

## 第三章

### 變數與資料型態

- 3.1 認識變數與常數
- 3.2 認識Java的基本資料型態
- 3.3 學習如何進行資料型態轉換
- 3.4 學習如何由鍵盤輸入資料

1

補充

### Java的資料型態

- 資料型態
  - 原始資料型態 (**primitive type**)
    - **整數(int)**與**浮點數(float)**
  - 非原始資料型態 (non-primitive type)
    - 字串(String)與陣列
    - 以特殊的方式指向變數的實體，類似C/C++中的pointer，但是在**Java中稱為reference**，此類變數在宣告時不會配置記憶體空間，而是給一個reference。

2

## 3.1 變數的使用

## 變數的使用

- 下面是變數使用的範例：

```

01 // app3_1,簡單的實例
02 public class app3_1
03 {
04     public static void main(String args[])
05     {
06         int num=6;           // 宣告num為整數，並設值為6
07         char ch='C'; // 宣告ch為字元變數，並設值為'C'
08         System.out.println(num+" is an integer");
09         System.out.println(ch+" is a character");
10     }
11 }

```

變數，其值可修改

常數

num = 3

常數

變數，其值可修改

ch = 'C'

```

/* app3_1 OUTPUT---
6 is an integer
C is a character
-----*/

```

3

## 3.1 變數的使用

## 變數宣告成final的格式

- 若是變數值不會變動，則可將該變數宣告成final：

**final** 資料型態 變數名稱 = 常數值;

利用final宣告的變數，其值不能再被更改，如

**final double PI=3.1415926;**

4

## 3.2 基本資料型態

## 基本資料型態

- 各種基本資料型態所佔的記憶體空間及範圍：

資料型態	位元組	表示範圍
long (長整數)	8	-9223372036854775808~9223372036854775807
int (整數)	4	-2147483648~2147483647
short (短整數)	2	-32768~32767
byte (位元)	1	-128~127
char (字元)	2	0~65535
boolean (布林)	1	布林值只能使用true或false
float (浮點數)	4	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
double (倍精數)	8	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$

5

原則：首位1表示負數，0表示正數。

題目1：求+4,-4的二進制表達

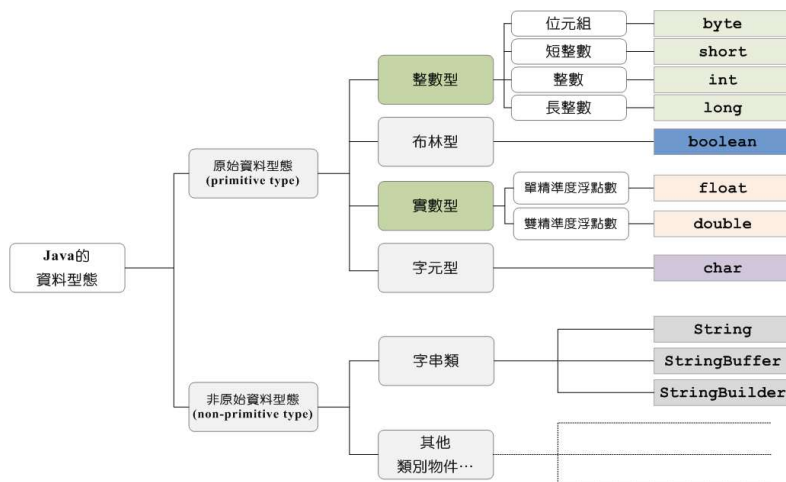
$$\begin{array}{r}
 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\
 4 \text{---} 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 = +4 \\
 -4 \text{---} 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 = -128 + 124 = -4
 \end{array}$$

題目2：求+127,-127的二進制表達

$$\begin{array}{r}
 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\
 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 = -2^7 + 1 = -127 \\
 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 = 127
 \end{array}$$

