Object 클래스

==> 자바로 만든 클래스의 최상위 클래스 역활을 하는 클래스이다
특별한 기능을 가지진 않고 자바 상속 관계를 명확하게 하기 위한 역활을 주로 많이 한다.

1. equals()

비교 함수이다.
문제는 Object가 가지고 있는 이 함수는 내용을 비교하는 함수가 아니고
주소를 비교하는 함수이다.

참고]
우리가 String의 equals는 내용을 비교하는 함수로 알고 있는데
이것은 String가 Object의 이 함수를 오버라이드하여 기능을 내용 비교로 변경해 놓았기 때문이다.

참고]
만약 우리가 만든 클래스도 내용 비교를 하도록 원하면 이 함수를 오버라이드하여 기능을 수정하면 된다.

2. toString()

자바는 주소를 내부적으로 사용하고 있다.
하지만 주소를 개발자에게 노출하지 않는다.

대신 주소를 출력하면 "클래스이름@해쉬코드값"을 출력하게 된다.

참고 해쉬코드값
==> 자바는 주소를 내부적으로 해쉬테이블을 이용해서
관리한다.
해쉬코드값은 그 주소를 관리하는 해쉬테이블의
코드값이 된다.

문제는 주소를 출력하려고 하는 순간 위의 모양을 만들어내야한다
==> 이 작업을 해주는 함수가 바로 toString()이다.

결론적으로 toString()는 주소를 출력할 때 **자동 호출되어서 출력할 내용을 만들어주는 함수**이다.

따라서 만든 클래스를 출력할 때 다른 내용(**클래스의 정보를 출력**하도록)하고자 할 경우 오버라이드해서 사용한다.

3. clone()

==> 자기 자신을 깊은 복사해주는 함수이다.

이 함수는 protected 함수이므로
상속을 받은 클래스나
같은 패키지에 있는 클래스에서만 사용할 수 있다.

4. hashCode

==> 자바는 주소를 노출하기 않도록 하기 위해서 해쉬 테이블을 이용해서 주소를 관리한다.
그 해쉬테이블에 주소를 관리하는 코드값을 알려주는 함수이다.

문제 1

원의 넓이와 둘레를 구할 수 있는 기능을 가진 클래스를 제작하세요
필요한 변수와 함수, 생성자는 생각에 따라서 만들어 사용하세요
단.

원의 반지름이 같으면 같은 클래스로 판정할 수 있도록
equals()를 오버라이드 하세요.

무제 2

사각형의 면적을 처리할 수 있는 클래스를 제작하세요
필요한 변수와 함수, 생성자는 생각에 따라서 만들어 사용하세요
단.

가로와 세로가 동시에 같은 클래스는 같은 클래스로 판정할 수
있도록 equals()를 오버라이드 하세요.

문제 3.

학생의 성적을 관리하는 클래스를 작성하세요 필요한 정보나 기능은 마음대로 꾸미세요.
단. 이 클래스를 출력하면 "???? 학생의 성적입니다."라고
출력되도록 하세요.

문제 4.

원의 넓이와 둘레를 구할 수 있는 기능을 가진 클래스를 제작하세요
필요한 변수와 함수, 생성자는 생각에 따라서 만들어 사용하세요
단.

단, 이 클래스를 출력하면 "넓이가 ???인 원 클래스입니다."
라고 출력하도록 하세요.

Math

주로 수학적인 기능을 처리하는 함수로 구성된 클래스이다.
모든 함수가 static 함수이므로 굳이 new 시키지 않아도 사용할 수 있는 클래스이며.....
아예 new 시키지 못하도록 해 놓은 클래스이다.

레퍼 클래스(Wrapper Class)

주소(객체타입)를 사용하는 어떤 곳에 Value Type을 사용할 수 없는 문제가 생길 수 있다.
이런 문제를 해결하기 위해서 만든 클래스를 레퍼 클래스라고 한다.
즉, 예를 들어 int를 주소로 사용할 수 있도록 하기 위해 만든 클래스

참고]

자바는 Object를 중심으로 계층 구조화 해서 클래스를 이용하도록 만드는 객체지향 언어이다.

