Computer Programming Lab 2

2017

유연일(Yeonil Yoo) & 정원일(Wonil Jeong)

Contact

Email: cp2017ta@tcs.snu.ac.kr

Kakao open chat: https://open.kakao.com/o/geg9SaA

(you may ask in Korean or English)

You may use your own laptop, personal choice of IDE

Programming midterm and final will be open note (no internet)

Java™ Platform, Standard Edition 8 API Specification

https://docs.oracle.com/javase/8/docs/api/

Char

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

String

public final class **String**
extends Object
implements Serializable, Comparable<String>, CharSequence

The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

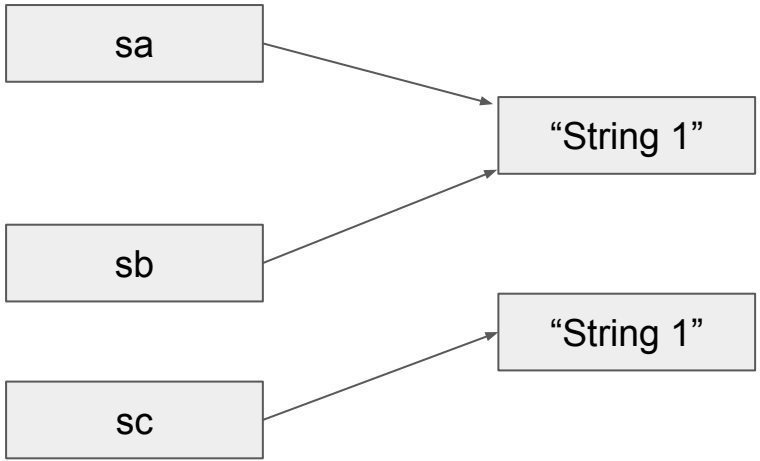
```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};
 String str = new String(data);
```

(https://docs.oracle.com/javase/8/docs/api/java/lang/String.html)

```java
public static void primitive_data() {
    int a = 10;
    int b;
    b = a;
    System.out.println(a + " " + b); //10 10
    char ca = 'a'; //2byte = 16bit = 2^16, 0 ~ 2^16-1
    System.out.println((int)ca); //97
    ca++;
    System.out.println(ca); //b
    System.out.println((char)98); //b
    System.out.println((char)0x62); //b
    String sa = "String 1";
    String sb = "String 1";
    System.out.println((sb == sa)); //true
    String sc = new String("String 1");
    System.out.println((sc == sa)); //false
}
```


sa
sb
sc
“String 1”
“String 1”

```java
public static void loop() {
    for(int i = 0; i < 10; i++) {
        System.out.print('*');
    }
    System.out.println();

    int i = 0;
    for(; i < 10;) {
        System.out.print('*');
        i++;
    }
}
```