0000 0  
 0001 1  
 0010 2  
 0011 3  
 0100 4  
 0101 5  
 0110 6  
 0111 7  
 1000 -8  
 1001 -8+1=-7  
 1010 -8+2=-6  
 1011 -8+3=-5  
 1100 -8+4=-4  
 1101 -8+5=-3  
 1110 -8+6=-2  
 1111 -8+7=-1

## Java的所有資料型態



7

### 3.2 基本資料型態

## 整數型態 int

- 整數型態可分為
  - 長整數 (long int)
  - 整數 (int)
  - 短整數 (short int)
  - 位元(byte)
- 下面為短整數型態宣告的範例：

```
short sum;           // 宣告sum為短整數
```

8

## 常數的資料型態

- **Java**把整數常數的型態視為**int**，超過範圍時會發生錯誤

```

01 // app3_2, 整數常數的使用--錯誤的範例
02 public class app3_2
03 {
04     public static void main(String args[])
05     {
06         long num=32967359818;
07         System.out.println("num= "+num);
08     }
09 }

```

敘述改成**long num=32967359818L;**  
即可編譯與執行

編譯上面的程式碼，將會得到下列的錯誤訊息：

```

C:\java\app3_2.java:6: integer number too large: 32967359818
    long num=32967359818;

```

9

## 3.1 課堂練習

- 試指出下列各常數之類型。

(a) 64.156

(b) 512

10

## 字元型態 (1/2)

- 字元型態佔 2 個位元組，用來儲存字元
- Java使用的編碼系統為Unicode(標準萬國碼)
- 宣告字元變數，並設值給它：
 

```
char ch;           // 宣告字元變數ch
ch='A';           // 將字元常數'A'設值給字元變數ch
```

在宣告的同時便設定初值

```
char ch1='A';      // 宣告字元變數ch1，並將字元常數'A'設值給它
char ch2=97;       // 將ch2設值為十進制97所對應的ASCII碼 (查表D4)
char ch3='7';       // 將ch3設值為字元常數'7'
```

11

## 字元型態 (1/2)

- 下面的程式以不同的格式列印字元變數：

```
01 // app3_6,字元型態的列印
02 public class app3_6
03 {
04     public static void main(String args[])
05     {
06         char ch1=71; // 設定字元變數ch1等於編碼為71的字元(查表：D3)
07         char ch2='G'; // 設定字元變數ch2等於'G' (查表：D3)
08         char ch3='\u0047'; // 以16進位值設定字元變數ch3 (查表：D3)
09
10         System.out.println("ch1="+ch1);
11         System.out.println("ch2="+ch2);
12         System.out.println("ch3="+ch3);
13     }
14 }
```

```
/* app3_6 OUTPUT---
ch1=G
ch2=G
ch3=G
-----*/
```

12

C-4 Java 7 教學手冊 (第五版)

附錄 D ASCII 碼表

十進位	二進位	八進位	十六進位	ASCII	按鍵
0	0000000	00	00	NUL	Ctrl+I
1	0000001	01	01	SOH	Ctrl+A
2	0000010	02	02	STX	Ctrl+B
3	0000011	03	03	ETX	Ctrl+C
4	0000100	04	04	EOT	Ctrl+D
5	0000101	05	05	ENQ	Ctrl+E
6	0000110	06	06	ACK	Ctrl+F
7	0000111	07	07	BEL	Ctrl+G
8	0001000	10	08	BS	Ctrl+H, Backspace
9	0001001	11	09	HT	Ctrl+I, Tab
10	0001010	12	0A	LF	Ctrl+J, Line Feed
11	0001011	13	0B	VT	Ctrl+K
12	0001100	14	0C	FF	Ctrl+L
13	0001101	15	0D	CR	Ctrl+M, Return
14	0001110	16	0E	SO	Ctrl+N
15	0001111	17	0F	SI	Ctrl+O
16	0010000	20	10	DLE	Ctrl+P
17	0010001	21	11	DC1	Ctrl+Q
18	0010010	22	12	DC2	Ctrl+R
19	0010011	23	13	DC3	Ctrl+S
20	0010100	24	14	DC4	Ctrl+T
21	0010101	25	15	NAK	Ctrl+U
22	0010110	26	16	SYN	Ctrl+V
23	0010111	27	17	ETB	Ctrl+W
24	0011000	30	18	CAN	Ctrl+X

