

Dasar-Dasar Pemrograman 2

Lab 02 Methods



FAKULTAS
ILMU
KOMPUTER

I. Methods

Mengapa method digunakan ?

Coba perhatikan contoh code di bawah

```
int sum = 0;
for (int i= 1; i<= 10; i++){
    sum += i;
}
System.out.println("Sum from 1 to 10 is " + sum);

sum = 0;
for (int i= 20; i<= 30; i++){
    sum += i;
}
System.out.println("Sum from 20 to 30 is " + sum);

sum = 0;
for (int i= 35; i<= 45; i++){
    sum += i;
}
System.out.println("Sum from 35 to 45 is " + sum);
```

Pada code di atas terdapat prosedur atau algoritma yang dilakukan secara berulang-ulang. Prosedur yang berulang tersebut dapat kita buat menjadi sebuah block method yang nantinya dapat kita panggil berkali-kali. Contoh method yang dapat kita buat dari code di atas adalah

```
public static int sum(int x1, int x2) {
    int sum = 0;
```

```

for (int i= x1; i<= x2; i++)
    sum += i;
return sum;
}

```

Sehingga pada block main kita hanya perlu memanggil method tersebut untuk menjumlahkan bilangan dari x1 hingga x2.

```

public static void main(String[] args) {
    System.out.println("Sum from 1 to 10 is " + sum(1, 10));
    System.out.println("Sum from 20 to 30 is " + sum(20, 30));
    System.out.println("Sum from 35 to 45 is " + sum(35, 45));
}

```

Pada DDP 1 kalian telah mengenal konsep **functions and methods**-sebuah mini program (*block of code*) yang ketika dipanggil akan menjalankan perintah-perintah yang didefinisikan di dalamnya. Berikut adalah format **umum** pendefinisian suatu method di Java.

```

accessModifier returnType methodName(parameterType1 parameterName1,
...) {
    // statements
}

```

- **accessModifier** menentukan siapa-siapa saja yang dapat mengakses method ini. Untuk sekarang, kita akan menggunakan *modifier* public yang berarti semua instance/class dapat mengaksesnya.
- **returnValueType** menentukan tipe data yang di-*return* oleh function ini. Jika function-nya tidak me-*return* apa-apa, gunakan *keyword* void.
- **methodName** dan **parameters** menjelaskan bagaimana function ini dipanggil dan apa saja parameter yang dibutuhkan.

Contoh paling umum yang paling sering kita temukan:

```

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

```

Selain contoh di atas, berikut contoh lainnya.

```

public static int multiply(int a, int b) {
    return a * b;
}

```

```
int a = 2;
int b = 3;
int c = multiply(a, b);

// c = 6
```

Note: **static** adalah *keyword* yang menandakan bahwa method ini tidak dimiliki oleh sebuah instance class tertentu sehingga method ini bisa dipanggil oleh siapa saja.

Method Overloading

Java mengenal konsep method overloading. Overloading merupakan konsep di mana terdapat 2 atau lebih method dengan **nama yang sama**, tetapi **parameter-parameternya (parameter listnya) berbeda**: entah itu banyaknya, tipenya, atau urutan parameternya. Java compiler akan menentukan method mana yang dipanggil berdasarkan method signature-nya. Method signature adalah gabungan antara nama method dan parameter list.

```
public static int multiply(int a, int b) {
    return a * b;
}

public static double multiply(double a, double b) {
    return a * b;
}
```

II. String methods

Objek bertipe data String memiliki beberapa *method* yang bisa kamu manfaatkan untuk mengolahnya. *Method* yang dapat berguna dalam pengerjaan soal lab pekan ini di antaranya adalah `.substring()` dan `.charAt()`.

1. substring()

Method `substring()` memiliki dua versi, yakni `substring(firstIndex)` dan `substring(firstIndex, lastIndex)`. *Method* `substring(firstIndex)` mengembalikan substring dari indeks `firstIndex` (inklusif) sampai akhir. Sementara itu, *method* `substring(firstIndex, lastIndex)` mengembalikan substring dari indeks `firstIndex` sampai `lastIndex-1`.

Contoh:

```
String message = "Hello, world!";
String substr1 = message.substring(3); // "lo, world!"
String substr2 = message.substring(2, 9); // "Llo, wo"
```

2. charAt()

Method charAt(i) mengembalikan karakter (bertipe data char) pada index ke-i dalam string.

```
String message = "Hello, world!";  
char c = message.charAt(1); // 'e'
```

Setiap character memiliki nilai ASCII-nya masing-masing, sebagai contoh 'a' memiliki nilai ASCII 97 dalam desimal. Contoh cara mendapatkan nilai ASCII dari suatu character

```
char someChar = 'a';  
int x = (int) someChar; //97
```

Untuk mengembalikan ASCII menjadi char dapat menggunakan cara sebagai berikut

```
int toChar = 90;  
char x = (char) toChar; //Z
```

Untuk nilai ASCII character lainnya dapat dilihat pada tabel di bawah ini.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

© w3resource.com

Selengkapnya mengenai string methods, silakan buka [dokumentasi Java mengenai String](#).