Scanner class

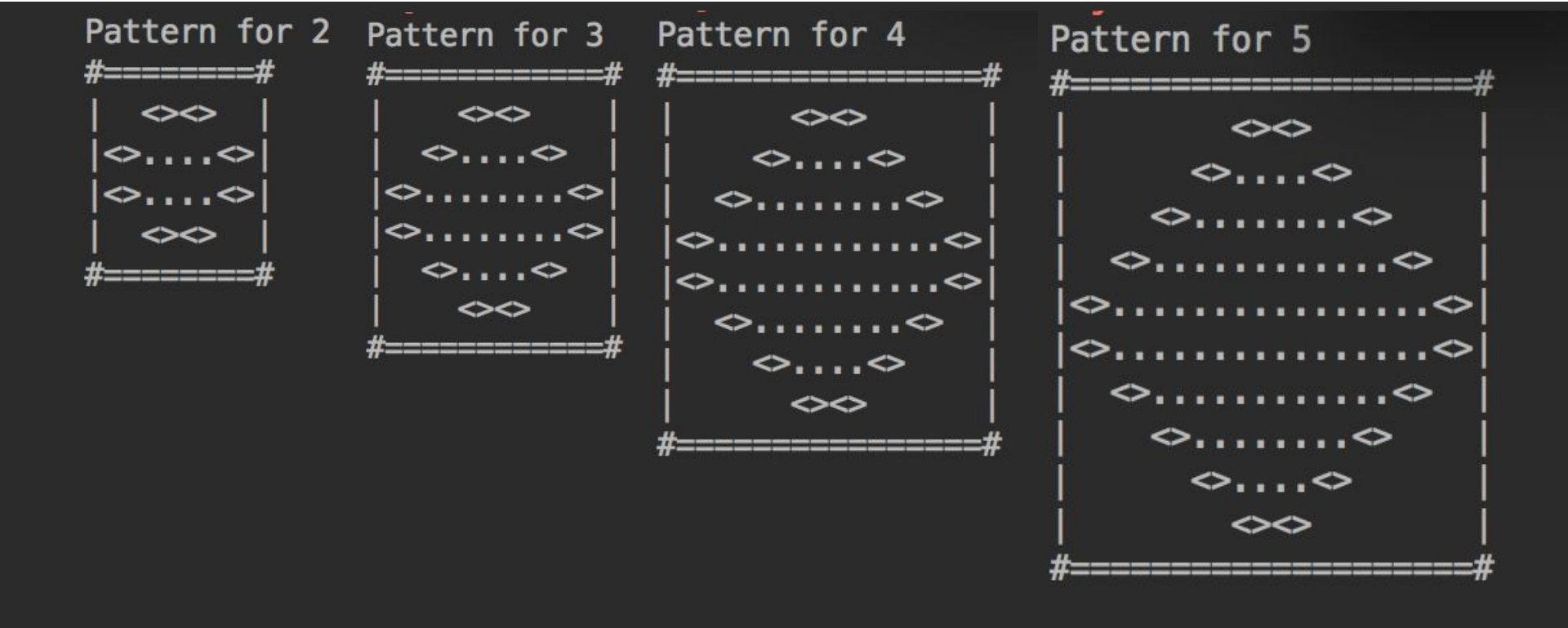
https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html

- Scanner generation
  - import java.util.Scanner
  - Scanner myScanner = new Scanner(inputStream(System.in));
- Scanner usage
  - String myString = myScanner.next();
  - String myString = myScanner.nextLine();
  - int myint = myScanner.nextInt();
  - double mydouble = myScanner.nextDouble();

```java
    public static int scan_example() {
        Scanner scan = new Scanner(System.in);
        System.out.print("Type int: ");
        //int read = scan.nextInt();
        int read = Integer.parseInt(scan.nextLine());
        System.out.println(read);
//*
        //scan.nextLine();
        System.out.print("Type some word: ");
        String line = scan.nextLine();
        System.out.println(line);
//*/
        return read;
    }
```

To do

Write a method that will print pattern depending on variable and return how many times loops have ran



```
public static void main(String[] args) {
   System.out.println("loop count: " + to_do(scan_example()));
}

public static int to_do(int size) {
   int count = 0;
   return count;
}
```

Prime, perfect, Fibonacci prime, Mersenne prime

Perfect Number: A positive number which is the sum of its proper positive divisors.
ex) 28 = 1+2+4+7+14

Fibonacci Prime: A prime number which is also fibonacci number.
ex) 13 is a prime and fibonacci number [1, 2, 3, 5, 8, 13]

Mersenne prime: A prime number which have 1 less value than power of 2.
ex) 31 = 2^5 - 1

Practice 2

Problem: Given a number “N”, print how many perfect num, fibonacci prime, mersenne prime in [1 … N] where format is like [4, 3, 2] and then print what numbers are perfect num, fibonacci prime, and mersenne prime line by line.

[Input] : int N
[output] :
    [pe, fi, me]
    [pe1, pe2 … pe_y]
    [fi1, fi2, … fi_z]
    [me1, me2, … me_t]