D-2 Java 7 教學手冊 (第五版)

ASCII 碼表 D-3

十進位	二進位	八進位	十六進位	ASCII	按鍵
25	0011001	31	19	EM	Ctrl+Y
26	0011010	32	1A	SUB	Ctrl+Z
27	0011011	33	1B	ESC	Esc, Escape
28	0011100	34	1C	FS	Ctrl+\
29	0011101	35	1D	GS	Ctrl+]
30	0011110	36	1E	RS	Ctrl+=
31	0011111	37	1F	US	Ctrl++
32	0100000	40	20	SP	Spacebar
33	0100001	41	21	!	!
34	0100010	42	22	"	"
35	0100011	43	23	#	#
36	0100100	44	24	\$	\$
37	0100101	45	25	%	%
38	0100110	46	26	&	&
39	0100111	47	27	'	'
40	0101000	50	28	(	(
41	0101001	51	29	)	)
42	0101010	52	2A	*	*
43	0101011	53	2B	+	+
44	0101100	54	2C	,	,
45	0101101	55	2D	-	-
46	0101110	56	2E	.	.
47	0101111	57	2F	/	/
48	0110000	60	30	0	0
49	0110001	61	31	1	1
50	0110010	62	32	2	2
51	0110011	63	33	3	3
52	0110100	64	34	4	4

十進位	二進位	八進位	十六進位	ASCII	按鍵
53	0110101	65	35	5	5
54	0110110	66	36	6	6
55	0110111	67	37	7	7
56	0111000	70	38	8	8
57	0111001	71	39	9	9
58	0111010	72	3A	:	:
59	0111011	73	3B	;	;
60	0111100	74	3C	<	<
61	0111101	75	3D	=	=
62	0111110	76	3E	>	>
63	0111111	77	3F	?	?
64	1000000	100	40	@	@
65	1000001	101	41	A	A
66	1000010	102	42	B	B
67	1000011	103	43	C	C
68	1000100	104	44	D	D
69	1000101	105	45	E	E
70	1000110	106	46	F	F
71	1000111	107	47	G	G
72	1001000	110	48	H	H
73	1001001	111	49	I	I
74	1001010	112	4A	J	J
75	1001011	113	4B	K	K
76	1001100	114	4C	L	L
77	1001101	115	4D	M	M
78	1001110	116	4E	N	N
79	1001111	117	4F	O	O
80	1010000	120	50	P	P

十進位	二進位	八進位	十六進位	ASCII	按鍵
81	1010001	121	51	Q	Q
82	1010010	122	52	R	R
83	1010011	123	53	S	S
84	1010100	124	54	T	T
85	1010101	125	55	U	U
86	1010110	126	56	V	V
87	1010111	127	57	W	W
88	1011000	130	58	X	X
89	1011001	131	59	Y	Y
90	1011010	132	5A	Z	Z
91	1011011	133	5B	[	[
92	1011100	134	5C	\	\
93	1011101	135	5D	]	]
94	1011110	136	5E	^	^
95	1011111	137	5F	_	_
96	1100000	140	60	`	`
97	1100001	141	61	a	a
98	1100010	142	62	b	b
99	1100011	143	63	c	c
100	1100100	144	64	d	d
101	1100101	145	65	e	e
102	1100110	146	66	f	f
103	1100111	147	67	g	g
104	1101000	150	68	h	h
105	1101001	151	69	i	i
106	1101010	152	6A	j	j
107	1101011	153	6B	k	k
108	1101100	154	6C	l	l

