	015. 공용코드 (변수, 에서드)	공유하는 방법 : 상	4											
				ralization 수행 :	일반화									
						유교도를 추측하여	부수퍼클레스(Object	ctList)를 경의하다	k.					
	ObjectLis	t - count	- size : int				인 용어로 변경한다.							
		- COUNT	- elementData :	object [] ==> Al										
			- DEFAULT_CAI		T									
			- add()	Adiri - 10										
			- toArray()											
	1		- get(int)											
			- get(int) - remove(int)											
			- remove(int)											
	BoardILis	. 1	MemberList											
	Boardick	100			: :기 : 같은 기능을 수행하는 메서드									
	get() <- 오반	다이용	getByEmail() <-	의원 대이디를 찾	타일이나, 개수, 순서가 다르더라:	=다면 이용을 될게	아는 것이 결혼성 1	KUM WU.	W#07#					
	remove() <- add() <- Q	보이다이용 	get()	~ MCCOCC	다입이다. 계수, 문제가 다느더다. 역 일관성을 제공하는 문법 "Overle	도 될 건 기능을 수 없		에 될슨 이용될 수	내용으로써 -					
	add() <- ⊻	바라이닝		>> 프로그래워크	기일관경을 제공하는 문법 "Uverio 여왕으로써 프로그래밍의 일관성	oading" = 모터트	8							
				'살은 이용을 주	어로드로써 프로그래워의 중독의	을 제공하는 준입								
	*상속과 추상클래스, 추상에서	노, 콘크리트 플레스												
	-	crete Class	→	^L	oject : 객체(클래스	,3101 31H:#4:**	으로 가져야 할 필요		MOIL					
	Coi	CIELE CIASS)다면 기준(별우)력	ニ노 가서야 할 필드	_과 네지느가 열0	I KU.					
				↑ (t	상속)									
		crete Class	1				면 기본적으로 가져							
	Col	C. CLE CIASS	→						느가 있다. 난 캠의 (추상 메서드)					
	014818	만들 수 있는 클래스		1 (4	상속) : 메서드 일부	는 서브들데스가 시	rC의 작활에 벚빛 ·	구는아노막 병식년	1 명의 (주양 배제도)					
-	원스턴스를	그를 구 있는 할데스					= 격계가 가져야할 :		אורנ					
					. MAHIO 01		- 역세가 가져아받 ! P현하고, 일부는 서!							
				1 (4	상속)	구 무성에지르는 기	r면이고, 말주는 제:	프롤대스에 비운다	,					
				AbetractSo	equentialList : 인덱스로 목	BO CIEL 2017	나 가져야할 필드와 I							
					LAMO		가기서아를 들으고 ! P현하고, 일부는 서!							
				1 (1	상속) : 양쪽받은 필	T TOWN_C 1		그들네그에 이윤다						
	Cor	crete Class	1 _	Arra	nyList : 배열을 이용	240 BES LIEF	 필요한 필드와 메.	서도가 이다						
		crete olass		Alle	· 012/71/11 그	이라지 않으므트	추상에서드를 구현한	MI_21 MG:						
					. 47414	2014 BC IL -								
-														
	A	m10		C obj = new C()); test() {									
		m40		obj.m5();	this.m5 ==> 0	C this.m4 ==> C	this.m3 ==> C	this.m2 ==> B	this.m1 ==> A					
	1			obj.m4();	}									
	В	m2()		obi.m3():	ĺ									
		m3()		obj.m3(); obj.m2();	super.m5() <=	=== error								
	1	- 0		obj.m1();	super.m4() ==									
	С	m3()			super.m2() ==									
		m4()			super.m1() ==									
		m5()			*상위에서 찾음									
i	*String 클래스 사용법													
				JVM stack		Heap								
	String s1;													
	s1 = new St	ing("Hello");		s1		200								
				200		Hello								
				s2		300								
	String s2 = r	ew String("Hello");		200		Hello	<< String 클래스	느의 인스턴스						
				x										
	String x =	"Hello";												
	String y =	"Hello";		у										
		String 상수품(p	pool) 0 											
		String 인스턴스												
		"중복생성하지	않는다"											
				500	String constant pool									
					Hello									
					<< String 클래스의 인스턴스									
						hush								
	*String intern()				JVM Stack		Heap							
							200							
	String s1 =	new String("He	llo")					Hello						
					s1									
	String s2 =	s1.intern()				00	String pool							
		>>s1과 같은 문	자열을 가진		s2		300							
		>>String 객체	String pool Ol A	찾아서 리턴	31	00		Hello						

	new String("H	lello")		s3 300											
0	"Lielle"			300	1										
String s3=	"Hello"	10													
	30	10													
Object	- equals()	<- 인스템스 주소	가 같은지 비교> == 여시	자와 동일하다.											
	- toString()	<- "클레스의 @	:가 같은지 비교> == 연신 식별번호" 리턴												
	- hashcode()	<- 식별번호 리턴		Member m = n	ew Member():										
				m.equals():	← object										
				m.equals(); m.toString();	← object										
	← (상속)	Member		m.hashCode();	← object										
	← (상속)	String	- equals() 오버라이딩	String s1 = new	String("Hello");										
			>>문자열을 비교	String s2 = new											
				s1 == s2 ==> fa	lse										
				s1.equals(s2) =	=>true										
public class Ex	xam0110														
public static vo	id main(String[] a	rgs) {				Meti	hod Area	JVM stack	Heap						
			 _												
String s1;		>> s1은 인스턴스	스 주소를 담는 레퍼런스다.												
s1 = new String	g("Hello");	>>Heap 0 "Hell	o" 문자 코드를 저장함 메드	:디솔 만들고 그 주소를 건	I면만나. (s1, s2)					1					
String s2 = new	v String("Hello");	SE UP FLE OF	스턴스다. = 주소가 다르기						l	1					
System.out.prir	mm(S1 -= SZ);	~ 제로 나는 인:	나도가 나도기	ME CL.				Si	tring constant po	i I					
3															
J															
										1					

-									

1									
1									
l									
l									
l									
l									

1									
ı									

1									
l									
l									

1									
1									
l									
l									
l									
l									

1									
1									
1									
1									
1									
1									
1									
1									
1									
1									
1									
-									
-									
1									
1									
1									
-									
1									
1									
-									
-									
1									
1									
1									
1									
1									
1									
1									
1									
1									
1									
1									

4										