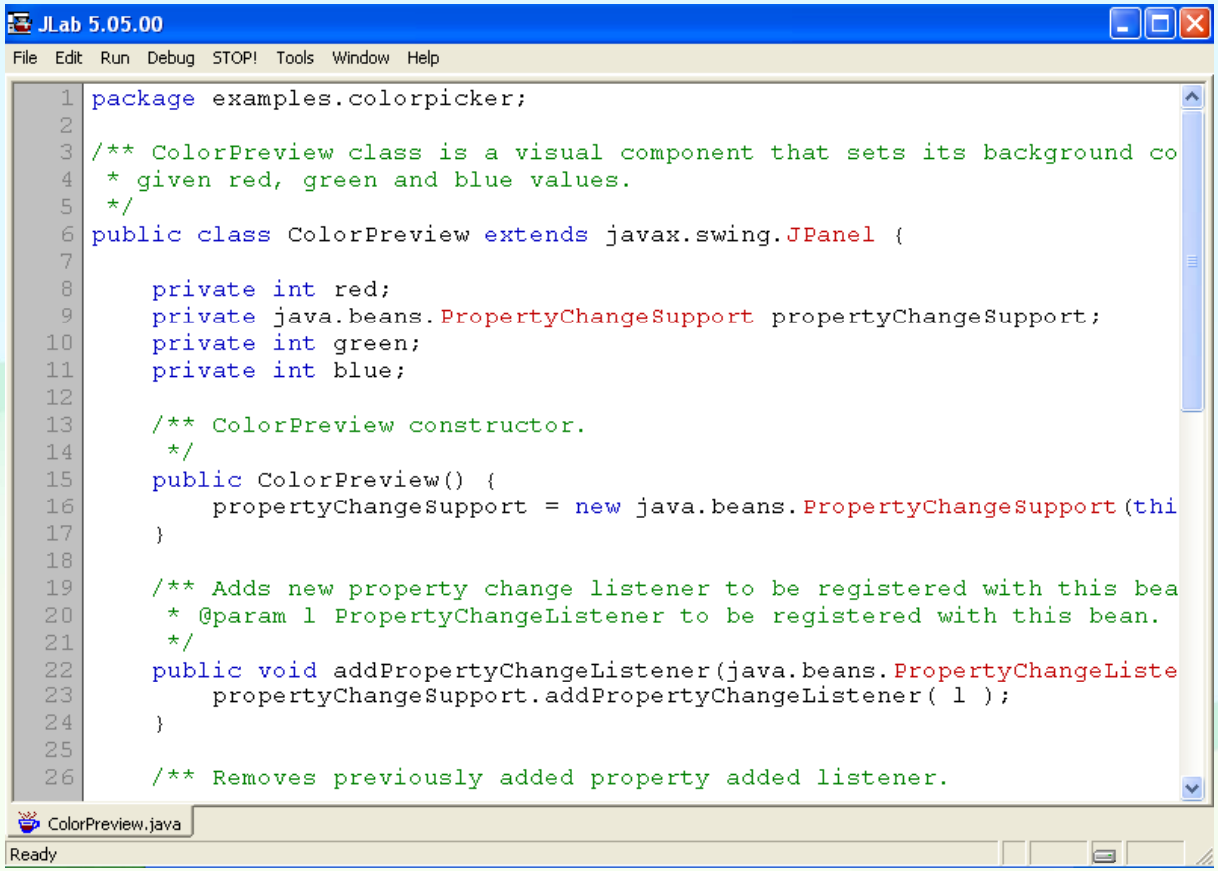
โปรแกรมที่ใช้เขียนต้นฉบับโปรแกรม

- **Notepad**
  - มาพร้อมกับ Windows
- **J-Lab**
  - http://www.cp.eng.chula.ac.th/~somchai/JLab/
- **Netbeans**
  - http://www.netbeans.org/
- **Eclipse**
  - http://www.eclipse.org/

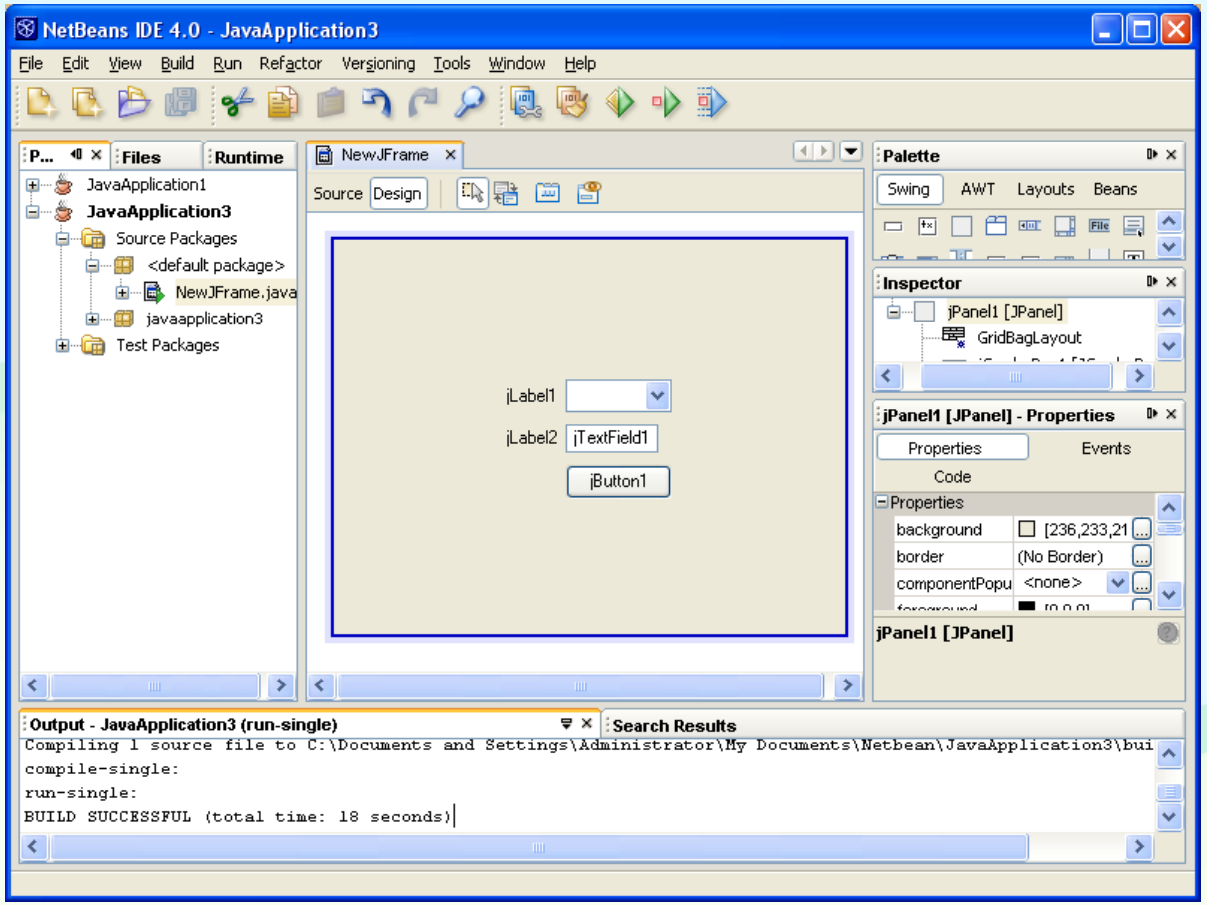
Notepad


HelloWorld.java - Notepad
File Edit Format View Help
public class HelloWorld
{
public static void main(String[] args)
{
System.out.println("Hello, World!");
}
}

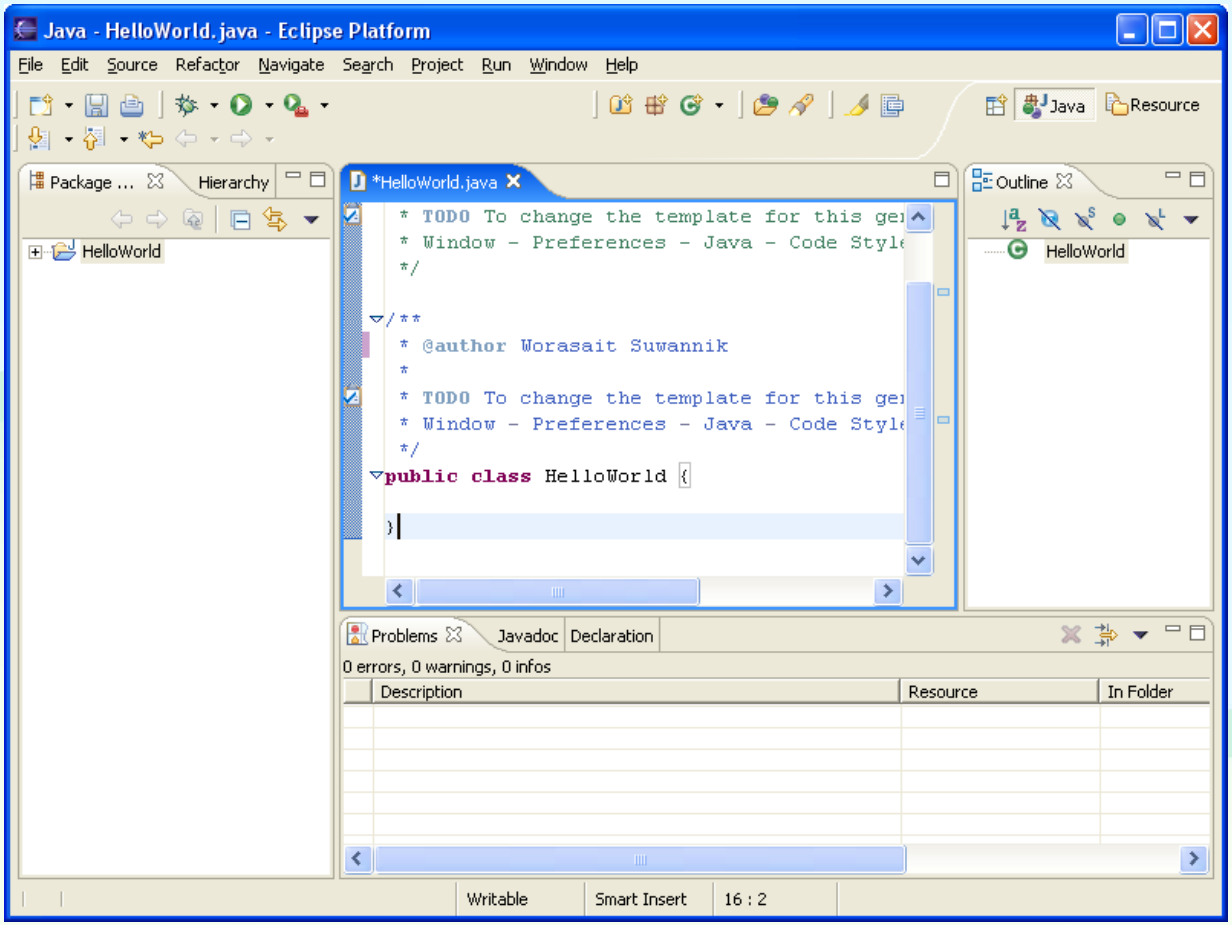
J-Lab


JLab 5.05.00
File Edit Run Debug STOP! Tools Window Help
1 package examples.colorpicker;
2
3 /** ColorPreview class is a visual component that sets its background co
4 * given red, green and blue values.
5 */
6 public class ColorPreview extends javax.swing.JPanel {
7
8 private int red;
9 private java.beans.PropertyChangeSupport propertyChangeSupport;
10 private int green;
11 private int blue;
12
13 /** ColorPreview constructor.
14 */
15 public ColorPreview() {
16 propertyChangeSupport = new java.beans.PropertyChangeSupport(thi
17 }
18
19 /** Adds new property change listener to be registered with this bea
20 * @param l PropertyChangeListener to be registered with this bean.
21 */
22 public void addPropertyChangeListener(java.beans.PropertyChangeListe
23 propertyChangeSupport.addPropertyChangeListener(l);
24 }
25
26 /** Removes previously added property added listener.
ColorPreview.java
Ready

Netbeans


NetBeans IDE 4.0 - JavaApplication3
File Edit View Build Run Refactor Versioning Tools Window Help
Files
Runtime
JavaApplication1
JavaApplication3
Source Packages
<default package>
NewJFrame.java
javaapplication3
Test Packages
NewJFrame
Source Design
jLabel1
jLabel2
jTextField1
jButton1
Palette
Swing AWT Layouts Beans
Inspector
jPanel1 [JPanel]
GridBagLayout
jPanel1 [JPanel] - Properties
Properties Events
Code
Properties
background [236,233,21
border (No Border)
componentPopu <none>
jPanel1 [JPanel]
Output - JavaApplication3 (run-single)
Search Results
Compiling 1 source file to C:\Documents and Settings\Administrator\My Documents\Netbean\JavaApplication3\bui
compile-single:
run-single:
BUILD SUCCESSFUL (total time: 18 seconds)

Eclipse


Java - HelloWorld.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Java
Resource
Package ...
Hierarchy
HelloWorld
*HelloWorld.java
* TODO To change the template for this ge
* Window - Preferences - Java - Code Style
*/
/**
* @author Worasait Suwannik
*
* TODO To change the template for this ge
* Window - Preferences - Java - Code Style
*/
public class HelloWorld {
}
Outline
HelloWorld
Problems
Javadoc
Declaration
0 errors, 0 warnings, 0 infos
Description
Resource
In Folder
Writable
Smart Insert
16 : 2

โปรแกรม HelloWorld ในแบบตัวอักษรและแบบกราฟิกส์

โปรแกรม HelloWorld

- พิมพ์คำว่า Hello, World! ขึ้นบนหน้าจอคอมพิวเตอร์

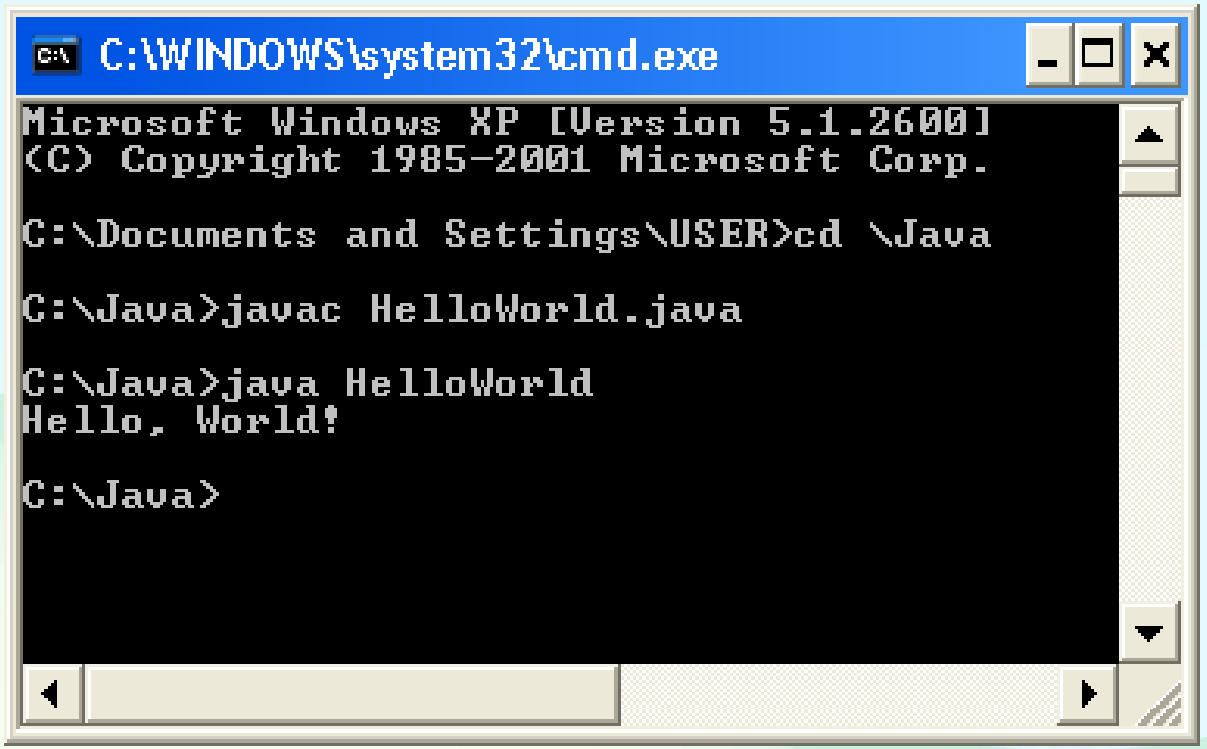
HelloWorld.java - Notepad

File Edit Format View Help

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
```

การเปิดโปรแกรม cmd

คอมไพล์โปรแกรม



โปรแกรม HelloWorld

```
public class HelloWorld

{




}
```

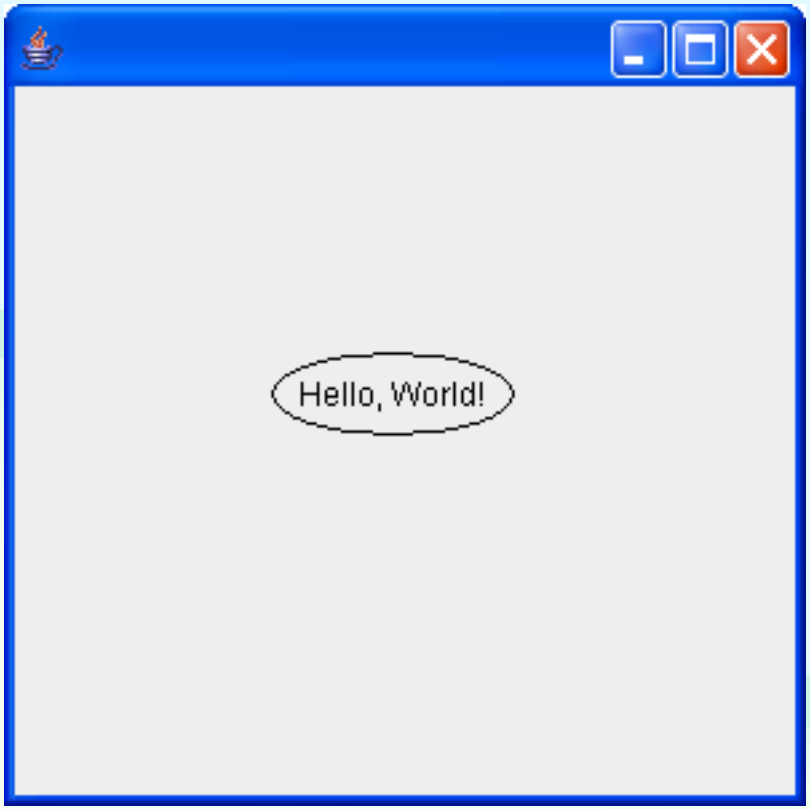
โปรแกรม HelloWorld

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

    }
}
```