十進位	二進位	八進位	十六進位	ASCII	按鍵
109	1101101	155	6D	m	m
110	1101110	156	6E	n	n
111	1101111	157	6F	o	o
112	1110000	160	70	p	p
113	1110001	161	71	q	q
114	1110010	162	72	r	r
115	1110011	163	73	s	s
116	1110100	164	74	t	t
117	1110101	165	75	u	u
118	1110110	166	76	v	v
119	1110111	167	77	w	w
120	1111000	170	78	x	x
121	1111001	171	79	y	y
122	1111010	172	7A	z	z
123	1111011	173	7B	{	{
124	1111100	174	7C		
125	1111101	175	7D	}	}
126	1111110	176	7E	~	~
127	1111111	177	7F	Del	Del + Rubout

## 3.2 基本資料型態

## 跳脫字元 (1/2)

- 反斜線「\」稱為**跳脫字元**
- 反斜線「\」加上控制碼，稱為**跳脫序列**

跳脫字元	所代表的意義	跳脫字元	所代表的意義
\f	換頁 (Form feed)	\\	反斜線 (Backslash)
\b	倒退一格 (Backspace)	\'	單引號 (Single quote)
\n	換行 (New line)	\"	雙引號 (Double quote)
\r	歸位 (Carriage return)	\uxxxx	十六進位的unicode字元
\t	跳格 (Tab)	\ddd	八進位Unicode字元，範圍在八進位的000~377之間



## 跳脫字元 (2/2)

- 利用跳脫序列列印字串：

```

01 // app3_7,列印跳脫字元
02 public class app3_7
03 {
04     public static void main(String args[])
05     {
06         char ch1="\\";           // 將ch1設值為\"
07         char ch2='\74';          // 以八進位值設定字元變數ch2 (查表：D3)
08         char ch3='\u003e';       // 以16進位值設定字元變數ch3 (查表：D3)
09
10         System.out.println(ch1+"Time flies."+ch1);
11         System.out.println("\Time is money!\");
12         System.out.println(ch2+"Tomorrow never comes"+ch3);
13     }
14 }

```

可改成 `char ch2=074;`

可改成 `char ch3=0x3e;`

```

/* app3_7 OUTPUT-----
Time flies.
Time is money!
<Tomorrow never comes>
-----*/

```

17

## 浮點數與倍精度浮點數型態 (1/3)

- 浮點數 (float) 長度為4個位元組  
有效範圍為  $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
- 倍精度 (double) 浮點數的長度為8個位元組  
有效範圍為  $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
- float與double型態的變數宣告範例如下：

```

double num;           // 宣告num為倍精度浮點數變數
float sum=2.0f;       // 宣告sum為浮點數變數，並設其初值為2.0

```

18

## 3.2 基本資料型態

## 浮點數與倍精度浮點數型態 (2/3)

## • float與double宣告與設值時注意事項

- `double num1=-5.6e64;`     // 宣告num1為double，其值為 $-5.6 \times 10^{64}$
- `double num2=-6.32E16;`   // e 也可以用大寫的 E 來取代
- `float num3=2.478f;`        // 宣告 num3 為 float，並設初值為 2.478
- `float num4=2.63e64;`        // 錯誤， $2.63 \times 10^{64}$  超過float可表示的範圍

## 浮點數使用的範例

```

01 // app3_8,浮點數的使用
02 public class app3_8
03 {
04     public static void main(String args[])
05     {
06         float num=5.0f;
07         System.out.println(num+"*"+num+"="+num*num); // 印出num*num的結果
08     }
09 }

```

```

/* app3_8 OUTPUT---
5.0*5.0=25.0
-----*/

```

19

## 3.2 基本資料型態

## 浮點數與倍精度浮點數型態 (3/3)

## • 浮點數型態的最大值與最小值的代碼

	float	double
所屬類別	<code>java.lang.Float</code>	<code>java.lang.Double</code>
最大值	<code>MAX_VALUE</code>	<code>MAX_VALUE</code>
最大值常數	<code>3.4028235E38</code>	<code>1.7976931348623157E308</code>
最小值	<code>MIN_VALUE</code>	<code>MIN_VALUE</code>
最小值常數	<code>1.4E-45</code>	<code>4.9E-324</code>