그런데 유일하게 Value Type(int, float, ...) 만큼은 객체지향 원리를 사용하지 않는다.
그래서 다형성 구현에 있어서 약간의 문제가 생겼다.

예]

```
void abc(Object o) {}
```

==> 이함수는 모든 내용을 처리하도록 만든 함수이다.
그러나 유일하게 Value Type(기본 데이터타입)은 줄 수 없다.

참고]

Boxing과 Unboxing

Boxing : Value Type을 주소 타입(객체타입)으로 변환하는 것
Unboxing : 주소타입(객체타입)을 Value Type(기본 데이터타입)으로 변환하는 것.

레퍼 클래스란?

Boxing 과 Unboxing 을 해주기 위한 클래스

참고] 자바 버전 1.4이후 부터는 Boxing과 Unboxing이 자동으로 처리된다.

예]

```
Integer     a = 10;     ==> 자동 박싱된다.
int         b = a;      ==> 자동 언박싱이 된다.
```

결과적으로 자바는 내부적으로 모든 정보는 주소(객체)로 통일되어서 사용할 수 있다.

참고]

다만 유틸리티적인 몇개의 함수는 아직 이용 가능하므로 완전히 버려진 클래스는 아니다.

예] Integer.parseInt()

java.util 패키지 속에 있는 유용한 클래스

1. Objects

==> 이 클래스는 100% static 함수로만 구성된 클래스로
클래스(객체)를 다룰때 유용한 몇가지 유틸리티적인 함수로 구성된 클래스이다.

예]

isNull(Object obj)

==> 객체에 내용이 있는지를 확인하는 함수
즉, new 결과가 있는지를 확인하는 함수

참고 null이란?

==> 주소 변수에 주소가 없는 상태를 말하는 것이다.
이것은 찾아갈 대상이 없다는 말이고
이 말은 곧 이것은 아직 사용할 준비가 되지
않았다는 말이다.

2. Random 클래스

==> 난수를 발생하는 클래스이다.
우리가 알고있는 Math.random()보다는
좀더 다양한 형태의 난수를 발생할 수 있는 장점이 있다.

참고 **seed**란?

컴퓨터에 난수 발생은 그 순간에 실제로 난수를 만들어주는 것이 아니고
이미 컴퓨터 내부에는 난수 테이블이 존재하고 있다.
그리고 이 난수 테이블에서 순서대로 꺼내서 주는 역활이다.
seed란? 난수 테이블에서 난수를 꺼내는 위치를 말하는 용어이다.

3. Arrays 클래스

역시 100% static 함수로만 구성된 클래스로
배열을 처리할 때 필요한 유틸리티적인 함수들로 구성된
클래스이다.

예]　sort(???)
　　==>　배열에 있는 데이터를 정렬하는 기능을 가진 함수이다.

정규표현식 검사

주어진 문자열이 특정한 규칙에 맞도록 만들어져 있는지를 확인하는 기능

예]

주민등록번호가 6자리-7자리 숫자로 구성되어 있는지?
아이디가 순수하게 숫자와 문자로만 구성되어 있는지?
아이디가 최소 몇글자 이상 구성되어 있는지?

사용하는 클래스

java.util.regex.Pattern

==> 정규식 검사에 사용할 **정규식 문법을 지정할 클래스**

java.util.regex.Matcher

==> **정규식 검사를 실제로 실행**할 클래스

정규표현식 문법

1. c[a-z]*
   ==> 글자를 그냥 기록하면 반드시 그 위치에 그 글자가 와야한다.
   []안에 기록하면 그 중 하나가 와야한다.
   *은 바로 앞에서 지정한 글자가 0개이상 올 수 있다.
   ==> c로 시작하고 알파벳 소문자만 0개 이상 오면 된다.