โปรแกรม HelloWorld

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
```

โปรแกรม HelloWorld แบบกราฟิก



โปรแกรม HelloWorld แบบกราฟิก

```
import java.awt.*;
import javax.swing.*;
public class HelloGraphicsWorld {

      public static void main(String[] args) {
    JFrame frame = new JFrame() {
      public void paint(Graphics g) {
        g.drawString("Hello, World!", 110, 150);
        g.drawOval(100, 130, 90, 30);
      }

    };

    frame.setSize(300, 300);
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setVisible(true);
  }
```

คำอธิบายในโปรแกรม

- คอมเมนต์บรรทัดเดียว

```
// print the word hello
```

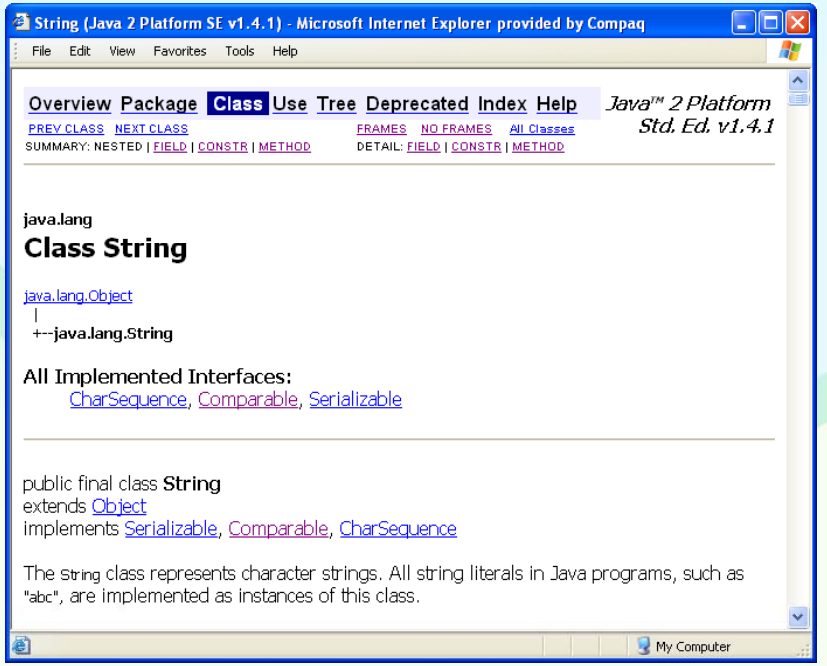
- คอมเมนต์ที่เป็นย่อหน้า

```
/*
    This program
    will print
    the word hello
```

คำอธิบายในโปรแกรม

- คอมเมนต์ที่เป็นย่อหน้าสำหรับการสร้างเอกสารด้วยโปรแกรม **javadoc**

```
/**


*/
```



สรุป

สรุป

- โปรแกรมในจาวาแพลตฟอร์มมาตรฐานมีอยู่สองประเภทคือ **แอพพลิเคชัน** และ **แอพเพลท**
- **จาวาแอพพลิเคชัน**สามารถทำงานได้เหมือนกับโปรแกรมทั่วไป มันสามารถทำงานบนคอมพิวเตอร์ต่างรุ่นต่างระบบปฏิบัติการได้ เพราะว่ามันทำงานอยู่บนโปรแกรมเครื่องจักรเสมือน
- **จาวาแอพเพลต**สามารถทำงานบนเว็บเพจได้ มันสามารถทำงานบนคอมพิวเตอร์ต่างรุ่นต่างระบบปฏิบัติการได้ ขอเพียงแค่มีโปรแกรมเว็บเบราเซอร์ที่สนับสนุนภาษาจาวาติดตั้งอยู่บนระบบปฏิบัติการนั้น

สรุป

- ซอร์สโค้ดโปรแกรมจาวาอยู่ในแฟ้มที่มีนามสกุล java
- โปรแกรมที่ใช้คอมไพล์ภาษาจาวาชื่อ javac
- ผลการคอมไพล์จะได้แฟ้มที่มีนามสกุล class
- ไฟล์นามสกุล class สามารถทำงานใน JRE ได้
- ใช้โปรแกรม java รันไฟล์ .class ที่มีเมธอด main()
- การพิมพ์ข้อความใช้คำสั่ง
  System.out.println("*ข้อความที่ต้องการพิมพ์*")

ชนิดข้อมูลพื้นฐาน

หัวข้อ

- ตัวแปร
- ชนิดข้อมูลพื้นฐาน
- การคำนวณ
- ค่าคงที่

ตัวแปร

ตัวแปร

- กล่องสี่เหลี่ยม

  $\square = 5 + 3$

- ตัวอักษร

  $x = 5 + 3$

- คำ

  $\text{count} = 5 + 3$

การประกาศตัวแปร

- รูปแบบ

  ชนิดข้อมูล ชื่อตัวแปร;

- ตัวอย่าง

```
int count;

double sum;
```

การตั้งชื่อ

- ต้องเริ่มต้นชื่อด้วย

  ตัวอักษรโรมัน (เช่น a, z, A, Z)

  เช่น count

  เครื่องหมาย _ หรือ $

  เช่น _height หรือ $name
- ตัวเลขสามารถอยู่ในชื่อตัวแปรได้
  - เช่น car12 หรือ c1a2r

ห้าม ขึ้นต้นชื่อด้วยตัวเลข

ห้าม ตั้งชื่อด้วยคำสงวน (reserved word)

คำสงวน (Reserved Words)

- abstract, assert, boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, enum, extends, final, finally, float, for, goto, if, implements, import, instanceof, int, interface, long, native, new, package, private, protected, public, return, short, static, strictfp, super, switch, synchronized, this, throw, throws, transient, try, void, volatile, while

กำหนดค่าให้ตัวแปร

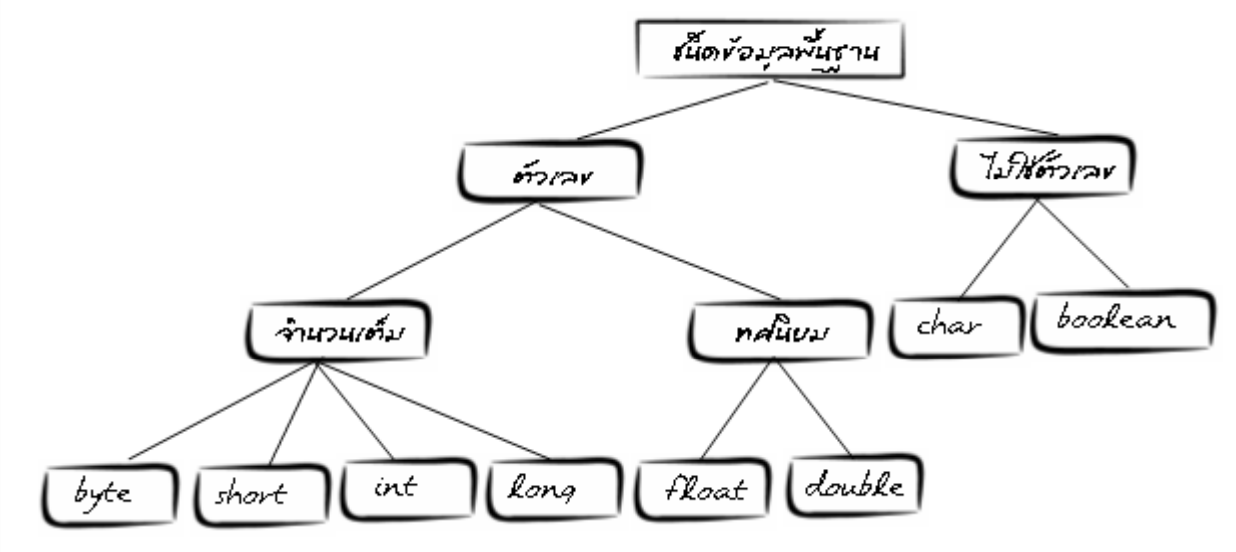
- รูปแบบ
  - *ตัวแปร = ค่าที่ต้องการกำหนด;*
- ตัวอย่าง
  - count = 3;

ชนิดข้อมูลพื้นฐาน

ชนิดข้อมูลพื้นฐาน

ชนิดข้อมูล	
boolean	
char	
string	
byte	
short	
int	
long	
float	
double	

แผนผัง


ชนิดข้อมูลพื้นฐาน
ตัวเลข
ไม่ใช่ตัวเลข
จำนวนเต็ม
ทศนิยม
char
boolean
byte
short
int
long
float
double

System.out.println(“ข้อความ" + ตัวแปร);

การรับค่า

```
import java.util.Scanner;

java.util.Scanner sc;

sc = new java.util.Scanner(System.in);

String s = Sc.nextLine(); //ใช้สำหรับรับค่าเป็น Sting

int i = Sc.nextInt(); //ใช้สำหรับรับค่าเป็น interger

float f = Sc.nextFloat(); //ใช้สำหรับรับค่าเป็น float

double d = Sc.nextDouble(); //ใช้สำหรับรับค่าเป็น double

long l = Sc.nextLong(); //ใช้สำหรับรับค่าเป็น long
```

การคำนวณ

การคำนวณ

- บวก

```
int a = 1;
int b = 2;
int sum = a + b;
```

- ลบ
  - double a = 1.2;
  - double b = 3.4;
  - double result = a - b;

การเพิ่มค่า

- เครื่องหมาย +=

```
int count = 0;

count += 1;
```

- เครื่องหมาย ++ (เพิ่มค่าอีกหนึ่ง)

```
int count = 0;

count++;
```

การคำนวณ

- คูณ

```
int result;

result = 3 * 4;
```

- หาร

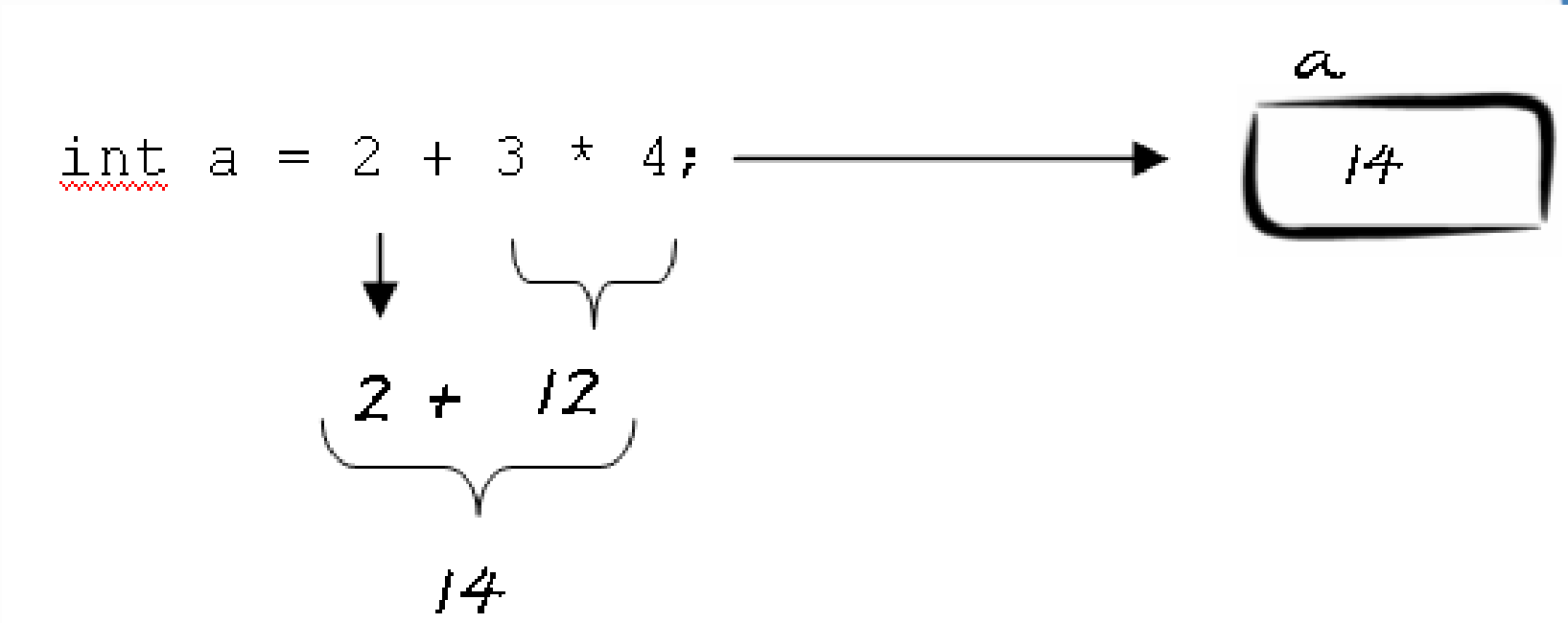
```
double a = 14;

double result = a / 3;
```

การหารเอาเศษ

- เครื่องหมาย %
- ตัวอย่าง

```
int remainder;

remainder = 14 % 7;

remainder = 15 % 7;
```

ใช้วงเล็บเพื่อจัดกลุ่มการคำนวณ



เปรียบเทียบผลการคำนวณ

- int a = 2 + (3 * 4);
- int a = (2 + 3) * 4;

การแสดงผลการคำนวณ

- คำสั่ง System.out.println()
- ตัวอย่าง

```
public class PrintMoney1 {
    public static void main(String[] arg) {
        int money = 12;
        money *= 3;
        System.out.println(money);
    }
```

การแสดงผลการคำนวณ

- ตัวอย่าง

```
public class PrintMoney2 {
    public static void main(String[] arg) {
        int money = 12;
        money *= 3;
        System.out.println("Money is " + money);
    }
```

โปรแกรมคำนวณราคารวมภาษีมูลค่าเพิ่ม

```
public class TaxCalculator {
    public static void main(String[] args) {
        double price      = 50;
        double vat        = price * 7 / 100;
        double totalPrice = price + vat;
        System.out.println("Price include VAT is " + totalPrice);
    }
}
```

ค่าคงที่

เลขพิศวง (Magic Number)

```
public class TaxCalculator2 {
    public static void main(String[] args) {
        double price      = 50;
        double totalPrice = price * 1.07;
        System.out.println("Price include VAT is " + totalPrice);
    }
}
```

ค่าคงที่

- แก้ปัญหาเลขพิศวง
- ตรงข้ามกับตัวแปร
- รูปแบบ
  - final *ชนิดข้อมูล ชื่อค่าคงที่ = ค่า*;
- ตัวอย่าง
  - final double VAT_RATE = 1.07;

การคำนวณภาษีโดยใช้ค่าคงที่

