1															
	*031.lterator III E	책용 : 데이터 조	회를 객체화												
	>데이터 조회와 전	한편된 필드와 메시	너도를 분리해서 오	리부 부품(클래스)으	으로 만든다 => 객	知 學									
	>기능 변경 및 교회														
1					1										
-				ļ		ļ									
				L/											
					1										
				1 '		1									
1															
	class를 경의한다	= 부중화 한다.													
	>데이터 조회와 관	·현된 코드를 즐겁	ዘ스로 분리한다.												
							itorator() ()								
			BoardHandler	_ →	D	1	< <interface>></interface>		AbstactList						
	App	-	BoardHandler	1 - I	BoardDao	→		→	AbstactList						
						,	List		iterator() { - }						
						→	< <interface>></interface>								
							Iterator								
1						· · · · · · · · ·		•							
+				$\overline{}$											+ + + + + + + + + + + + + + + + + + + +
	*032. java.util 🖪	키지의 컬렉션 AF	기 사용												
1	*033. 파일 입출력	: binary 저장 / S	H DI												
1															
		save()													
	BoardDao	→	File: 파일	1											
	DoardDao	←	FIIE: ME												
		load()													
		0													
					$\overline{}$										
1															
	*바이너리(binary														
	>byte 행식의 데이터 :	- 바이너리													
				바이트 배열		2. 순차적으로 저	2								
GIOLEI TEX	객체(인스턴스)		바이트	바이트	HFOI≡	→	_	114	일						
네이디 시장	4A(0262)	→		DIOI =	DIOI S			щ	14						
		1. 필드에 저장된				Ť									
		바이트 배열로 변	8			FileOutputStrea	m								
				바이트 배열		1. 순차객으로 읽	기								
GIOLEI SIOI	객체(인스턴스)	←	바이트	바이트	바이트	←		II)	일						
		2. 바이트 배열의	가유			T			_						
		필드에 저장				FileInputStream									
1		==에 시장				Fileliiputotrealii									
<u> </u>															
	*FileOutputStream	im 클래스													
1						쓰기									
1			→	FileOuto	outStream	⇒	파일	i							
1	thide 3 24			. neoutpt		4 - /									
	1byte 출력		write()												
	바이트 배열 출력		write(byte[])												
			←	FileInput	itStream	←-									
	1byte 읽기		read()												
1	바이트 배열 읽기		read(byte[])												
1			(_,)												
—												_			
	*파일 입출력 예외														
	l을 사용자에게 명확	하게 안내하지 듯	한다.	update()											
단점 : 오류 상황		선택4 : 흐음자에	게 위임	insert()		- 생성									
단점 : 오류 상황	JVM			findBy()		- write()									
	JVM			ubj()	DAO	→ -	FileOuto	utStream	→	파일					
단점 : 오류 상황 main()	JVM ↓	execute()	Handler				. iic Catp		-						
	JVM	execute() →	Handler					1							
	JVM ↓ App	execute() → dela3	Handler	선택 2		- close()									
main()	JVM ↓ App	execute() → 선택3 호출자에게 일임	Handler			- 등등등	*예외	발생"							
main()	JVM ↓ App	execute() → 선택3 호출자에게 일임	Handler	선택 2		- close() - 등등등 보고	"예외	발생"							
main()	JVM ↓ App	execute() → 선택3 호출자에게 일임	Handler	선택 2		- 등등등 보고	"예외	발생"							
main()	JVM ↓ App	execute() → 선택3 호출자에게 일임	Handler	선택 2	1	- 등등등	"예외	발생"							
main()	JVM ↓ App	execute() → 선택3 호출자에게 일임	Handler	선택 2 호출자에게 일임		- 등등등 보고 (throw)	*예외	발생"							
main()	JVM ↓ App	execute()	Handler	선택 2 호출자에게 일임	선택 1 : DAO에/	- 등등등 보고 (throw)	*예외	발생"							
main()	JVM ↓ App	execute()	Handler	선택 2 호출자에게 일임		- 등등등 보고 (throw)	*예외	받생"							
main()	JVM ↓ App >현재 에서드를 1	execute()	Handler Handler 에게 예외 상황 알	선택 2 호출자에게 일임	선택 1 : DAO에/	- 등등등 보고 (throw)	"에 외	받생"							
main()	JVM ↓ App	execute()	Handler	선택 2 호출자에게 일임		- 등등등 보고 (throw)	"에 오!	발생"							
main()	JVM ↓ App >현재 에서드를 1	execute()	Handler Handler 에게 예외 상황 알	선택 2 호출자에게 일임	선택 1 : DAO에/	- 등등등 보고 (throw)	*예외	발생"							
main()	JVM ↓ App >현재 에서드를 1	execute()	Handler Handler 에게 예외 상황 알	선택 2 호출자에게 일임	선택 1 : DAO에/	- 등등등 보고 (throw)	*에 오	발생"							
main()	JVM ↓ App >현재 에서드를 1	execute()	Handler Handler 에게 예외 상황 알	선택 2 호출자에게 일임	선택 1 : DAO에/	- 등등등 보고 (throw)	"예외	발생"							

57.0.4 do								
new FileOutputStream("board.data") JVM을 실행하는 폴더에 파일을 생성한다.								
eclipse의 경우 프로젝트 플데(에 : appl) 명칭장의 경우 Sjava ~ java를 실행하는 플더 ~> board.data 파일이 만들어짐								
명령황의 경우 \$java ~								
java를 실행하는 폴더 -> board.data 파일이 만들어짐								

-									

1									
1									
l									
l									
l									
l									

1									

	l									
	-									
	-									
_	l									

1									

1									
l									

	1									
	1									
	l									
	l									
	l									
	l									
	l									
	l									

1										
1										
1										
1										
1										
1										
1										
1										
1										
1										