下面的範例印出 float 與 double 的最大與最小值：

```

01 // app3_9,印出Java定義的浮點數常數值
02 public class app3_9
03 {
04     public static void main(String args[])
05     {
06         System.out.println("f_max="+Float.MAX_VALUE);
07         System.out.println("f_min="+Float.MIN_VALUE);
08         System.out.println("d_max="+Double.MAX_VALUE);
09         System.out.println("d_min="+Double.MIN_VALUE);
10     }
11 }

```

```

/* app3_9 OUTPUT-----
f_max=3.4028235E38
f_min=1.4E-45
d_max=1.7976931348623157E308
d_min=4.9E-324
-----*/

```

20

## 基本資料型態的預設值

- 表3.2.5 基本資料型態的預設值

資料型態	預設值
byte	(byte) 0
short	(short) 0
int	0
long	0L
float	0.0f
double	0.0d
char	\u0000
boolean	false

21

### 3.2 基本資料型態

## 布林型態

- 宣告布林變數的範例：

```
boolean status=true;    // 宣告布林變數status，並設值為true
```

在程式中印出布林值：

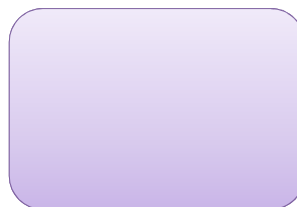
```
01 // app3_10,印出布林值
02 public class app3_10
03 {
04     public static void main(String args[])
05     {
06         boolean status=false; // 設定status布林變數的值為false
07         System.out.println("status="+status);
08     }
09 }
```

```
/* app3_10 OUTPUT---
status=false
-----*/
```

22

### 3.2 課堂練習(1/2)

- 下述變數是屬於何種資料型態？
  - (a) 886
  - (b) 12366L
  - (c) 53.472
  - (d) 383.34F
  - (e) 367.87D



23

### 3.2 課堂練習(2/2)

- 下述資料用什麼型態的變數來描述比較適合？
  1. 一顆蘋果的重量。
  2. 出生滿月的小嬰兒之的身高與體重。
  3. 這本Java書的總頁數。
  4. 某個縣市所有國小學生人數的總數。
  5. 今天有太陽。
  6. 亞洲人口的總數。
  7. 台北101的樓層數。
  8. 火星到地球的距離。

24

## 自動型態的轉換 (1/2)

**型態轉換**發生在運算子左右兩邊的運算元型態不同時  
Java會在下列條件皆成立時，自動做資料型態的轉換：

- (1) 轉換前的資料型態與轉換後的型態相容
  - (2) 轉換後的資料型態之表示範圍比轉換前的型態大
- 例如：**int 和 float 相加，int 會被轉成 float**  
**char 和 int 相加，char 會被轉成 int**

自動資料型態的轉換只限該行敘述  
透過自動型態的轉換，可以保證資料的精確度  
這種轉換也稱為**擴大轉換 (augmented conversion)**

25

## 自動型態的轉換 (2/2)

- 浮點數與整數作運算的結果：

```
01 // app3_11,型態自動轉換
02 public class app3_11
03 {
04     public static void main(String args[])
05     {
06         int a=45;           // 宣告a為整數
07         float b=2.3;        // 宣告b為浮點數
08
09         System.out.println("a="+a+",b="+b);    // 印出a、b的值
10         System.out.println("a/b="+a/b);        // 印出a/b的值
11     }
12 
```

Exception in thread "main" java.lang.Error:  
Unresolved compilation problem:  
Type mismatch: cannot convert from double to float

26

## 自動型態的轉換 (2/2)

- 浮點數與整數作運算的結果：

```

01 // app3_11,型態自動轉換
02 public class app3_11
03 {
04     public static void main(String args[])
05     {
06         int a=45;                // 宣告a為整數
07         float b=2.3f;            // 宣告b為浮點數
08
09         System.out.println("a="+a+",b="+b);    // 印出a、b的值
10         System.out.println("a/b="+a/b);        // 印出a/b的值
11     }
12 }