```
public class TaxCalculatorConst {
    public static void main(String[] args) {
        final double VAT_RATE = 1.07;

        double tvPrice  = 10000;
        double dvdPrice = 600;
        double cdPrice  = 200;

        double totalTVPrice  = tvPrice  * VAT_RATE;
        double totalDVDPrice = dvdPrice * VAT_RATE;
        double totalCDPrice  = cdPrice  * VAT_RATE;
```

สรุป

สรุป

- ตัวแปรเป็นเหมือนกับกล่องที่สามารถใส่ค่าต่างๆลงไป
- ตัวแปรแบ่งออกเป็น 2 ประเภทคือ
  - ตัวแปรชนิดข้อมูลพื้นฐาน
  - ตัวแปรที่อ้างถึงวัตถุหรือเรฟเฟอร์เรนซ์

สรุป

- การบรรยายนี้กล่าวถึงตัวแปรประเภทแรก นั่นคือตัวแปรชนิดข้อมูลพื้นฐาน ซึ่งสามารถแบ่งออกเป็น 2 ประเภทย่อยๆ คือ
  - ตัวเลข ได้แก่ byte, short, int, long, float และ double
  - ไม่ใช่ตัวเลข ได้แก่ char และ boolean

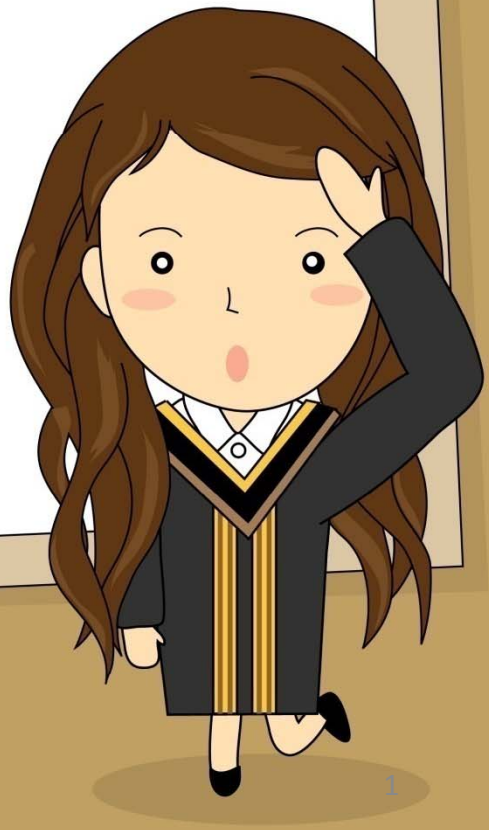
สรุป

- ตัวแปรที่เป็นตัวเลขสามารถนำมาคำนวณได้โดยใช้เครื่องหมาย +, -, *, /, และ % เป็นต้น
- การคำนวณจะเริ่มจากเครื่องหมายที่มีความสำคัญมากกว่าก่อน
- เพื่อความแน่นอน ให้ใช้วงเล็บช่วยจัดลำดับการคำนวณ

สรุป

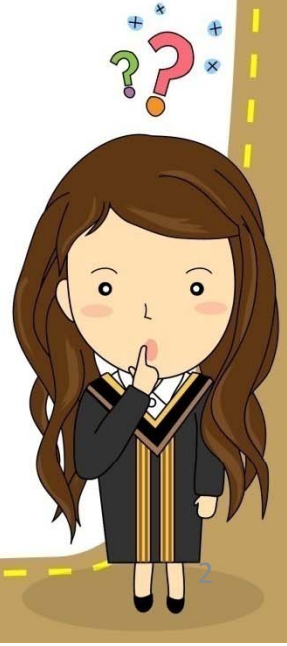
- ไม่ควรใช้เลขพิศวงในการคำนวณ ให้ใช้ค่าคงที่แทน
- การประกาศค่าคงที่ทำได้โดยใช้คำว่า final

ประโยคควบคุม

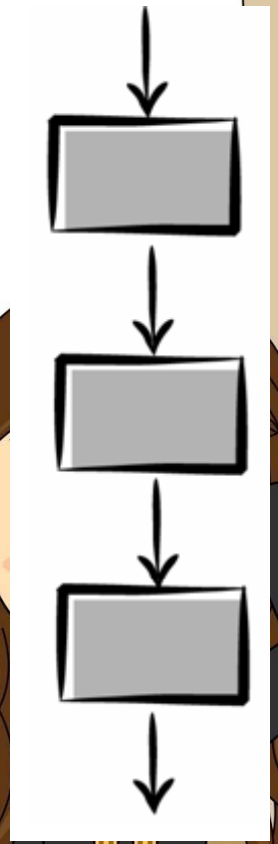


หัวข้อ

- การทำงานแบบเรียงลำดับ
- ประโยค if, if-else
- ประโยค switch
- ประโยคที่ใช้ทำงานซ้ำหรือลูป
  - for
  - while
  - do-while
- คำสั่ง break
- คำสั่ง continue

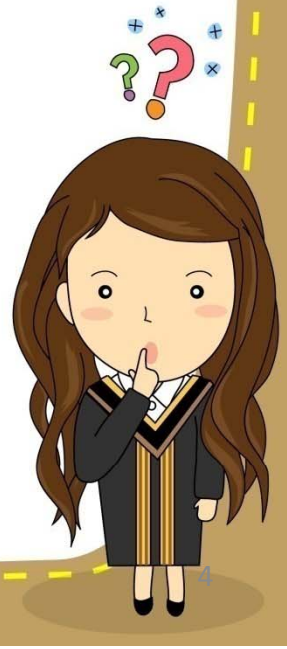


การทำงานแบบเรียงลำดับ

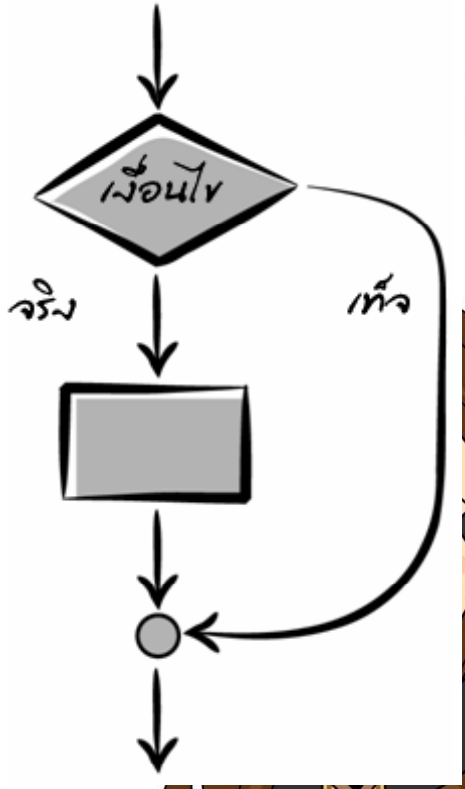


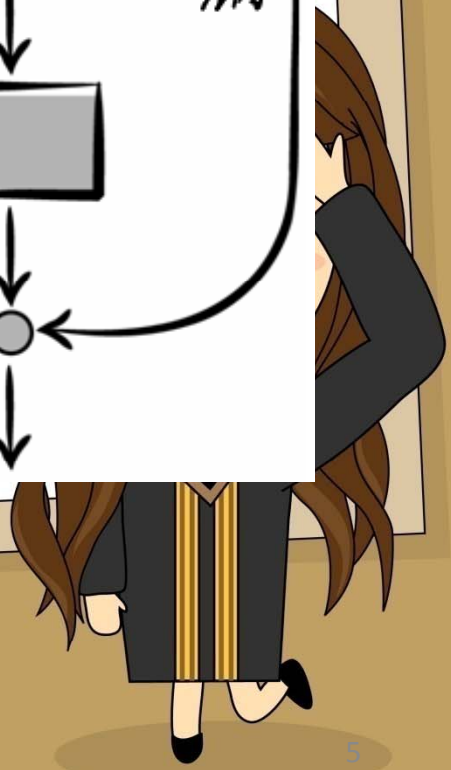
การทำงานแบบเรียงลำดับ

```
public class Shopping
{
   public static void main(String[] args)
   {
      int cash      = 500;
      int bookPrice = 180;
      int foodPrice =  20;
      int sodaPrice =   7;
      cash -= bookPrice;
      cash -= foodPrice;
      cash -= sodaPrice;
      System.out.println("Cash = " + cash);
   }
}
```



ประโยค if





ประโยค if

- รูปแบบ

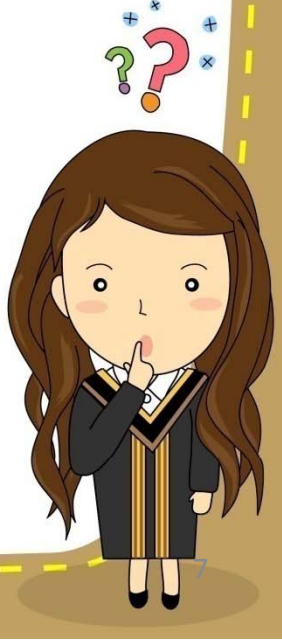
```
if (เงื่อนไข)
{
    ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นจริง;
}
```

- ตัวอย่าง

```
if (cash >= bookPrice)  {
   System.out.println("Buy book.");
}
```

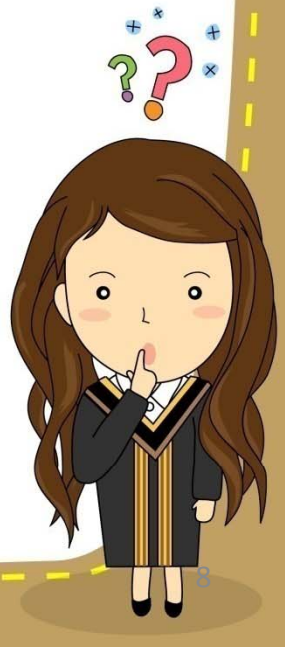
ตัวอย่างการตัดสินใจซื้อหนังสือ

```
public class If {
    public static void main(String[] args)    {
        int cash      = 200;
        int bookPrice = 180;

        if (cash >= bookPrice)  {
            cash -= bookPrice;
            System.out.println("Buy book.");
        }

        System.out.println("Cash = " + cash);
    }
}
```



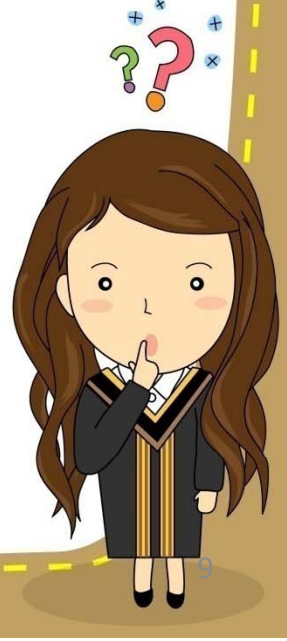
การเปรียบเทียบ

1. <
2. >
3. <=
4. >=
5. ==
6. !=



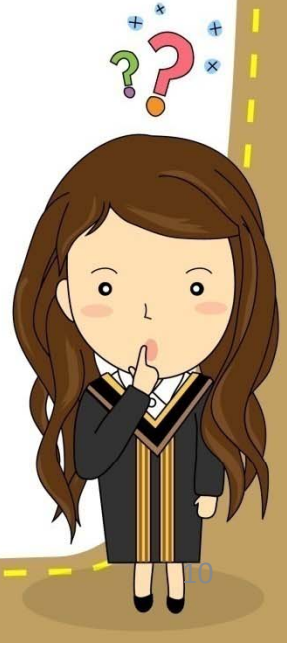
การเปรียบเทียบทศนิยม

- แทนที่จะเทียบความเท่ากัน บางทีควรเทียบว่ามีค่าใกล้เคียงกันหรือไม่
- เช่น $d = \sqrt{2}$
  - d * d ไม่เท่ากับ2
  - d * d มีค่าใกล้เคียงกับ 2



ตัวดำเนินการแบบบูล (Boolean operator)

- && คือ AND
  - เช่น เงื่อนไข cash >= price && price < 150
- || คือ OR
- ! คือ NOT

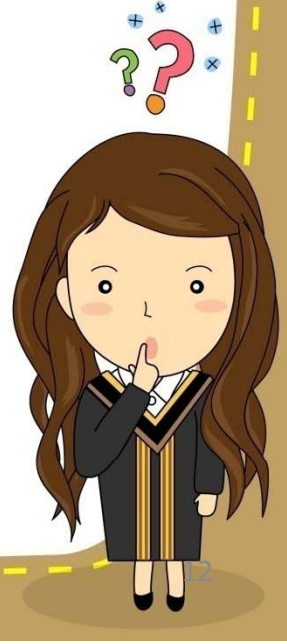


ตัวดำเนินการแบบบูล

เงื่อนไข		ตัวดำเนินการแบบบูล		
A	B	A && B	A || B	!A
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

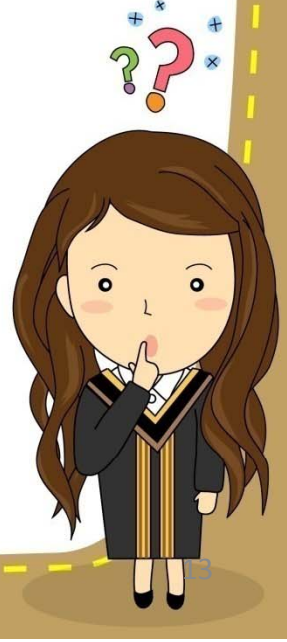
ตัวดำเนินการแบบบูล

```
public class IfAnd
{
   public static void main(String[] args)
   {
      int cash  = 220;
      int price = 100;

      if (cash >= price && price < 150)
         System.out.println("Buy book.");
   }
}
```



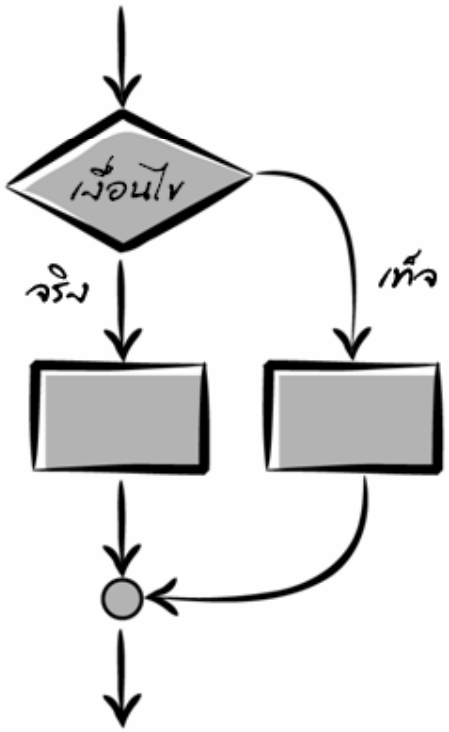
การใช้ตัวแปรแทนเงื่อนไข

- if (มีเงินพอ && ราคาที่สมเหตุผล)
- if (cash >= price && price < 150)
- boolean hasEnoughCash = cash >= price;
  boolean reasonablePrice = price < 150;
  if (hasEnoughCash && reasonablePrice)

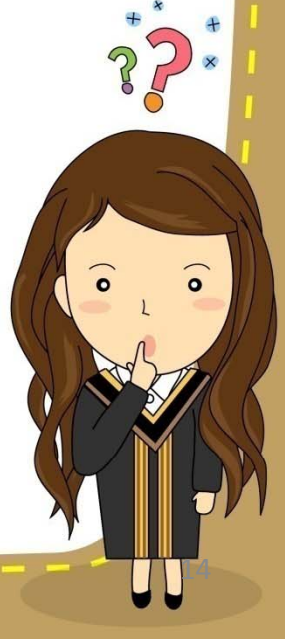


การทำงานเมื่อเงื่อนไขเป็นเท็จ (else)

- รูปแบบ

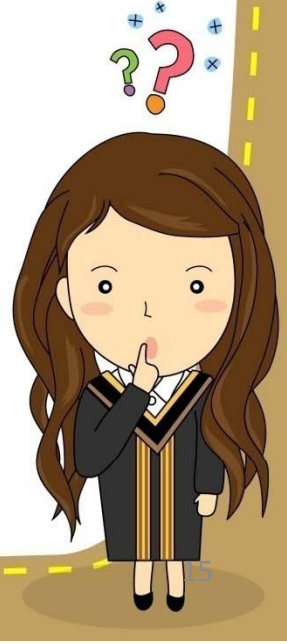



```
if (เงื่อนไข)
{
   ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นจริง;
}
else
{
   ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นเท็จ;
}
```



ตัวอย่างการตัดสินใจใช้บัตรเครดิต

```
 int cash  = 200;
int price = 380;

if (cash >= price)
{
   cash -= price;
   System.out.println("Use cash");
}
else
{
   System.out.println("Use card");
}
system.out.println(“Cash = “ + cash);
}
}
```



การกำหนดค่าตามเงื่อนไข

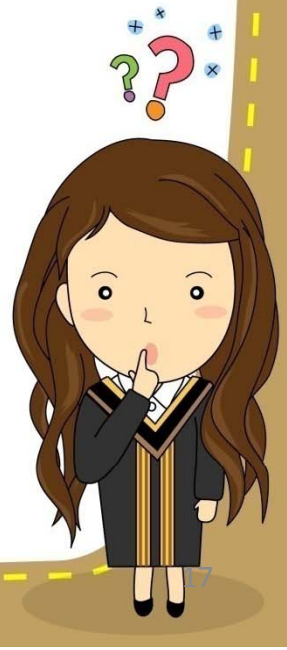
- รูปแบบ
  - *ตัวแปร* = *เงื่อนไข* ? *a* : *b*;
- ตัวอย่าง

```
int price = 99;
int withdraw = price <= 100 ? 100 : 200;
```

ถ้าเงื่อนไขเป็นจริง withdraw มีค่าเป็น 100 ถ้าเงื่อนไขเป็นเท็จ withdraw มีค่าเป็น 200

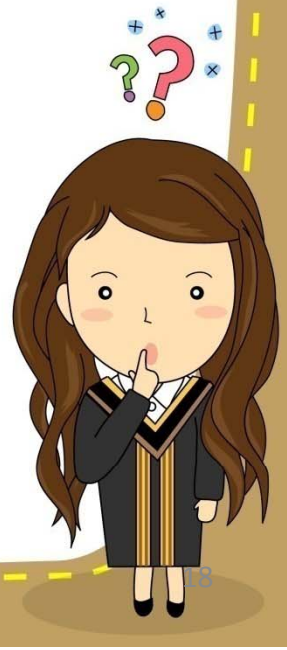
การจัดระเบียบโปรแกรม

```
public class IfAssignMessy
{
public static void main( String[] args )
{
int price = 99;
int withdraw;
if( price <= 100 )
withdraw = 100;
else
withdraw = 200;
System.out.println("Withdraw = " + withdraw);
}
}
```



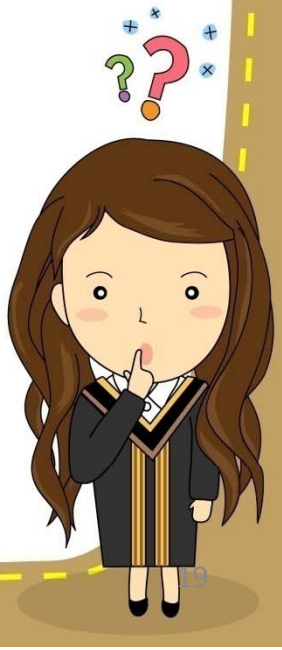
การจัดระเบียบโปรแกรม

```
public class IfAssign
{
    public static void main(String[] args)
    {
        int price = 99;
        int withdraw;

        if (price <= 100)
            withdraw = 100;
        else
            withdraw = 200;

        System.out.println("Withdraw = " + withdraw);
    }
}
```



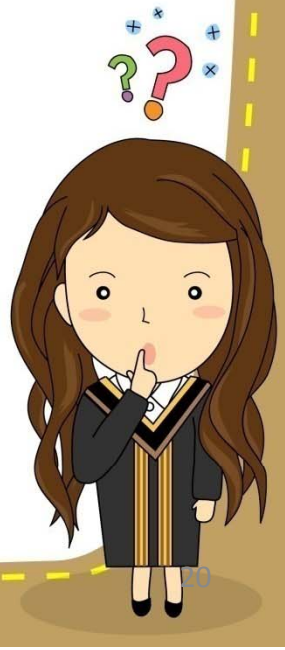
การจัดระเบียบโปรแกรม

```
public class IfAssign {
    public static void main(String[] args) {
        int price = 99;
        int withdraw;

        if (price <= 100)
            withdraw = 100;
        else
            withdraw = 200;