2. c[a-z]
   ==> c로 시작하고 다음 글자는 반드시 알파벳 소문자 이어야 한다.

3. c[a-zA-Z]
   ==> c로 시작하고 다음 글자는 알파벳 문자가 와야 한다.

4. c[a-zA-Z0-9]
   ==> c로 시작하고 다음 글자는 알파벳 문자나 숫자가 와야 한다.
   c\w로 줄여서 사용할 수 있다.

5. .*
==> .은 모든 문자(알파벳과 한글까지 포함)를 의미한다.
==> 무슨글자가 와도 상관이 없다.

6. c.
==> 첫글자는 c로 시작하고 다음은 한글자만 무슨글자가 와도 상관이 없다.

7. c.*
==> 첫글자는 c로 시작하고 다음은 무슨글자가 몇글자가 와도 상관이 없다.

8. c\.
==> \.은 반드시 .만 와야 한다.

9. c\d
==> 숫자 문자를 의미한다.
==> 첫글자는 c가 와야하고 다음은 반드시 숫자가 와야 한다.
c[0-9]와 동일한 의미를 같는다.

10. c.*t
==> 첫글자는 c가 와야하고 몇글자가 와도 상관이 없지만 마지막 글자는 t가 와야 한다.

11. [b|c].*
==> b 또는 c 둘중 하나가 반드시 와야 한다.
==> [bc]로 사용해도 동일한 효과이고
[b-c]로 사용해도 동일한 효과이다.

12. [^b|c]
==> ^의 의미는 NOT의 의미이다.

13. .*a.*
==> a라는 문자가 한글자라도 포함되면 된다.

14. .*a.+
==> *는 0글자 이상 몇글자가 되어도 상관없지만
+는 1글자 이상 몇글자가 되어도 상관없다.

15. [b|c].{2}
==> {2}는 글자수를 의미한다.

16. .{2,3}
==> {2, 3}는 글자수를 의미한다. OR의 의미를 가진다.

예] 0[0-9]{2}-[0-9]{3, 4}-[0-9]{4}
0[1-9][0-9]-[0-9]{3, 4}-[0-9]{4}

문제 1]
아이디를 강제로 입력하고 그 아이디가 원하는 형태의 아이디인지를 검사하는 프로그램을 작성하세요
단, 아이디는 첫글자는 반드시 영문자로 시작해야하며
두번째 글자부터는 영문자와 숫자를 혼용할 수 있고
글자수는 8글자 이상이 되어야 한다.

힌트] {8,} ==> 8글자 이상을 의미하는 정규식 문법이다.

Scanner

==> 외부장치를 이용해서 데이터를 받아들이기 위한 클래스이다.

원래 자바는 IO 라는 개념을 이용해서 외부장치와 연결한다.
그러다보니 간단한 테스트를 위해서도 IO 처리를 해야하므로 개발자가 매우 불편하였다.

1.5부터 만들어진 기능이다.

생성자
==> 외부장치인 File 이나 IO인 Stream을 넣어달라고 하고 있다.

참고

System.in 이라는 **스트림**이 있다.
==> 이것은 내부적으로 키보드하고 연결한 스트림으로 만들어 놓은 것이다.

우리가
Scanner scan = new Scanner(System.in);
==> 키보드를 통해서 입력받는 Scanner 가 된것이다.

BigDecimal

==> 정수로 표현할 수 없는 숫자를 사용하기 위한 클래스이다.

주로 데이터베이스와 연동할 때
오라클의 데이터 형태는 NUMBER(숫자)
==> 오라클의 숫자는 38자리 까지 사용할 수 있다.
32자
이런 데이터는 자바에서는 처리할 수 없으므로
이런 유형의 데이터를 처리하기 위해서 만들어진 클래스이다.

예>

```
BigDecimal big = new BigDecimal("12345678901234567890");
```

형식화 클래스

데이터를 지정된 형식의 문자열로 변환해주는 클래스

1. DecimalFormat

==> 처리결과중 숫자를 보기 좋게 만들기 위한 클래스이다.