```

```

/* app3_11 OUTPUT---
a=45,b=2.3
a/b=19.565218
-----*/

```

27

## 自動型態的轉換 (2/2)

- 浮點數與整數作運算的結果：

```

01 // app3_11,型態自動轉換
02 public class app3_11
03 {
04     public static void main(String args[])
05     {
06         int a=45;                // 宣告a為整數
07         double b=2.3;            // 宣告b為浮點數
08
09         System.out.println("a="+a+",b="+b);    // 印出a、b的值
10         System.out.println("a/b="+a/b);        // 印出a/b的值
11     }
12 }

```

```

/* app3_11 OUTPUT---
a=45,b=2.3
a/b=19.565217391304348
-----*/

```

28

## 強制型態轉換 (1/3)

- 將資料型態強制轉換成另一種型態的語法：

資料型態的強制性轉換

(欲轉換的資料型態) 變數名稱;

強制型態轉換也稱為顯性轉換 (explicit cast)

29

## 強制型態轉換 (2/3)

- 整數與浮點數進行強制轉換的範例：

```

01 // app3_12,強制轉換
02 public class app3_12
03 {
04     public static void main(String args[])
05     {
06         int a=36;
07         int b=7;
08
09         System.out.println("a="+a+",b="+b);    // 印出a、b的值
10         System.out.println("a/b="+a/b);        // 印出a/b的值
11         System.out.println("(float)a/b="+((float)a/b));
12     }
13 }

```

也可寫成

將a轉換成浮點數後，再除以b

`a / (float)b`      `(float)a / (float)b`

```

/* app3_12 OUTPUT---
a=36,b=7
a/b=5
(float)a/b=5.142857
-----*/

```

30

## 強制型態轉換 (3/3)

- 強制型態轉換的注意事項
  - 變數強制轉換成另一種型態，原先的型態不會被改變
  - 縮小轉換（**narrowing conversion**）可能會漏失資料的精確度
  - **Java不會主動做縮小轉換**

31

## 3.3 課堂練習

- ex3\_3\_1.java
  - 如果想把 $128*64$ 的運算結果轉換成浮點數，應該如何撰寫？

```
/* output----  
128*64=8192.0  
-----*/
```

32



## 輸入資料的基本架構 (1/3)

- 資料輸入的格式：

輸入資料的基本格式

```
import java.util.Scanner;
public class class name      // 類別名稱
{
    public static void main(String args[])
    {
        Scanner 物件名稱=new Scanner(System.in); // 宣告 Scanner 類別的變數
        資料型態 變數名稱 a; // 宣告變數
        ... ..
        變數名稱=物件名稱.相對應型態的 Scanner 類別函數; // 讀入資料至變數
        ... ..
    }
}
```

33

## 輸入資料的基本架構 (2/3)

- 輸入資料時Scanner類別提供的函數：

資料型態	相對應的函數	資料型態	相對應的函數
byte	nextByte()	float	nextFloat()
short	nextShort()	double	nextDouble()
int	nextInt()	String	next()
long	nextLong()	String	nextLine()

使用Scanner類別輸入資料的範例：

```
Scanner scn=new Scanner(System.in); // 宣告 Scanner 類別的物件
int i; // 宣告整數變數
double d; // 宣告倍精度浮點數
i=scn.nextInt(); // 輸入整數數值，給 i 存放
d=scn.nextDouble(); // 輸入倍精度浮點數數值，給 d 存放34
```

## next()與nextLine()的差別

- **next()** 是可以輸入字串，但是只要一按空白鍵 就會中斷後面的輸入
  - 例:你輸入了"ABC CBA AAA"列印出來的會是"ABC"
- **nextLine()** 也是字串輸入，但即使按了空白鍵以後輸入也不會中斷
  - 例:輸入了"ABC CBA AAA"列印出來會是"ABC CBA AAA"