        System.out.println("Withdraw = " + withdraw);
    }
}
```



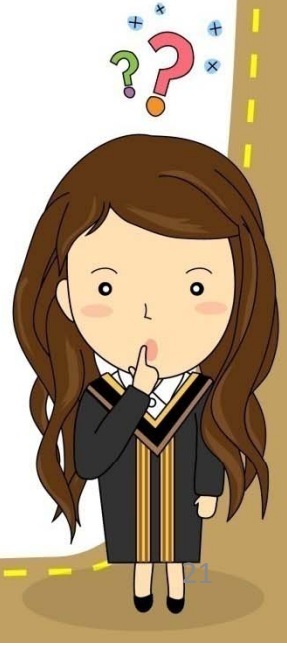
รูปแบบ if-else if

```
if (เงื่อนไข1)
{
    ประโยคที่จะทำงานเมื่อ เงื่อนไข1 เป็น จริง;
}
else if (เงื่อนไข2)
{
    ประโยคที่จะทำงานเมื่อ เงื่อนไข2 เป็น จริง;
}
else
{
    ประโยคที่จะทำงานเมื่อ เงื่อนไขก่อนหน้านี้ทั้งหมด เป็น เท็จ;
}
```

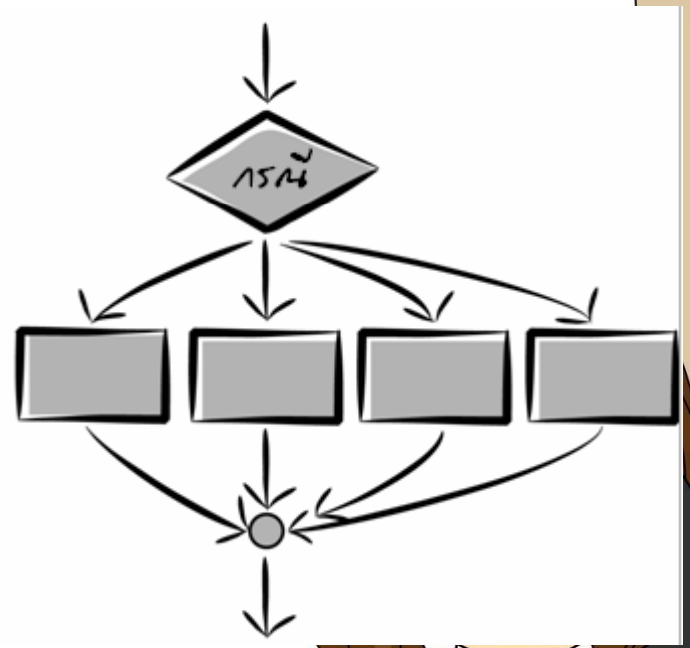


ตัวอย่างการคิดเกรด

```
char grade;
int score = 70;

if (score >= 90)
   grade = 'A';
else if (score >= 80)
   grade = 'B';
else if (score >= 60)
   grade = 'C';
else if (score >= 40)
   grade = 'D';
else
   grade = 'F';
```

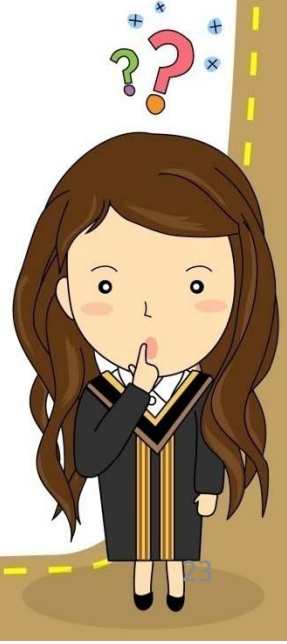


ประโยค swit



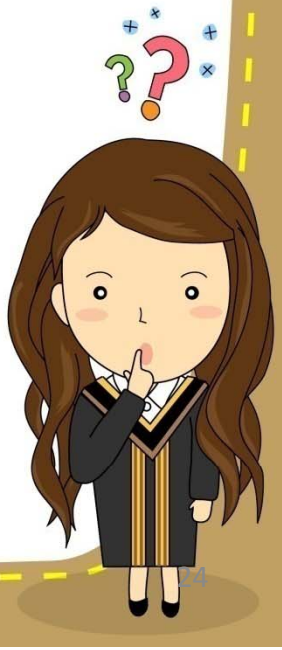
รูปแบบ

```
switch (กรณี)
{
case กรณี1:
   ประโยค;
   break;
case กรณี2:
   ประโยค;
   break;
default:
   ประโยค;
}
```



ตัวอย่างการพิมพ์ข้อความชมเชยตามเกรดที่ได้

```
char grade = 'B';
switch (grade)
{
case 'A':
   System.out.println("ดีมาก");
   break;
case 'B':
   System.out.println("ดี");
   break;
case 'C':
   System.out.println("พอใช้");
   break;
case 'D':
   System.out.println("ต้องปรับปรุง");
   break;
default:
   System.out.println("แก้ไข");
}
```



Menu

1. บวก
2. ลบ
3. คูณ
4. หาร

แบบฝึกหัด

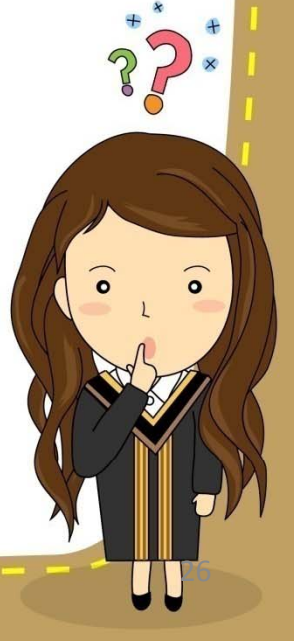
ให้รับ ชื่อผู้ซื้อ และเลือกหนังสือ และให้กำหนดจำนวนหนังสือ โดยมีเมนูดังนี้

MENU

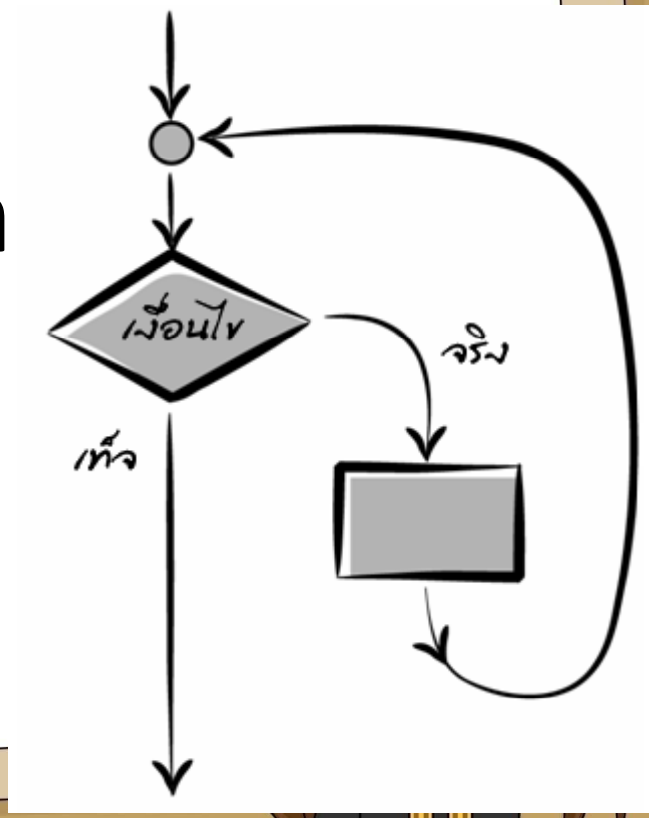
1. การเขียนโปรแกรมภาษาจาวา
2. คู่มือ VB.NET
3. หนังสือภาษาไทย

โดยกำหนดว่า หนังสือการเขียนโปรแกรมภาษาจาวา ราคา 350

คู่มือ VB.NET 540 หนังสือภาษาไทย 150

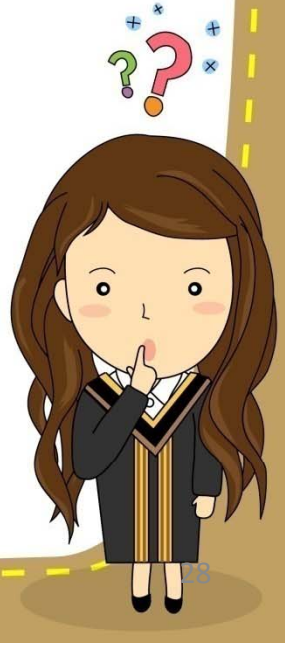


การวนลูปแบบ wh



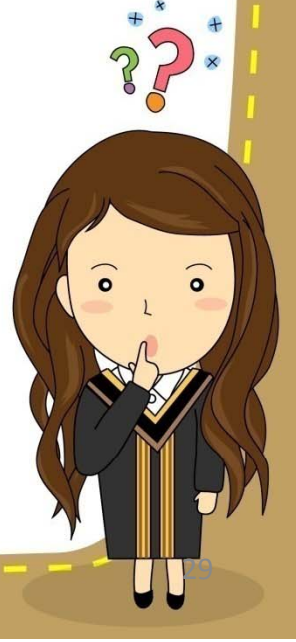
รูปแบบ

```
while (เงื่อนไข)
{
    ประโยคที่ทำซ้ำขณะที่เงื่อนไขเป็นจริง;
}
```

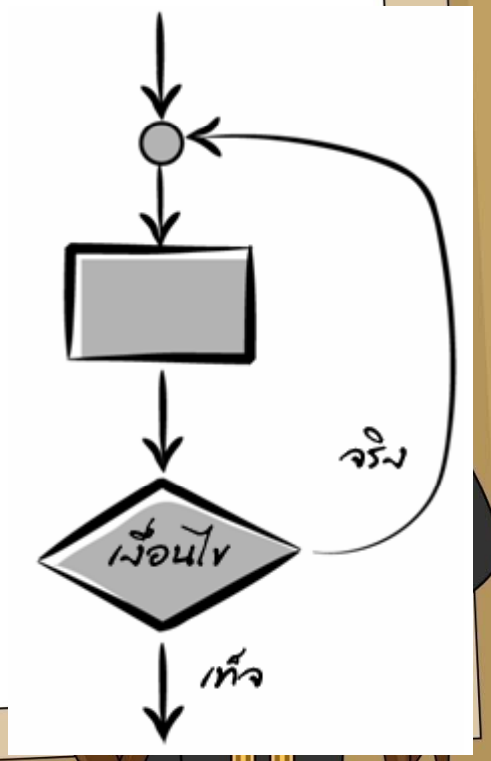


ตัวอย่างการคำนวณดอกเบี้ย

```
int year = 0;

double balance = 100;
while (balance <= 10000)
{
   year++;
   balance *= 1.05;
}
```

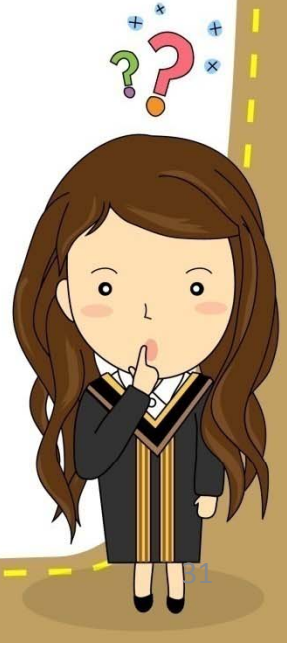


การวนลูปแบบ do-while



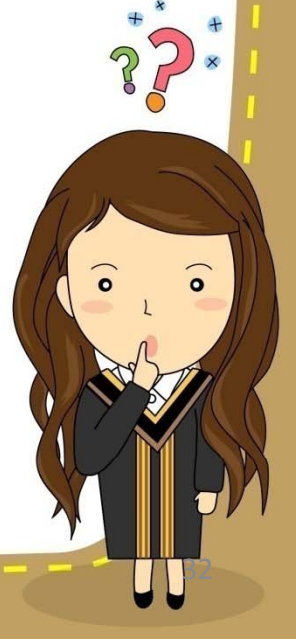
รูปแบบ

```
do
{
    ประโยค;
} while (เงื่อนไข);
```

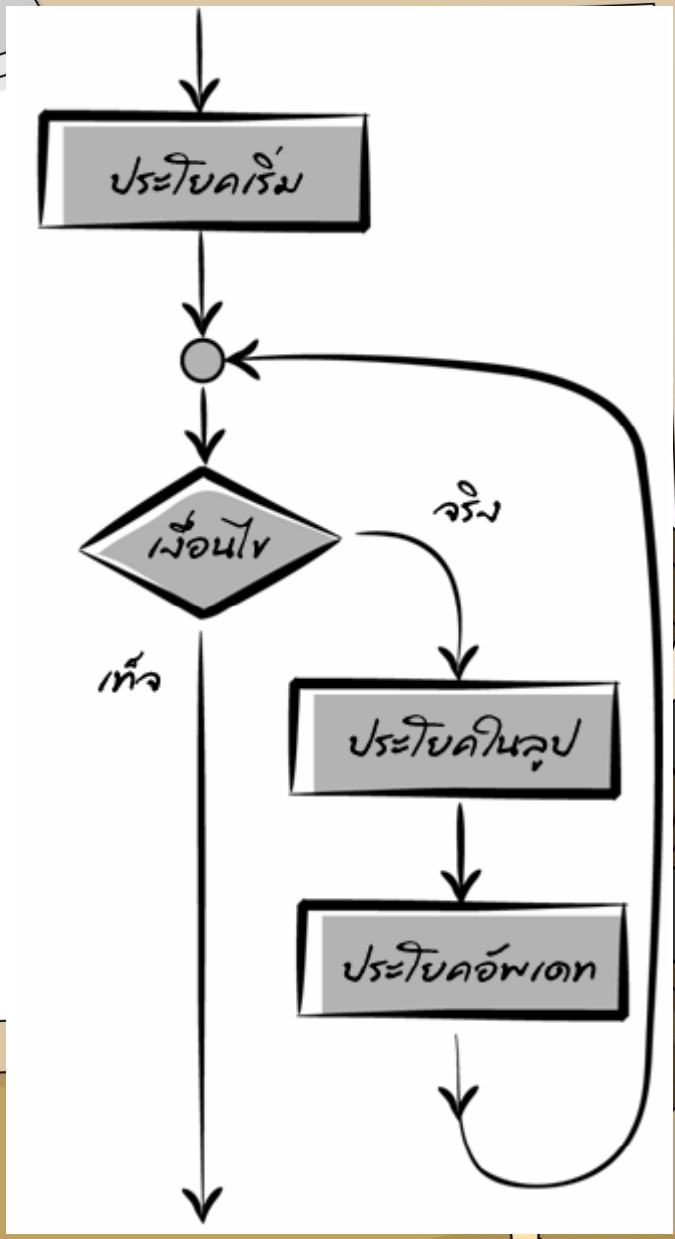


ตัวอย่างวนลูปจนกว่าจะได้ข้อมูลที่ถูกต้อง

```
do
{
    รับข้อมูล;
} while (ข้อมูลไม่ถูกต้อง);
นำข้อมูลที่ถูกต้องไปประมวลผล;
```

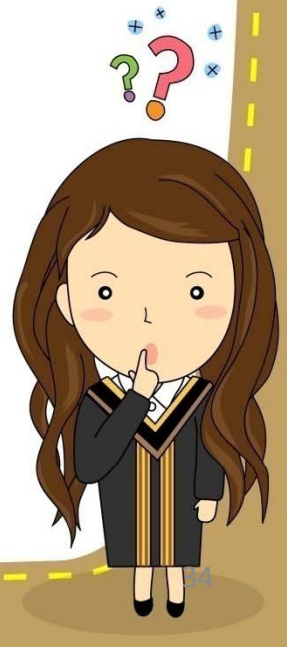


การวนลูปแบบ for



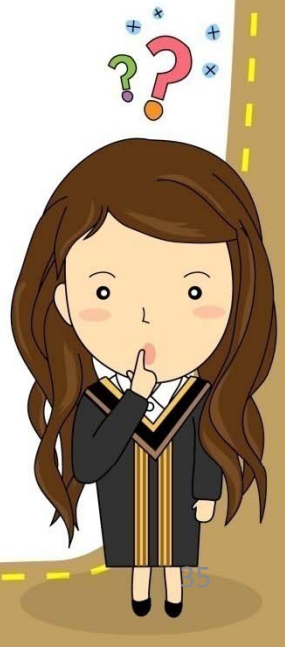
รูปแบบ

```
for (ประโยคเริ่ม; เงื่อนไข; ประโยคอัพเดท)
{
    ประโยคในลูป;
}
```



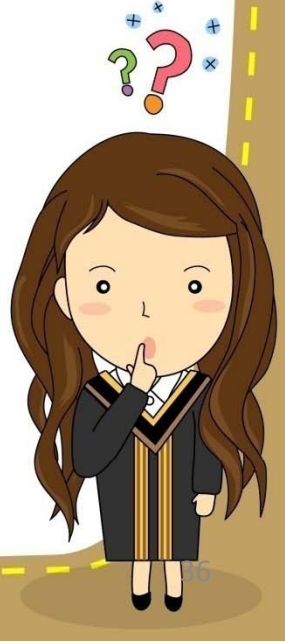
ตัวอย่างวนลูปเป็นจำนวน 5 ครั้ง

```
for (int i = 0; i < 5; i++)
{
    // ประโยคที่จะทำซ้ำ 5 ครั้ง
}
```

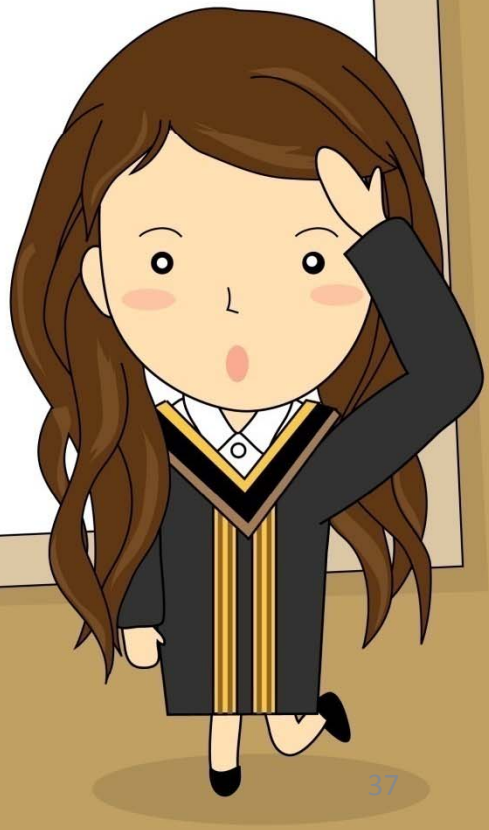


ตัวอย่างการคำนวณยอดเงินในบัญชีเมื่อเวลาผ่านไป 95 ปี