Soal Lab 02

Your Lay's in April



Berhubung bulan Agustus Lay's tidak lagi diproduksi, Sasha, Nicolo, Jean dan Connie sepakat untuk menimbun Lay's untuk konsumsi pribadi. Mengingat bulan April adalah bulan Puasa, mereka berencana untuk memborong Lay's sebanyak mungkin tepat sebelum puasa dimulai, untuk mengantisipasi orang-orang yang membeli Lay's untuk dijadikan parcel lebaran.

Mereka berhasil membuat daftar toko-toko yang menjual Lay's, namun perbincangan mereka berhasil disadap oleh Gabi, yang berencana juga untuk memborong, tapi untuk dijual kembali. Gabi berhasil mendapatkan informasi mengenai toko yang mereka incar dan memborong Lay's di toko-toko tersebut.

Untuk mencegah kebocoran lagi, Jean mengusulkan untuk berkomunikasi dengan cipher khusus yang telah mereka sepakati, di mana mereka menggunakan Caesar's Cipher tapi dilakukan enkripsi dan dekripsi berkali-kali. Bantulah mereka membuat program yang akan membantu mereka untuk menerjemahkan cipher-cipher tersebut!

Program anda diwajibkan mengimplementasikan dua Method untuk enkripsi dan dekripsi

Format Input

***Anda tidak perlu handle input yang tidak sesuai format**

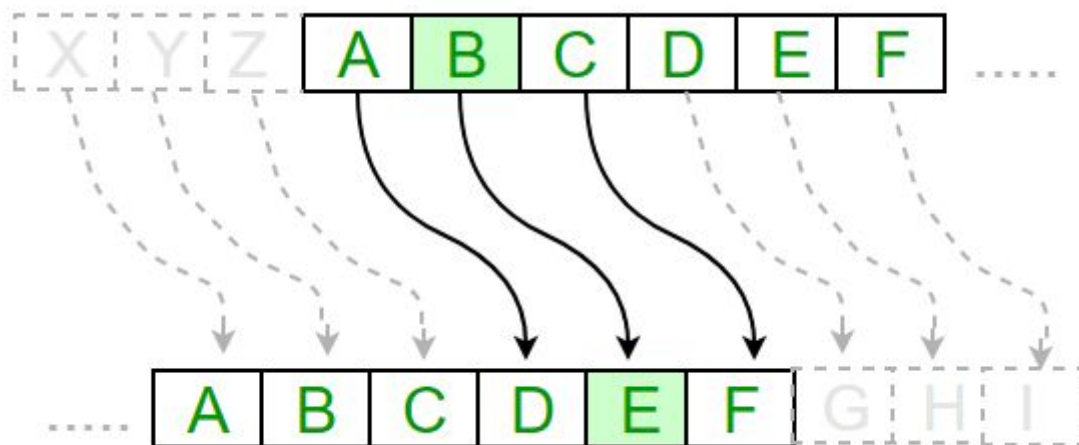
[PESAN] [URUTAN TAHAP ENKRIPSI/DEKRIPSI]

Pesan : Pesan yang akan di-decrypt atau di-encrypt (**PESAN PASTI UPPERCASE**)

Urutan Tahap Enkripsi/Dekripsi : String dengan panjang N yang berisi karakter E (enkripsi) dan/atau D (dekripsi). Dari kiri ke kanan, jalankan algoritma enkripsi/dekripsi Caesar's Cipher sebanyak N kali.

Sebagai contoh, EEED menandakan tiga kali tahap enkripsi dan diikuti satu kali tahap dekripsi.

Caesar Cipher menggeser huruf sebanyak 3 kali ke kanan sebagai berikut:



Format Output

[PESAN]

Pesan : Pesan yang sudah di-encrypt atau di-decrypt

Contoh Masukan 1

SEAWEEEDGANG EEED

Contoh Keluaran 1

YKGCKKJMGTM

Penjelasan Contoh 1

- Ketika menjalankan E pertama, pesan akan menjadi seperti berikut

VHDZHHGJDQJ

- Ketika menjalankan E kedua, pesan akan menjadi seperti berikut

YKGCKKJMGTM

- Ketika menjalankan E ketiga, pesan akan menjadi seperti berikut

BNJFNNMPJWP

- Ketika menjalankan D, pesan akan menjadi seperti berikut (kembali seperti E kedua)

YKGCKKJMGTM

Contoh Masukan 2

HXDBCJACNMCQRBBCXAH DDD

Contoh Keluaran 2

YOUSTARTEDTHISSTORY

Penjelasan Contoh 2

- Ketika menjalankan D pertama, pesan akan menjadi seperti berikut

EUAYZGXZKJZNOYYZUXE

- Ketika menjalankan D kedua, pesan akan menjadi seperti berikut

BRXVWDUWHGWKLVWRUB

- Ketika menjalankan D ketiga, pesan akan menjadi seperti berikut

YOUSTARTEDTHISSTORY

Komponen Penilaian

- 45% Kebenaran kode.
- 45% Penggunaan Method Java yang benar.
- 10% Dokumentasi dan kerapian kode.

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.

Lab02_[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap].zip

Contoh:

Lab02_A_LN_1234567890_DekDepe.zip