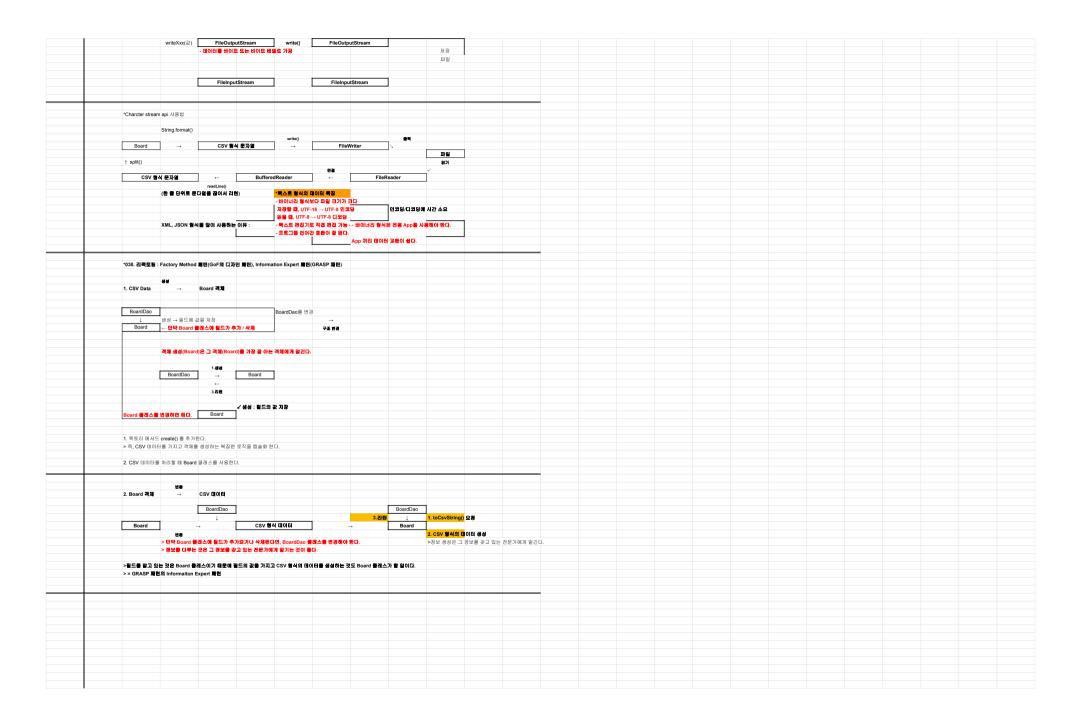
		맛은 가공해서 출력	역하는 코드를 객치	화시키기 ==> 메/	서드, 클래스 라는	분목에 강추기 = 켕슬회	# (encapsulation)									
class DataOutp																
void writeInt(int	value) {															
		1노들 메서드 안이	에다 넣는다(감춘디	1)												
encapsulatim (경험의) [기도이제 가입기	U 21 4128 01 E I	- 74													
목삼한 코드를 (서드안에 감춰서	쓰기 경계 만드는	2 .													
	는 "데코레이터"(0.00												
java.io 패키지의	스트리밍 API를	사용하여 Primitive	e type, string type	을 출력												
"추가기능" = "(
중간에서 가공 :	러리 일을 하는 객기															
*035. 자바 I/O S	treams : java.io	•														
		byte Stream (by	yte, byte[])		character stream	m stream (char, char []	1)									
		Input	tStream / Output			Reader/Writer										
Data Sink Stre		파일	FileInputStream		File	e Reader										
데이터 저장소0		M M	FileOutputStrea	ım		e Writer										
직접 읽고/쓰기		에모리	ByteArray	InputStream	char / String											
수행하는 클래스	:	예포터	ByteArray	OutputStream	char / String											
		프로세스	Piped	InputStream		d Reader										
			Piped	OutputStream	Piped	d Writer										
Data processir			m / DataOutputStr		BufferedReader	r/ BufferedWriter										
중간에서 데이터		ObjectInputStre	eam / ObjectOutpu	tStream												
가공하는 일을			tream / BufferedOu													
수행하는 클래스	:	PrintStream			PrintWriter											
"Decorator"																
		그 강 그대로 출	=		문제에 해당하는	= 코드값을 알아내서 코	a드 값을 흐음									
버耳(buffer)																
	부분을 당는 역열															
			는데는 시간이 많이	I 소요 됨.												
Data	→	buffer	1	-		파일										
		RAM	-	버퍼가 꼭 차면		Hdd										
				실제 파일로 음력	4											
	i e			1												
		1. 음력함 데이터	크를 RAM에 모아되	F었다가												
					2.RAM이 작차면	면 HDD로 클릭함으로써	H 입출력에 소요되는 시간을 줄	원다 .								
	1 byte씩 바로 B	로 HDD로 클릭하	하는 것 보다													
			있다가 한번에 좀	력하는 것이 시간이	이 열 소요된다.											
*036. 객체를 입	음력 하기															
									•객체 직결화 방식으로 입							
스탠스 ==> Board	→	DataOut	tputStream	→	FileOut	putStream	→ 파일		장점 : 입출력 코딩이 쉽다							
	필드별로 용격															
	- writeXxx()															
									단점 : 자바가 아니라면 읽	고 쓰기가 어렵다.						
									>> 다른 프로그래밍 언어	원의 쓰기가 어렵다.						
인스틴스 ==> Board	→	ObjectOu	utputStream	→	FileOut	putStream	→ INQ	1								
	언스런스를 통해로 (1								
	- writeObject()															
	40															
								$\overline{}$		$\overline{}$					$\overline{}$	
*Serialize St De	Serialize															
		en en			28		∠ 클래스 정보	이름, 패키지명 등)								
	인스턴스	1 -	ObjectOu	tputStream		바이트 배열		rialVersionUID)								
		1		화) = 바이트 배열5	\$		✓ 필드 값									
			✓ marshaling					음력하는 것보다 파	1일 크기가 더 커지는 단점(이 있다.						
							·그러나 코딩0									
					9171		✓ 문계시 제비	이름, 패키지영 등)								
	인스템스	44	Objecting	tputStream	읽기	바이트 배열		이용, 패키시영 등) erialVersionUID)								
	진스런스	-			— ← MIO(0) 0) ► detail			riaiversionUID)								
			Deserialize (♀	덕별화) = 바이트 t	메벨에 있는 data	를 가지고 인스턴스 생성	성 /필드값									
									$\overline{}$	$-\!-\!-$				$\overline{}$		-
		NO 315 5:3														
	바법 및 String 타		여 출력하는 코드를													
- primitive 타입.	lł string 타입의 값	을 바이트 또는 비	바이트 배열로 가공	하는 기능을 액체화	화한다.											
- primitive 타입: - 이렇게 중간에	ll string 타입의 값 서 가공해주는 클리	을 바이트 또는 비 배스를 사용하여 입	바이트 배열로 가공 입출력을 수행한다	하는 기능을 객체되	한 한다.											



	1									
	l									
	l									
	1									
	1									
	l									
	l									
	1									
	l									

1									
-									
1									