```
double balance = 100;


for (int i = 0; i < 95; i++)
{
    balance *= 1.05;  (balance = balance*1.05)
}

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

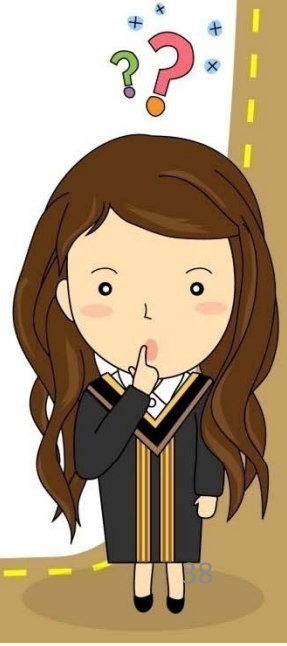


คำสั่ง break



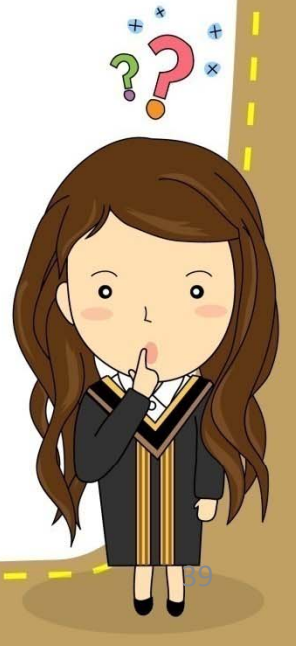
การออกจาก while ด้วยคำสั่ง break

```
while( เงื่อนไข ในการวนลูป)
{
    ....
    if( เงื่อนไขที่จะออกจากลูป )
        break;
    ....
}

คำสั่งหลังลูป;
```

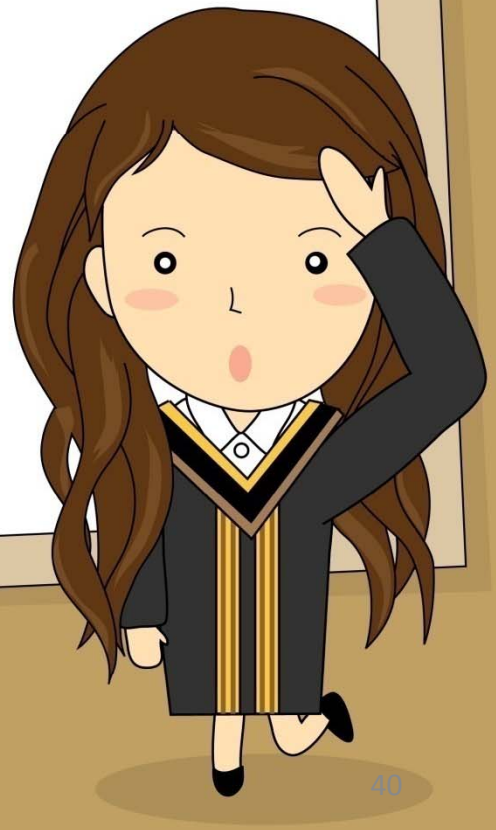


การออกจาก for ด้วยคำสั่ง break

```
for (int i = 0; i < 5; i++)
{
   System.out.print("<");

   if (i == 2)
      break;

   System.out.print(i + ">");
}
```

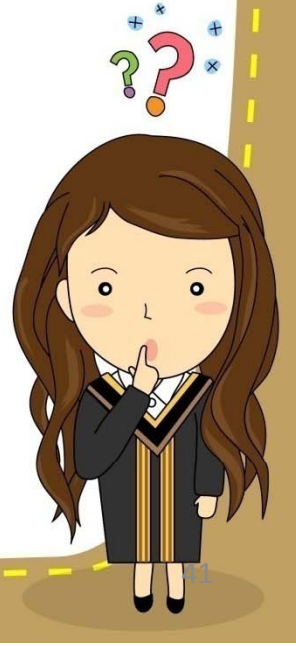


คำสั่ง continue



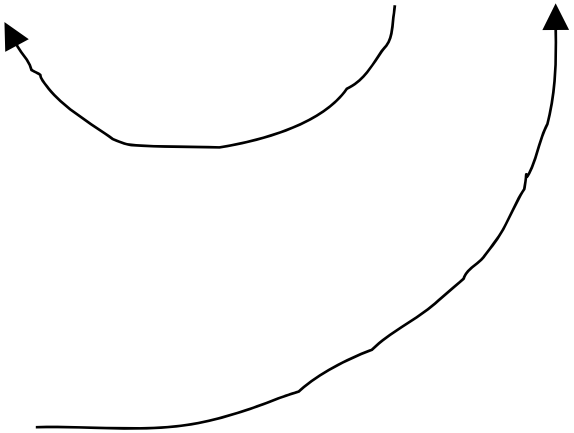
ข้ามไปตรวจสอบเงื่อนไข

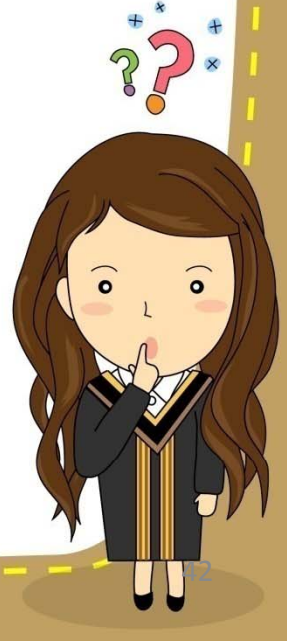
```
while( เงื่อนไขในการวนลูป)
{
        ....
        if( เงื่อนไขที่จะกลับไปต้นลูป )
                continue;
        ....
}
```



ข้ามไปตรวจสอบเงื่อนไข

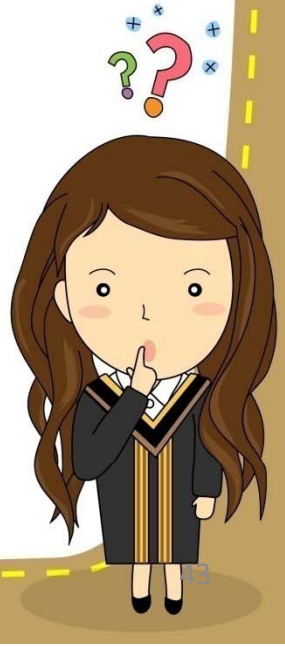
```
for (ประโยคเริ่ม; เงื่อนไข; ประโยคอัพเดท)
{
        ....
        if( เงื่อนไขที่จะกลับไปต้นลูป )
                continue;
        ....
}
```



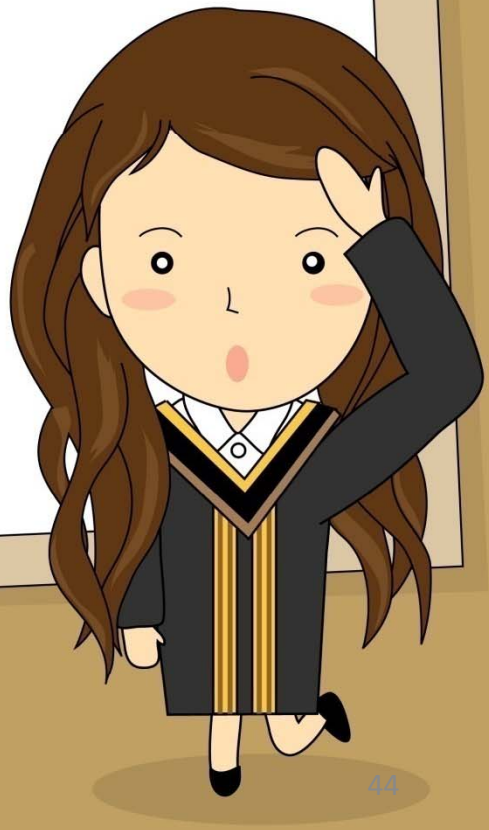


ตัวอย่างโปรแกรม

```
for (int i = 0; i < 5 ;  i++ )
{
   System.out.print("<");

   if (i == 2)
      continue;

   System.out.print(i + ">");
}
```



สรุป

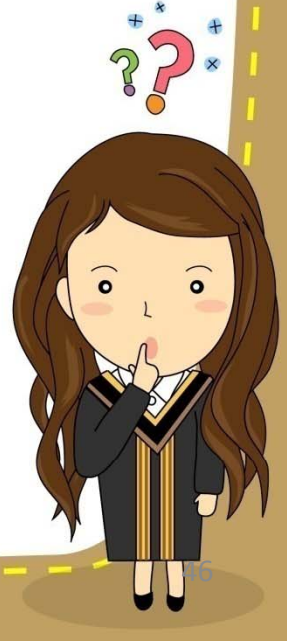


สรุป

- โดยทั่วไปโปรแกรมจะทำงานเรียงลำดับทีละคำสั่ง
- ประโยคควบคุมมีหน้าที่ควบคุมการลำดับการทำงานของคำสั่งต่างๆ
- คำสั่งที่อยู่ถัดจากประโยค if จะทำงานเมื่อเงื่อนไขเป็นจริง

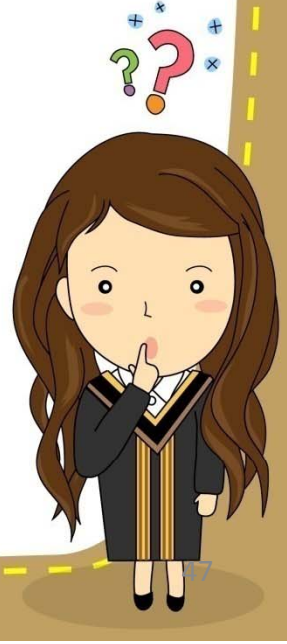
สรุป

- ประโยค **if-else** ทำให้โปรแกรมทำงานอย่างหนึ่งเมื่อเงื่อนไขเป็นจริง และทำงานอีกอย่างหนึ่งเมื่อเงื่อนไขเป็นเท็จ
- ประโยค **if-else if** ใช้กรณีที่มีหลายๆ เงื่อนไข
- ประโยค **switch** ใช้กรณีที่มีหลายๆ เงื่อนไข โดยจะเปรียบเทียบกับค่าคงที่



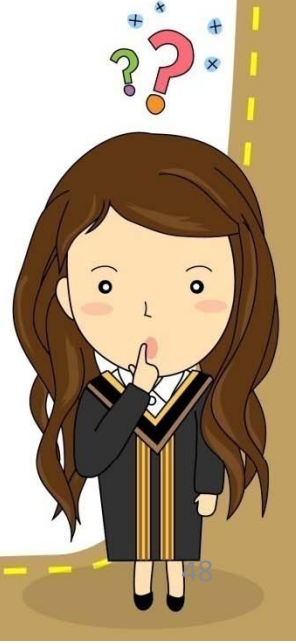
สรุป

- ประโยค **while** ใช้เพื่อวนทำงานซ้ำจนกว่าเงื่อนไขจะเป็นเท็จ
- ประโยค **do-while** จะทำงานในกลุ่มประโยคหลัง **do** หนึ่งครั้ง ตรวจสอบเงื่อนไข และจะวนลูปไปเรื่อยๆจนกว่าเงื่อนไขจะเป็นเท็จ
- ประโยค **for** เป็นลูปที่เหมาะสำหรับการกำหนดจำนวนครั้งการวนซ้ำ



สรุป

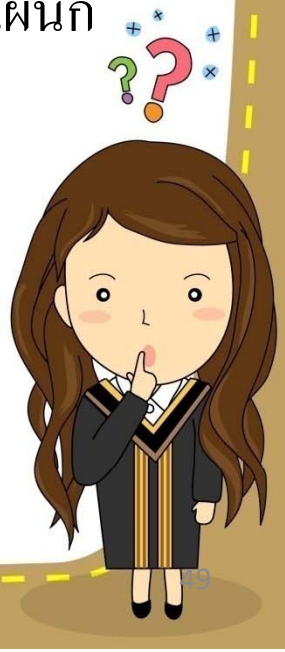
- คำสั่ง **break** ใช้เพื่อออกจากลูป
- คำสั่ง **continue** จะตรวจสอบเงื่อนไขของลูป ถ้าเป็นจริงจะทำงานคำสั่งที่อยู่ต้นลูป แต่ถ้าเป็นเท็จจะออกจากลูป



แบบฝึกหัด

- จงเขียนโปรแกรมเพื่อแสดงชื่อและเงินเดือนโดยใช้คำสั่ง if- else ที่ประมวลผลแสดงผลดังต่อไปนี้

- เงินเดือนมากกว่า 200000 แสดงผล คุณเป็นผู้จัดการ
- เงินเดือนมากว่า 150000 – 200000 แสดงผลคุณเป็นผู้จัดการฝ่าย
- เงินเดือนมากกว่า 100000 – 150000 แสดงผลคุณเป็นผู้จัดการแผนก
- เงินเดือนตั้งแต่ 50000 – 100000 แสดงผลคุณเป็นหัวหน้าหน่วย
- เงินเดือนต่ำกว่า 50000 แสดงผลคุณเป็นพนักงาน



แบบทดสอบ

ให้เขียนโปรแกรมตรวจสอบ **Username = add** และ

Password = add

จำนวน 3 ครั้ง ***ถ้าถูกครั้งใดครั้งหนึ่งแสดงว่า **ATC**

ครั้งที่ 1 แสดงข้อความว่า กรุณาตรวจสอบครั้งที่ 2

ครั้งที่ 2 แสดงข้อความว่า กรุณาตรวจสอบครั้งที่ 3

ครั้งที่ 3 แสดงข้อความว่า ออกจากโปรแกรม

สายอักขระ

คือสายที่ไม่ได้เอาไปคำนวณ

จะเป็นตัวเลขหรือตัวอักษรก็ได้

หัวข้อ

- สายอักขระ
- คลาส String
- คลาส StringBuffer

สายอักขระ

สายอักขระ

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!2");
    }
}
```

ประโยชน์ของสายอักขระ

- บอกผลลัพธ์ของการคำนวณให้เราทราบ
- รายงานขั้นตอนการทำงานของโปรแกรม
- เก็บข้อมูลที่เป็นตัวอักษร ตัวเลข สัญลักษณ์

คลาส String

การใช้งานสตริง

- ประกาศตัวแปร

  String name;

- กำหนดค่า

  name = new String("Smith");

- แสดงผล

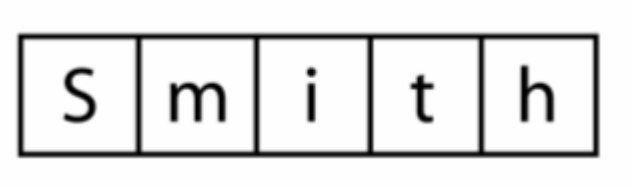
  System.out.println(name);

เรฟเฟอร์เรนซ์ (Reference)

- int n
  - n เป็นข้อมูลชนิดจำนวนเต็ม
- String name
  - ไม่ได้หมายความว่า name เป็นวัตถุ String
  - แต่เป็นการบอกว่า name เป็นเรฟเฟอร์เรนซ์ (reference) หรือตัวที่ใช้อ้างอิงไปที่วัตถุ String

วัตถุและเรฟเฟอร์เรนซ์

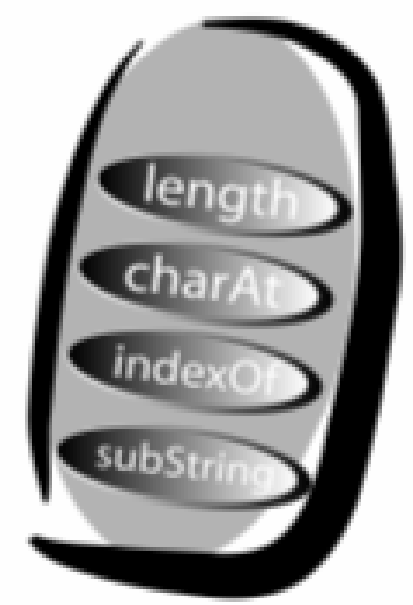
- วัตถุ



- เรฟเฟอร์เรนซ์

การสั่งงานสตริง

- รูปแบบ
  - เรฟเฟอร์เรนซ์.ข้อความ(สิ่งที่ส่งไปพร้อมกับข้อความ)
- ตัวอย่าง
  - name.length();
  - name.charAt(1);



เมธอดที่น่าสนใจ

- length() ความยาวของสตริง
- charAt() ตัวอักษรในตำแหน่งที่กำหนด
  - ตัวอักษรตัวแรกคือตำแหน่งที่ 0
  - ตัวอักษรสุดท้ายคือ length() - 1
- indexOf() ตำแหน่งของสายอักขระในสตริง
- substring() สตริงที่อยู่ในช่วงที่กำหนด

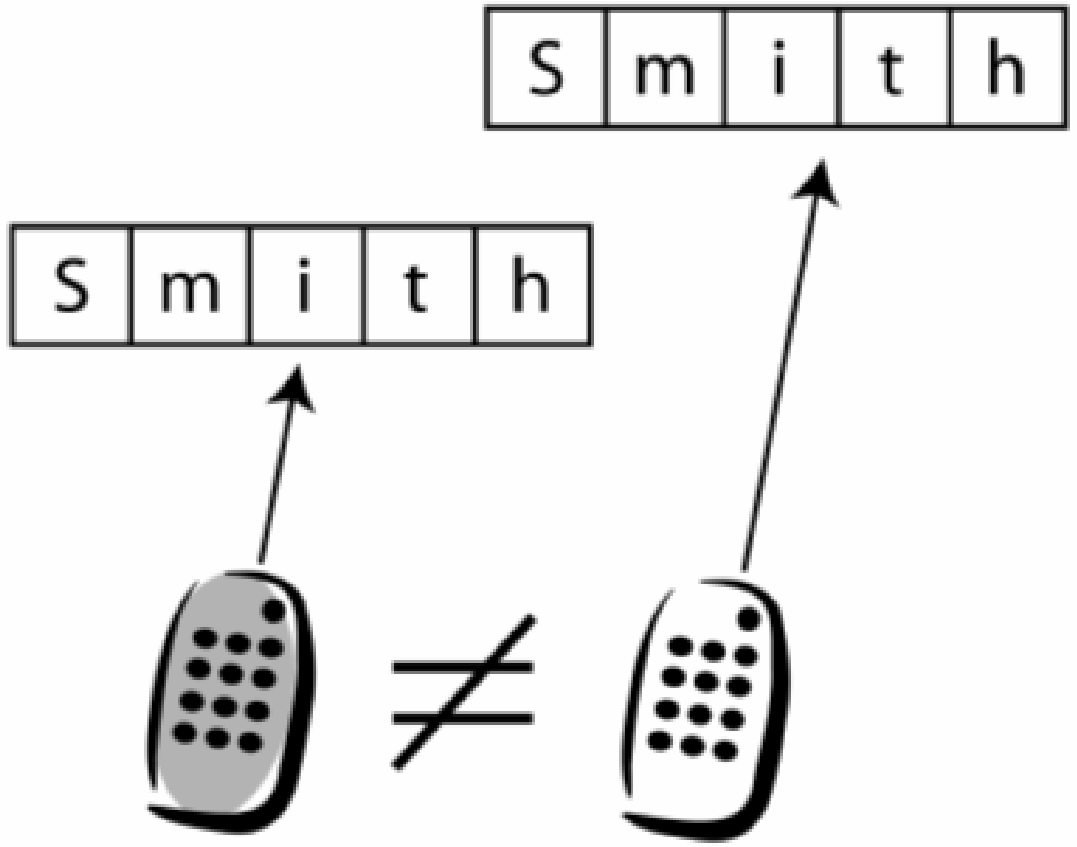
ตัวอย่างการใช้เมธอด