**next()**是以空白或換行做為區隔  
**nextLine()**則是以換行做為區隔讀整行

35

### 3.4 由鍵盤輸入資料

## 輸入資料的基本架構 (3/3)

- 由鍵盤輸入字串的範例

```

01 // app3_13, 由鍵盤輸入資料
02 import java.util.Scanner;           // 載入 Scanner 類別
03 public class app3_13
04 {
05     public static void main(String args[])
06     {
07         Scanner scn=new Scanner(System.in); // 宣告 Scanner 類別的物件
08         String name;
09         int age;
10
11         System.out.print("What's your name? ");
12         name=scn.nextLine();           // 輸入字串
13         System.out.print("How old are you? ");
14         age=scn.nextInt();             // 輸入整數
15
16         System.out.print("Hi, "+name+", you're ");
17         System.out.println(age+" years old.");
18     }
19 }

```

/\* app3\_13 OUTPUT-----  
What's your name? Junie Hong  
How old are you? 6  
Hi, Junie Hong, you're 6 years old. 36  
-----\*/

## 3.4 課堂練習

- ex3\_4\_1.java
- 請撰寫一程式，由鍵盤輸入 "I am a good student!!" 字串，輸出結果也是 "I am a good student!!" 字串。

```
/* output-----
請輸入資料:I am a good student
I am a good student
-----*/
```

37

## 3.4 課堂練習

- ex3\_4\_2.java
- 請撰寫一程式，由鍵盤讀入兩個數值1200與2100，並計算二數之和。

```
/* output-----
Input first integer:1200
Input second integer:2100
1200+2100=3300
-----*/
```

38

## 輸入數值--不合型態的輸入

- 若是需要輸入數值，卻輸入字元 'k'，則會出現類似下列的錯誤訊息：

```
What's your name? Junie Hong
How old are you? k
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:909)
    at java.util.Scanner.next(Scanner.java:1530)
    at java.util.Scanner.nextInt(Scanner.java:2160)
    at java.util.Scanner.nextInt(Scanner.java:2119)
    at app3_13.main(app3_13.java:14)
```

39

## 以Scanner類別輸入字元

用next() 取得字串後，再利用charAt(0) 函數取出字串中第0個字元即可：

```
01 Scanner scn=new Scanner(System.in);
02 String str;
03 char ch;
04 str=scn.next();           // 輸入字串
05 ch=str.charAt(0);         // 取出字串第 0 個字元
```

40

## 3.4 課堂練習

- ex3\_4\_3.java
- 請將鍵盤中輸入的字元，以前後3個\*包圍，假設輸入的字元是**SWEET**，接著取出**SWEET**字串中的第二個位置的字元**W**，使輸出結果為**\*\*\*W\*\*\***

```
/* output-----
請輸入:SWEET
***W***
-----*/
```

41

## Ch03\_回家作業

- hw3\_1.java
- 請撰寫一個程式, 在螢幕上顯示如下的訊息：

```
/* output-----
\這是第 3 章的習題\
-----*/
```

42

## Ch03\_回家作業

- hw3\_2.java

請先利用筆或計算機計算右列公式，求出x 之值，然後再撰寫程式，求出右列數學公式x 之值（請宣告適當的資料型態）。若發現兩者有所不同，則請修正運算式或資料型態。

```
/* output-----
   x=0.7868852459016394
   -----*/
```

a=2 b=6 c=4 d=5 e=6

$$x = \frac{b^2 - 4ac}{d + \frac{1}{2e}}$$

43

## Ch03\_回家作業

- hw3\_3.java

- 請撰寫一程式，由鍵盤輸入一個整數num，其輸出結果為這個整數的-(num)。

```
/* output-----
   Input an integer:6
   num=6
   -(num)=-6
   -----*/
```

44

## Ch03\_回家作業

- hw3\_4.java
  - 試由**鍵盤讀入**兩個整數 "3721" 與 "1572"，再計算此二數之差。

```
/* output-----
Input first integer:3721
Input second integer:1572
3721-1572=2149
-----*/
```

45

## Ch03\_回家作業

- 題目 1: 請說明為什麼byte(位元)的整數表達範圍為: -128~127
- 題目 2: 求+125,-48的二進制表達。

46