```
public class StringMessage {
    public static void main(String[] args){
        String name ="Smith";

        System.out.println(name.length());
        System.out.println(name.charAt(2));
        System.out.println(name.indexOf("t"));
        System.out.println(name.substring(1, 3));
    }
}
```

การเปรียบเทียบสตริง (แบบผิดๆ)

```
public class StringCompareInCorrect {
    public static void main(String[]  args) {
        String name1 = new String(“Smith”);
        String name2 = new String(“Smith”);
        System.out.println(name1 == name2);
  }
}
```

การเปรียบเทียบสตริง (แบบผิดๆ)



การเปรียบเทียบสตริง

```
public class StringCompare {
    public static void main(String[]  args) {
        String name1 = new String(“Smith”);
        String name2 = new String(“Smith”);
        System.out.println(name1.equals(name2));
    }
}
```

การต่อสตริง

- เมธอด concat()
- เครื่องหมาย +
- เครื่องหมาย +=

เมธอด concat()

```
public class StringConcat1 {
    public static void main(String[] args) {
        String name = "Smith";
        String lastName = " Brown";
        String fullName = name.concat(lastname);
        System.out.println(fullName);
    }
}
```

เครื่องหมาย +

```
public class StringConcat2 {
    public static void main(String[] args) {
        String name = "Smith";
        String lastName = " Brown";
        String fullName = name + lastName;
        System.out.println(fullName);
    }
}
```

เครื่องหมาย +=

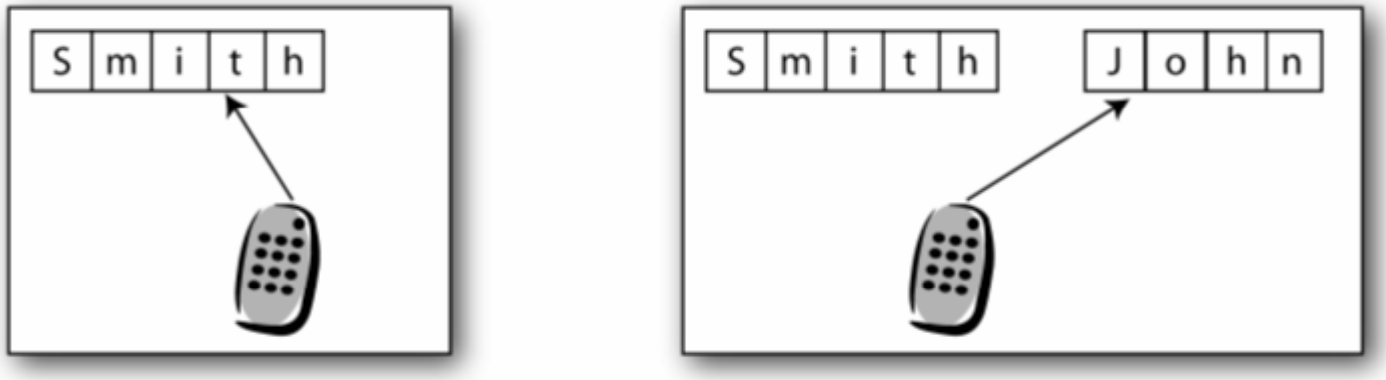
```
public class StringAppend {
    public static void main(String[]  args) {
        String name = "Smith";
        String lastName = " Brown";
   name += lastName;
        System.out.println(name);
  }
}
```

ความมั่นคงของสตริง (Immutability)

- ไม่สามารถเปลี่ยนแปลงค่าของสายอักขระได้เลย หลังจากที่เราได้สร้างมันขึ้นมา
- ในบรรดาข้อความ (message) ที่วัตถุ String รู้จักนั้น ไม่มีข้อความใดเลยที่ทำให้สายอักขระเปลี่ยนแปลงได้
  - length()
  - substring()

สตริงไม่เปลี่ยนแปลง

```
String str;
str = new String("Smith");
str = new String("John");
```



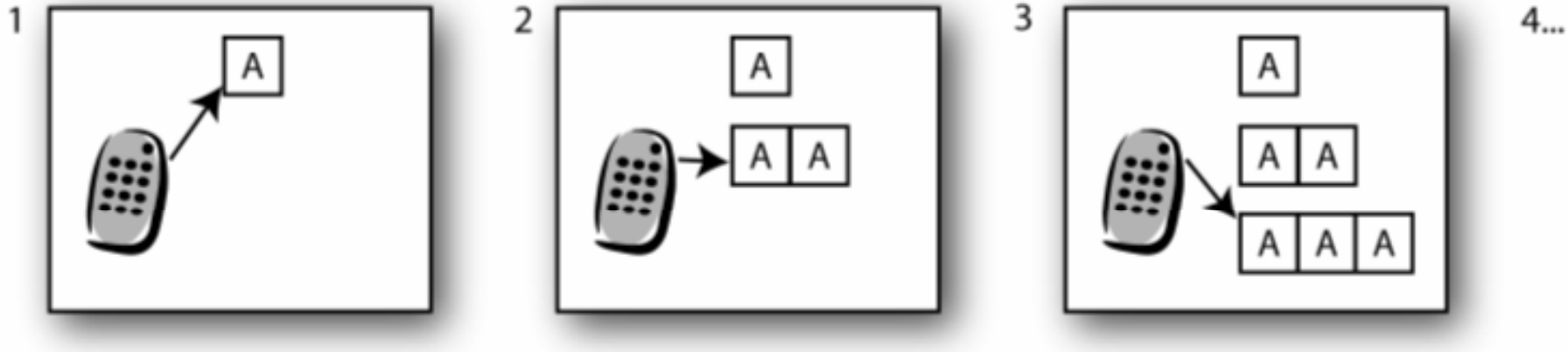
สตริงไม่เปลี่ยนแปลง

```
String str = "Smith";
System.out.println(str.toUpperCase());
System.out.println(str);
```

สตริงไม่เปลี่ยนแปลง

```
public class StringDeficiency {
    public static void main(String[] args)
    {
        String str = "A";
        for (int i = 0; i < 100; i++)
            str += "A";
        System.out.println(str);
    }
}
```

สตริงไม่เปลี่ยนแปลง



คลาส StringBuffer

ข้อดีของ StringBuffer

- ประหยัดทั้งหน่วยความจำและเวลาในการประมวลผล
  - วัตถุ StringBuffer สามารถเปลี่ยนแปลงค่าในสายอักขระที่มันนำเสนอได้เอง โดยไม่ต้องสร้างวัตถุขึ้นมาใหม่

เมธอดที่น่าสนใจ

- append() การต่อสายอักขระ
- insert() การแทรกสายอักขระ
- delete() การลบสายอักขระย่อย

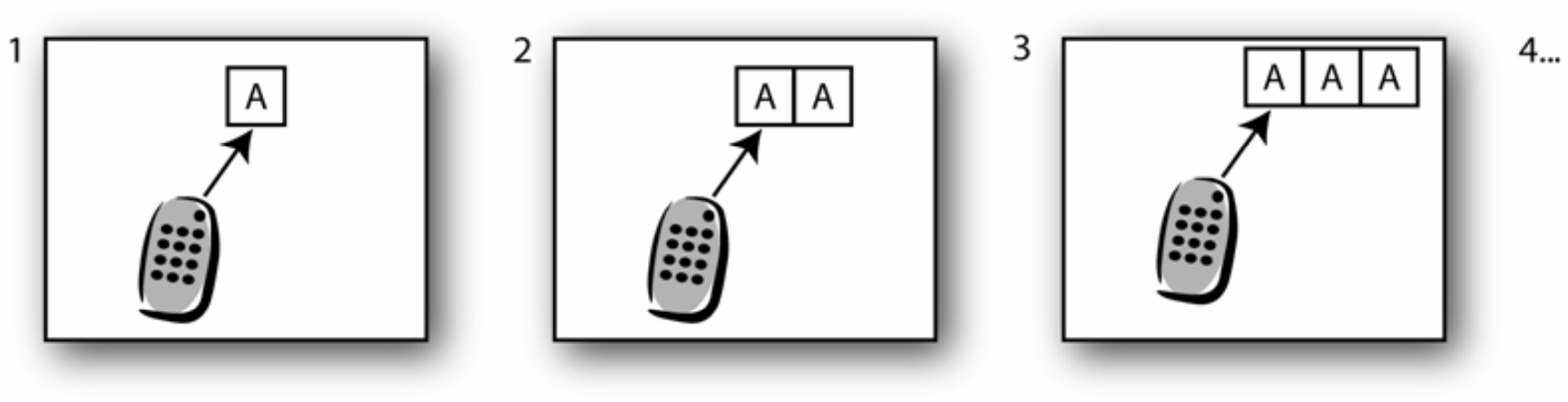
ตัวอย่างการใช้เมธอด

```
StringBuffer sb =new StringBuffer("John");
System.out.println(sb);

sb.append("Hunter");
System.out.println(sb);

sb.insert(4, "y");
System.out.println(sb);

sb.delete(2, 4);
System.out.println(sb);
```

ทดลองเปลี่ยนแปลง StringBuffer

```
public class StringBufferAppend {
    public static void main(String[] args){
        StringBuffer sb;
        sb =new StringBuffer("A");

        for (int i =0; i < 100; i++)
            sb.append("A");

        System.out.println(sb);
    }
}
```

ทดลองเปลี่ยนแปลง StringBuffer

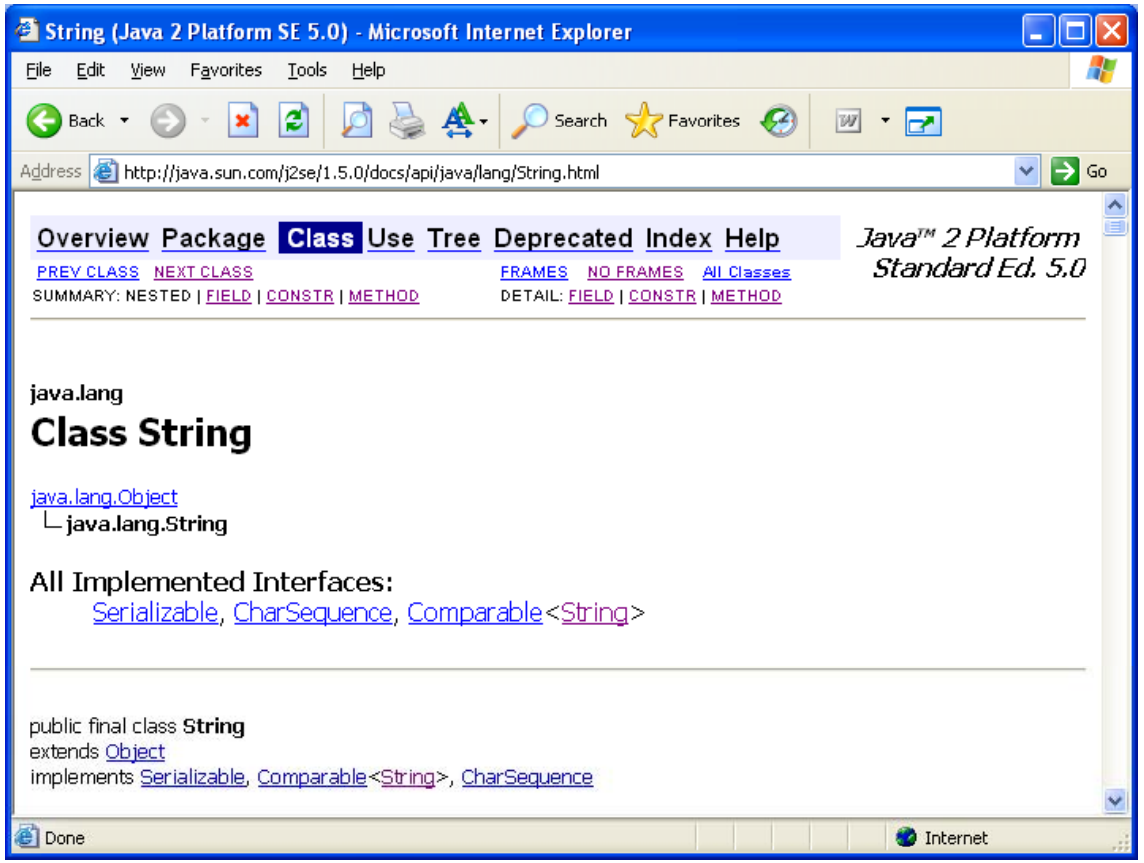


StringBuilder

- มีเมธอดที่เหมือนกับ StringBuffer
- ทำงานได้เร็วกว่า StringBuffer
- StringBuffer ปลอดภัยสำหรับการทำงานในแบบ multiple threads

Java Doc

เอกสารอ้างอิง



แหล่งเอกสาร

- http://java.sun.com/j2se/1.5.0/docs/api
- http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html
- http://java.sun.com/j2se/1.5.0/docs/api/java/lang/StringBuffer.html

สรุป

สรุป

- สายอักขระมีไว้เพื่อการแสดงผลข้อความ
- สายอักขระได้ถูกนำเสนอด้วยวัตถุ String และ StringBuffer
- เราสามารถใช้งานวัตถุได้โดยผ่านทางเรฟเฟอร์เรนซ์
- เรฟเฟอร์เรนซ์เปรียบเสมือนรีโมทคอนโทรล
- วัตถุเปรียบได้กับสิ่งที่รีโมทนั้นควบคุมอยู่
- การส่งข้อความหาวัตถุเปรียบได้กับการกดปุ่มรีโมทคอนโทรล

สรุป

- การเปรียบเทียบสตริงทำได้โดยการส่งข้อความ equals( )
- การต่อสตริงนอกจากจะใช้การส่งข้อความ concat() แล้ว ยังสามารถใช้เครื่องหมาย + ได้อีกด้วย
- สายอักขระของวัตถุ String ไม่สามารถถูกเปลี่ยนแปลงได้
- ถ้าต้องการเปลี่ยนแปลงค่าในสายอักขระบ่อยๆ ควรใช้วัตถุ StringBuffer

สายอักขระ

คือสายที่ไม่ได้เอาไปคำนวณ

จะเป็นตัวเลขหรือตัวอักษรก็ได้

หัวข้อ

- สายอักขระ
- คลาส String
- คลาส StringBuffer

สายอักขระ

สายฮักษระ

```
public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello, World! 2");

    }

}
```

ประโยชน์ของสายอักขระ

- บอกผลลัพธ์ของการคำนวณให้เราทราบ
- รายงานขั้นตอนการทำงานของโปรแกรม
- เก็บข้อมูลที่เป็นตัวอักษร ตัวเลข สัญลักษณ์

คลาส String

การใช้งานสตริง

- ประกาศตัวแปร

```
String name;
```

- กำหนดค่า

```
name = new String("Smith");
```

- แสดงผล

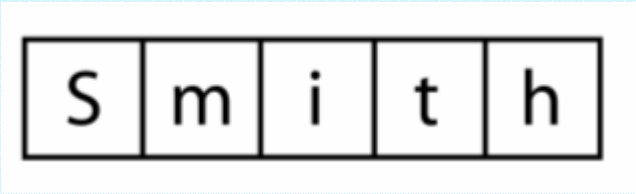
```
System.out.println(name);
```

เรฟเฟอร์เรนซ์ (Reference)

- int n
  - n เป็นข้อมูลชนิดจำนวนเต็ม
- String name
  - ไม่ได้หมายความว่า name เป็นวัตถุ String
  - แต่เป็นการบอกว่า name เป็นเรฟเฟอร์เรนซ์ (reference) หรือตัวที่ใช้อ้างอิงไปที่วัตถุ String

วัตถุและเรฟเฟอร์เรนซ์

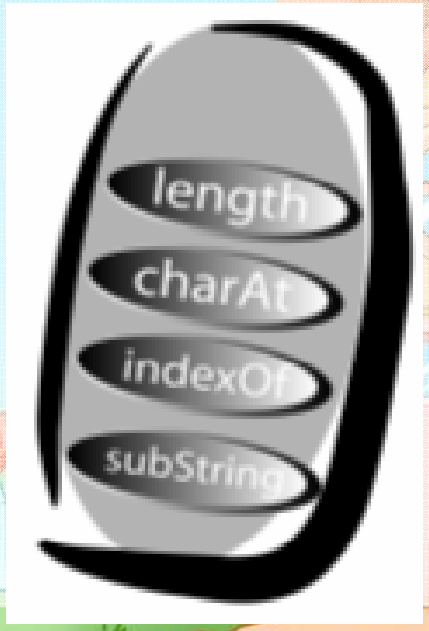
- วัตถุ



- เรฟเฟอร์เรนซ์

การสั่งงานสตริง

- รูปแบบ
  - เรฟเฟอร์เรนซ์.ข้อความ(สิ่งที่ส่งไปพร้อมกับข้อความ)
- ตัวอย่าง
  - name.length();
  - name.charAt(1);



เมธอดที่น่าสนใจ

- length() ความยาวของสตริง
- charAt() ตัวอักษรในตำแหน่งที่กำหนด
  - ตัวอักษรตัวแรกคือตำแหน่งที่ 0
  - ตัวอักษรสุดท้ายคือ length() - 1
- indexOf() ตำแหน่งของสายอักขระในสตริง
- substring() สตริงที่อยู่ในช่วงที่กำหนด

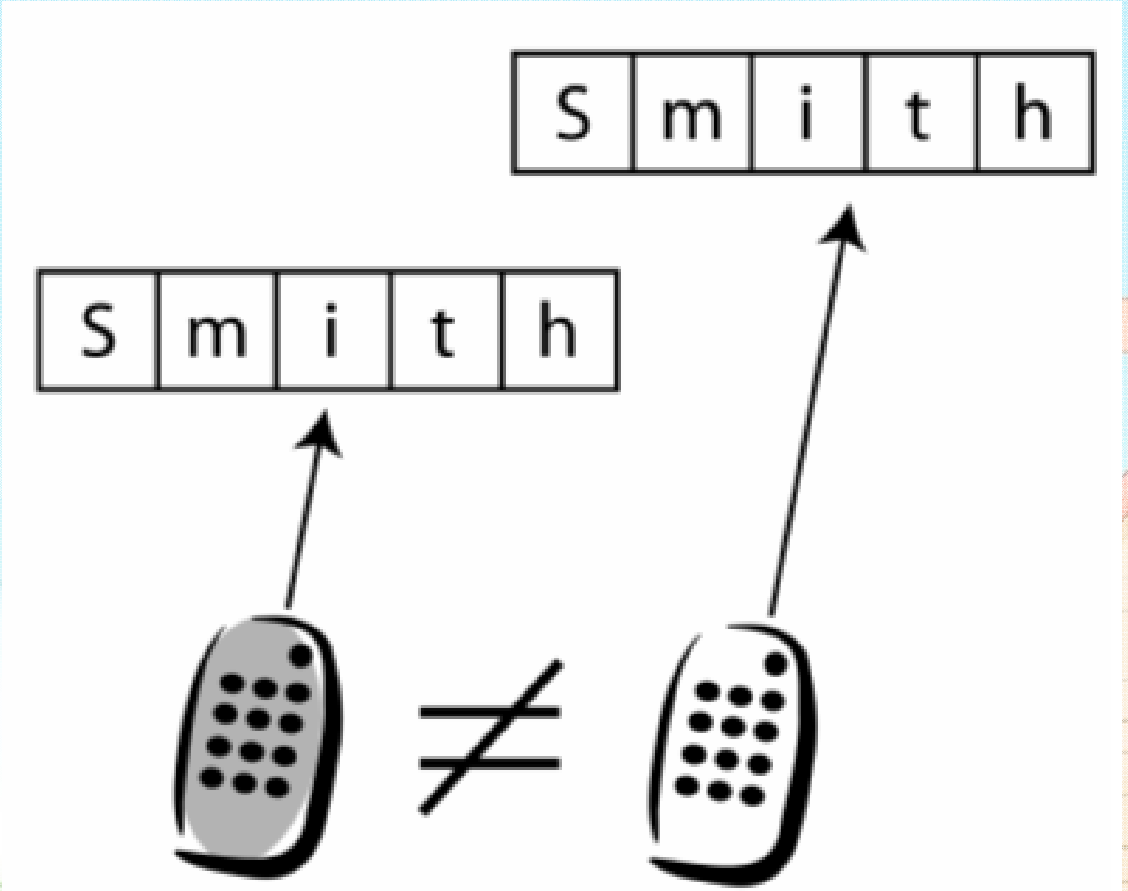
ตัวอย่างการใช้เมธอด

```
public class StringMessage {
    public static void main(String[] args) {
        String name = "Smith";
        System.out.println(name.length());
        System.out.println(name.charAt(2));
        System.out.println(name.indexOf("t"));
        System.out.println(name.substring(1, 3));
    }
```

การเปรียบเทียบสตริง (แบบผิดๆ)

```
public class StringCompareInCorrect {
    public static void main(String[]  args) {
        String name1 = new String("Smith");
        String name2 = new String("Smith");
        System.out.println(name1 == name2);
    }
}
```

การเปรียบเทียบสตริง (แบบผิดๆ)



การเปรียบเทียบสตริง

```
public class StringCompare {
    public static void main(String[]  args) {
        String name1 = new String(“Smith”);
        String name2 = new String(“Smith”);
        System.out.println(name1.equals(name2));
    }
}
```

การต่อสตริง

- เมธอด concat()
- เครื่องหมาย +
- เครื่องหมาย +=

เมธอด concat()

```
public class StringConcat1 {
    public static void main(String[] args) {
        String name = "Smith";
        String lastName = " Brown";
        String fullName = name.concat(lastname);
        System.out.println(fullName);
    }
}
```

เครื่องหมาย +

```
public class StringConcat2 {
    public static void main(String[] args) {
        String name = "Smith";
        String lastName = " Brown";
        String fullName = name + lastName;
        System.out.println(fullName);
    }
}
```

เครื่องหมาย +=

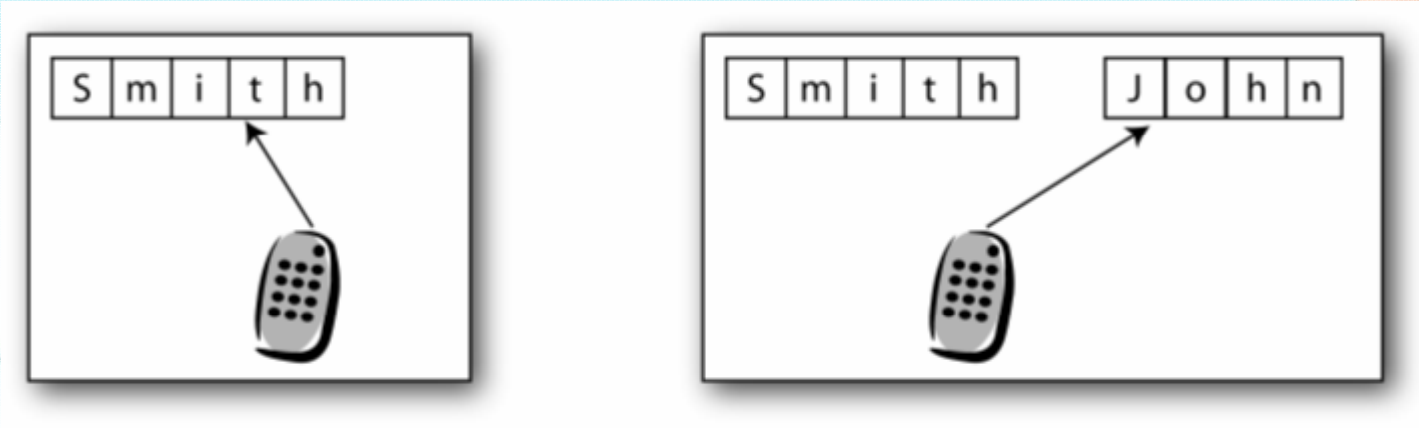
```
public class StringAppend {
    public static void main(String[]  args) {
        String name = "Smith";
        String lastName = " Brown";
        name += lastName;
        System.out.println(name);
    }
}
```

ความมั่นคงของสตริง (Immutability)

- ไม่สามารถเปลี่ยนแปลงค่าของสายอักขระได้เลยหลังจากที่เราได้สร้างมันขึ้นมา
- ในบรรดาข้อความ (message) ที่วัตถุ String รู้จักนั้น ไม่มีข้อความใดเลยที่ทำให้สายอักขระเปลี่ยนแปลงได้
  - length()
  - substring()

สตริงไม่เปลี่ยนแปลง

```
String str;
str = new String("Smith");
str = new String("John");
```



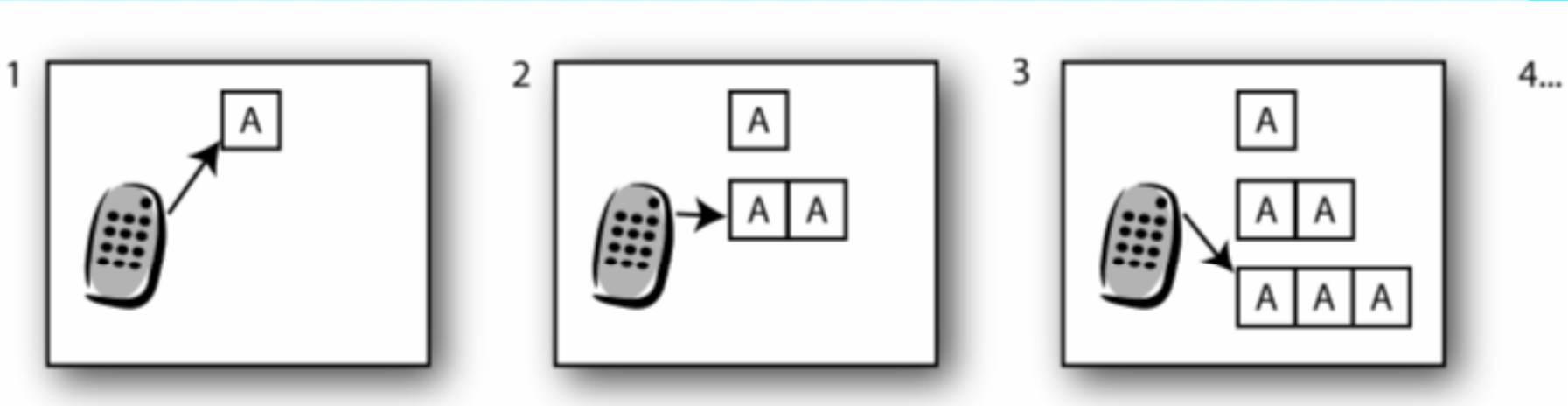
สตริงไม่เปลี่ยนแปลง

```
String str = "Smith";
System.out.println(str.toUpperCase());
System.out.println(str);
```

สตริงไม่เปลี่ยนแปลง

```
public class StringDeficiency {
    public static void main(String[] args)
    {
        String str = "A";
        for (int i = 0; i < 100; i++)
            str += "A";
        System.out.println(str);
    }
}
```

สตริงไม่เปลี่ยนแปลง



คลาส StringBuffer

ข้อดีของ StringBuffer

- ประหยัดทั้งหน่วยความจำและเวลาในการประมวลผล
  - วัตถุ StringBuffer สามารถเปลี่ยนแปลงค่าในสายอักขระที่มันนำเสนอได้เอง โดยไม่ต้องสร้างวัตถุขึ้นมาใหม่

เมธอดที่น่าสนใจ

- append() การต่อสายอักขระ
- insert() การแทรกสายอักขระ
- delete() การลบสายอักขระย่อย

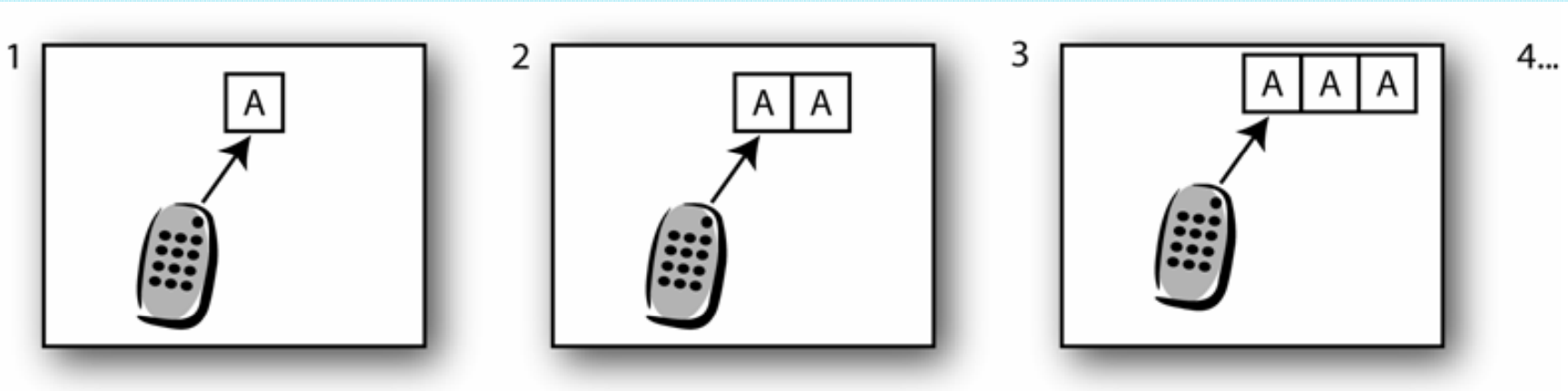
ตัวอย่างการใช้เมธอด

```
StringBuffer sb = new StringBuffer("John");
System.out.println(sb);

sb.append(" Hunter");
System.out.println(sb);

sb.insert(4, "y");
System.out.println(sb);

sb.delete(2, 4);
System.out.println(sb);
```

ทดลองเปลี่ยนแปลง StringBuffer

```
public class StringBufferAppend {
    public static void main(String[] args) {
        StringBuffer sb;
        sb = new StringBuffer("A");

        for (int i = 0; i < 100; i++)
            sb.append("A");

        System.out.println(sb);
    }
}
```

ทดลองเปลี่ยนแปลง StringBuffer



StringBuilder

- มีเมธอดที่เหมือนกับ StringBuffer
- ทำงานได้เร็วกว่า StringBuffer
- StringBuffer ปลอดภัยสำหรับการทำงานในแบบ multiple threads

Java Doc

เอกสารอ้างอิง

แหล่งเอกสาร

- http://java.sun.com/j2se/1.5.0/docs/api
- http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html
- http://java.sun.com/j2se/1.5.0/docs/api/java/lang/StringBuffer.html

สรุป

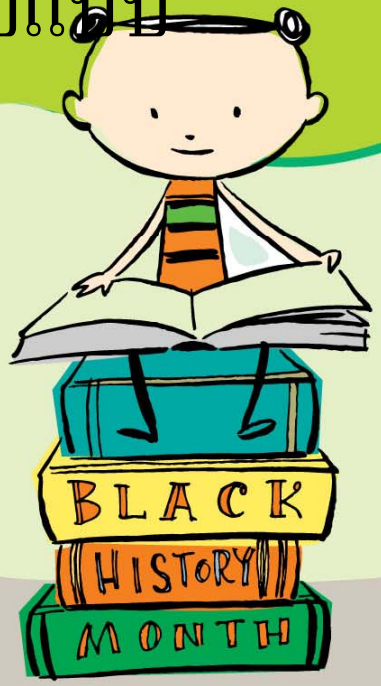
สรุป

- สายอักขระมีไว้เพื่อการแสดงผลข้อความ
- สายอักขระได้ถูกนำเสนอด้วยวัตถุ String และ StringBuffer
- เราสามารถใช้งานวัตถุได้โดยผ่านทางเรฟเฟอร์เรนซ์
- เรฟเฟอร์เรนซ์เปรียบเสมือนรีโมทคอนโทรล
- วัตถุเปรียบได้กับสิ่งที่รีโมทนั้นควบคุมอยู่
- การส่งข้อความหาวัตถุเปรียบได้กับการกดปุ่มรีโมทคอนโทรล

สรุป

- การเปรียบเทียบสตริงทำได้โดยการส่งข้อความ equals( )
- การต่อสตริงนอกจากจะใช้การส่งข้อความ concat() แล้ว ยังสามารถใช้เครื่องหมาย + ได้อีกด้วย
- สายอักขระของวัตถุ String ไม่สามารถถูกเปลี่ยนแปลงได้
- ถ้าต้องการเปลี่ยนแปลงค่าในสายอักขระบ่อยๆ ควรใช้วัตถุ StringBuffer

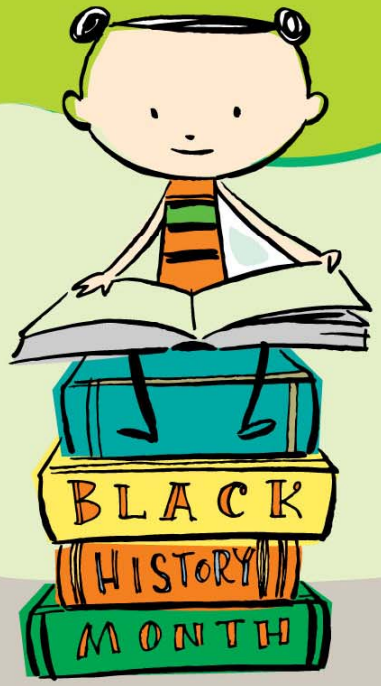
คณิตศาสตร์ ตัวห่อหุ้ม และการจัดรูปแบบ



หัวข้อ

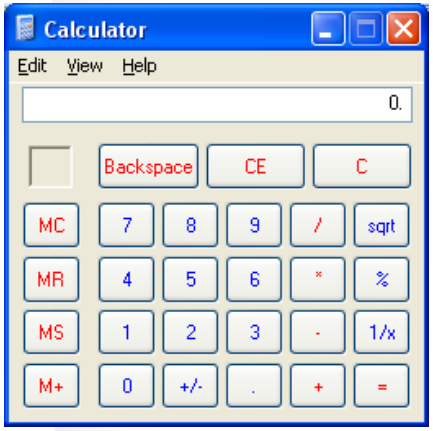
- คลาส Math
- การรับข้อมูลจากผู้ใช้
- ตัวห่อหุ้ม (Wrappers)
- การจัดรูปแบบ

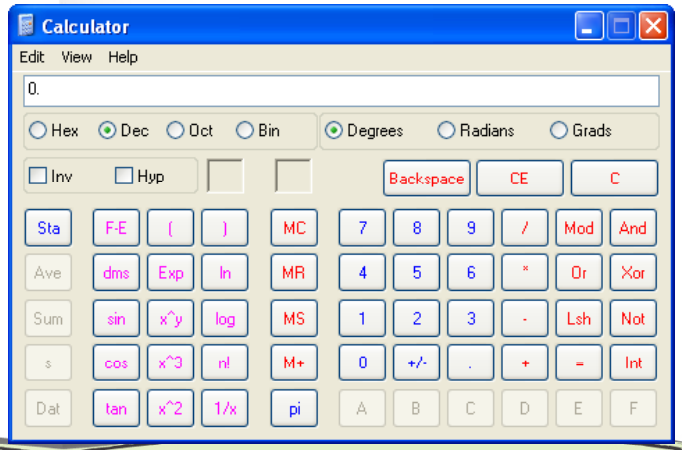
คลาส Math



การคำนวณ

- ง่าย
  - + - * / %
- ซับซ้อน
  - คลาส Math





เมธอดในคลาส Math

- sqrt() ใช้ในการหาค่ารากที่สอง
  - Math.sqrt(9) จะมีค่าเท่ากับ 3
- pow() ใช้ในการหาค่ายกกำลัง
  - Math.pow(2, 4) จะเท่ากับ 2^4 หรือ 16 นั่นเอง
- abs() ใช้ในการหาค่าสัมบูรณ์
  - Math.abs(-4) จะเท่ากับ 4

เมธอดในคลาส Math

- ceil() ใช้ในการหาค่าเลขจำนวนเต็มน้อยที่สุดที่มากกว่าตัวเลขที่ระบุไว้
  - Math.ceil(3.27) จะมีค่าเท่ากับ 4
  - Math.ceil(-3.27) จะมีค่าเท่ากับ -3
- floor() ใช้ในการหาค่าเลขจำนวนเต็มที่มากที่สุดที่น้อยกว่าตัวเลขที่ระบุไว้
  - Math.floor(3.27) จะมีค่าเท่ากับ 3
  - Math.floor(-3.27) จะมีค่าเท่ากับ -4

เมธอดในคลาส Math

- round() ใช้ในการปัดเศษทศนิยมโดยจะหาเลขจำนวนเต็มที่ใกล้กับตัวเลขที่ระบุมากที่สุด
  - Math.round(3.27) จะมีค่าเท่ากับ 3
  - Math.round(-3.27) จะมีค่าเท่ากับ -3
- min() ใช้หาค่าตัวเลขที่น้อยที่สุดระหว่างเลขสองตัวตามที่ระบุไว้
  - Math.min(3, 5) จะได้ค่าเท่ากับ 3
- max() ใช้หาค่าตัวเลขที่มากที่สุดระหว่างเลขสองตัวตามที่ระบุไว้
  - Math.max(3, 5) จะได้ค่าเท่ากับ 5

เมธอดในคลาส Math

- toDegrees() แปลงมุม radian เป็นมุม degree
  - Math.toDegrees(Math.PI) มีค่าเท่ากับ 180.0
- toRadians() แปลงมุม degree เป็นมุม radian
  - Math.toRadians(180) มีค่าเท่ากับ 3.141592653589793
- sin()
  - Math.sin(Math.toRadians(90)) มีค่าเท่ากับ 1

ปริมาตรวัตถุทรงกลม

- ปริมาตร = $(4 \div 3) \times \prod \times$ รัศมี^{3}

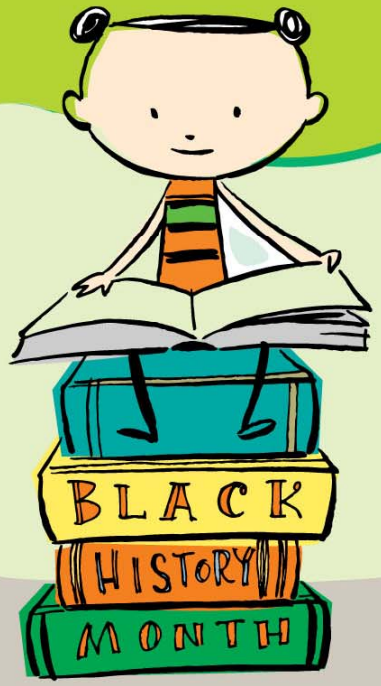
```
public class BallVolume {
    public static void main(String[] args) {
        double radius = 10;
        double volume = 4.0 / 3.0 *
                        Math.PI *
                        Math.pow(radius, 3);

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

เลขสุ่ม

- เมธอด random()
  - return ค่าสุ่มในช่วง 0 เกือบถึง 1

```
for (int i = 0; i < 10; i++){
    double r = Math.random() * 100;
    int rand = (int) r;

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

การรับข้อมูลจากผู้ใช้

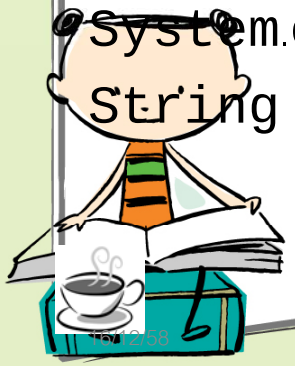


คลาส java.util.Scanner

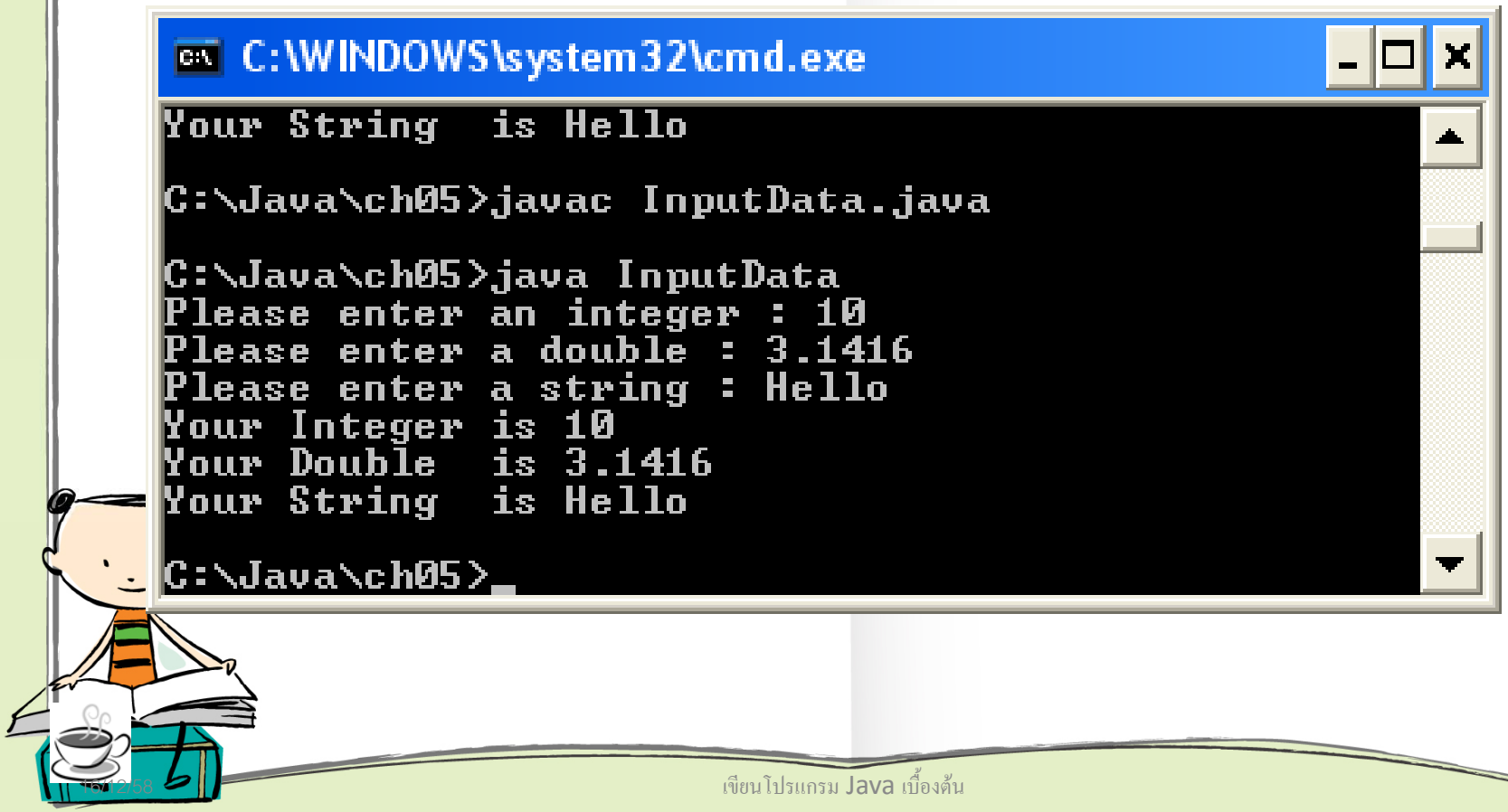
- คอนสตรักเตอร์
  - java.util.Scanner(InputStream src);
- เมธอด
  - nextInt();
  - nextDouble();
  - next();

ตัวอย่าง

```
java.util.Scanner sc;
sc = new java.util.Scanner(System.in);

System.out.print("Please enter an integer :");
int i = sc.nextInt();

System.out.print("Please enter a double :");
double d = sc.nextDouble();

System.out.print("Please enter a string :");
String s = sc.next();
```



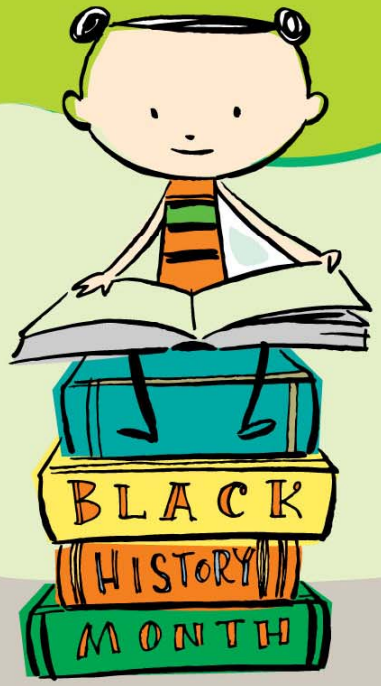
ผลการทำงาน


C:\WINDOWS\system32\cmd.exe
Your String is Hello
C:\Java\ch05>javac InputData.java
C:\Java\ch05>java InputData
Please enter an integer : 10
Please enter a double : 3.1416
Please enter a string : Hello
Your Integer is 10
Your Double is 3.1416
Your String is Hello
C:\Java\ch05>

ถ้าไม่อยากใช้ชื่อเต็ม

```
import java.util.Scanner;

class MyClass {
    public static void main(String[] a) {
        Scanner sc = new Scanner(System.in);
        ...
    }
}
```

ตัวห่อหุ้ม (Wrappers)



ตัวห่อหุ้ม

- ห่อชนิดข้อมูลพื้นฐาน
  - เพื่อใส่ใน Collection
- แปลง String เป็นชนิดข้อมูลพื้นฐาน
- แปลงจากชนิดข้อมูลพื้นฐานเป็น String
- มีค่าคงที่ของค่าที่มากสุดและน้อยของชนิดข้อมูลพื้นฐาน

คลาสที่เป็นตัวห่อหุ้ม

ชนิดข้อมูลพื้นฐาน	คลาสในกลุ่ม Wrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

การสร้างวัตถุ Wrapper

```
Boolean     bo = new Boolean(true);
Boolean     bo = new Boolean("true");

Character   c  = new Character('c');

Byte        by = new Byte((byte)20);
Byte        by = new Byte("20");

Short       s  = new Short((short)20);
Short       s  = new Short("20");
```

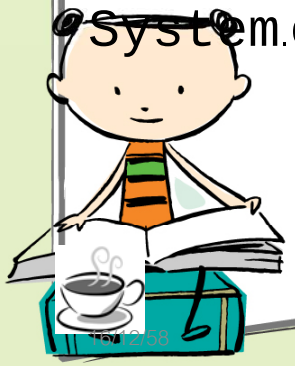
การแปลง Wrapper กลับเป็นชนิดข้อมูลพื้นฐาน

- รูปแบบ
  - *ตัวแปรชนิดข้อมูลพื้นฐาน* = *ตัวห่อหุ้ม.ชนิดข้อมูลพื้นฐาน*Value();
- ตัวอย่าง

```
String s = "20";
Integer i = new Integer(s);
int number = i.intValue();
```

ตัวอย่าง

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

System.out.print("Please enter a number :");
String s = sc.next();

Integer i = new Integer(s);
int number = i.intValue();

System.out.print("Your number plus 10 equals ");
System.out.println(number + 10);
```



การแปลง Wrapper เป็นสตริง

- รูปแบบ
  - *เรฟเฟอร์เรนซ์สตริง* = *ตัวห่อหุ้ม*.toString();
- ตัวอย่าง

```
int number = 20;
Integer i = new Integer(number);
String s = i.toString();
```

การแปลงสตริงเป็นชนิดข้อมูลพื้นฐาน

- รูปแบบ
  - *ตัวแปรชนิดข้อมูลพื้นฐาน* = *คลาสห่อหุ้ม*.parse*ชนิดข้อมูลพื้นฐาน*();
- ตัวอย่าง

```
Scanner sc = new Scanner(System.in);
System.out.print("Please enter a number : ");
int number = Integer.parseInt(sc.next());
```

การแปลงชนิดข้อมูลพื้นฐาน เป็นสตริง

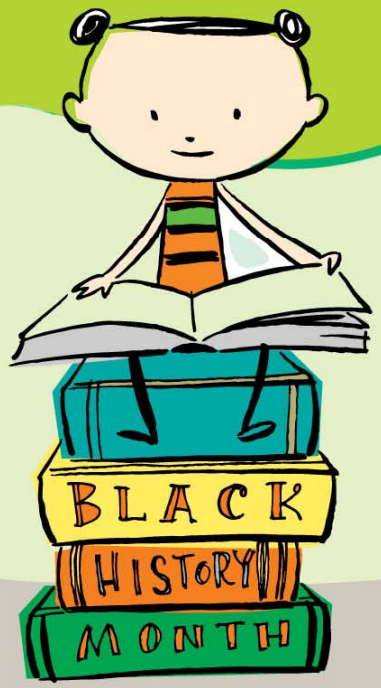
- รูปแบบ
  - *เรฟเฟอร์เรนซ์สตริง* = *คลาสห่อหุ้ม*.toString(*ข้อมูลพื้นฐาน*);
- ตัวอย่าง

```
Scanner sc = new Scanner(System.in);
System.out.print("Please enter a number : ");
String s = Integer.toString(sc.nextInt());
```

ค่าคงที่ในคลาส Wrapper

- Integer
  - Integer.MIN_VALUE = -2147483648
  - Integer.MAX_VALUE = 2147483647
- Double
  - Double.MIN_VALUE = 4.9E-324
  - Double.MAX_VALUE = 1.79769313486231570E308

การจัดรูปแบบ



การจัดรูปแบบด้วยเมธอด printf()

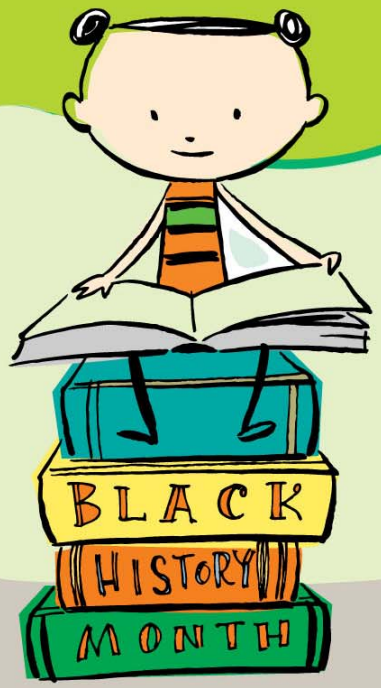
- รูปแบบ
  - *ตัวแสดงผล*.printf("*รูปแบบ*", *ตัวแปร1*, *ตัวแปร2*, ...);
- ตัวอย่าง

```
int     i = 3277;
double  d = 132324.25;
System.out.printf("%d\n", i);
System.out.printf("%x\n", i);
System.out.printf("%f\n", d);
System.out.printf("%16f\n", d);
System.out.printf("%16.2f\n", d);
System.out.printf("%,16.2f\n", d);
```

การจัดรูปแบบด้วยคลาสในกลุ่ม Format

```
double d = 37625.72558;

NumberFormat nf;

nf = NumberFormat.getCurrencyInstance();
System.out.println(nf.format(d)); // ฿37,625.73

nf = NumberFormat.getIntegerInstance();
System.out.println(nf.format(d)); // 37,625
```

สรุป



สรุป

- คลาส Math มีเมธอดสำหรับการคำนวณทางด้านคณิตศาสตร์ชั้นสูง
- คลาสประเภท Wrapper ใช้แปลงตัวแปรชนิดข้อมูลพื้นฐานไปเป็นข้อความและกลับกัน
- การแปลงวัตถุ String ไปเป็นจำนวนเต็มใช้เมธอด Integer.parseInt()
- การแปลงจำนวนเต็มไปเป็นวัตถุ String ใช้เมธอด Integer.toString()

สรุป

- เมธอด println() ใช้สำหรับแสดงผลลัพธ์ออกทางหน้าจอ
- เมธอด printf() ใช้แสดงผลลัพธ์เช่นเดียวกับเมธอด println() แต่สามารถจัดรูปแบบการแสดงผลได้หลากหลาย
- คลาส NumberFormat ช่วยจัดรูปแบบการแสดงผลได้