생성방법

DecimalFormat(String pattern) 을 주로 이용한다.

==> 어떤 모양으로 변환시킬지를 알려주는 패턴을 알려주면 만들어 준다.

주요함수

format(double number)

==> 숫자를 원하는 형태로 변환시켜주는 함수

패턴 지정법]

0	한자리 숫자를 의미하며 무효숫자도 표시하게 한다.
#	한자리 숫자를 의미하며 무효숫자는 표시하지 않게 한다.
.	소수의 위치를 알려준다.
-	음수일 경우 - 부호의 위치를 알려준다.
,	3자리마다 , 표시를 하도록 한다.
E	실수일 경우 지수 형태로 표시하도록 한다.
:	패턴을 구분한다. (양수일 경우와 음수일 경우를 구분해서 처리할 경우 사용)
%	백분율 표시를 하도록 하다.
\u00A4	화폐단위를 표시하도록 한다.

2. SimpleDateFormat

==> 처리결과중 날짜를 보기 좋게 만들기 위한 클래스이다.

생성방법과 주요함수는 위와 동일하다.

3. ChoiceFormat

==> switch등을 이용해야 하는 경우에 이것을 간소화 시키기 위한 방식이다.
즉, 특정 범위의 값은 특정 문자열로 교체해서 반환해주는 기능을 가진 클래스이다.

생성방법]
ChoiceFormat(double[] limits , String[] formats)
double[] limits : 치환될 범위
String[] formats : 치환할 문자

4. MessageFormat

==> 특정 문자열에 특정 위치에 내용만 변경되는 경우
문자열 전체를 만들지 않고 변경되는 내용만 변화시켜서 하나의 문자열을 만들어내는 클래스이다.

예]
우리가 오라클에 INSERT 명령을 자바로 만든다고 할 경우

"INSERT INTO emp VALUES(1234, '저기자', 'MANAGER', ...)"
"INSERT INTO emp VALUES(1111, '이기자', 'MANAGER', ...)"
"INSERT INTO emp VALUES(2222, '장독대', 'MANAGER', ...)"
"INSERT INTO emp VALUES(3333, '박아지', 'MANAGER', ...)"

이런 문자열을 여러개 만들 경우 처음부터 다시 만들어야 하는 불편함이 있다.
이것을 해결하는 클래스가 MessageFormat이다.

참고함수]
parse(String source)
==> 주어진 문자열 중에서 실제 변화되는 데이터 부분만 알아내는 함수이다.

스트림(Stream)

자바는 외부장치와 데이터를 주고 받은 방법을 하나의 방법으로 통일시켜 놓았다.
그것이 바로 **Stream** 이다.

스트림을 빨대로 생각하면 정확한 원리가 이해된다.

즉, 스트림을 데이터를 주고 받을 외부장치에 꽂기만 하면
나는 빨기만 하면(같은 함수를 이용하면)
외부장치의 데이터가 들어온다. 라는 개념이다.

참고]

스트림은 단향방이다.
즉, 하나의 스트림은 한방향으로만 데이터를 보낼 수 있다.
만약 같은 외부장치에 데이터를 주기도 하고 받기도 하고자 하면 스트림이 두개 있어야 한다.

스트림의 종류(방향성)

1. 프로그램으로 데이터가 들어오는 방향
   InputStream, Reader
2. 프로그램에서 데이터가 나가는 방향
   OutputStream. Writer

스트림의 종류(데이터의 양)

1. byte 단위
   InputStream, OuputStream
   ==> 속도는 조금 느리다....(기계어 처리도 가능하다)
2. char(2바이트) 단위
   Reader, Writer
   ==> 속도가 좀더 빠르다.....(오직 문자처리만 가능하다)

참고]

byte가 char로 합쳐지는 순간 운영체제에 따라서 비트 순서가 바뀔 수 있기 때문이다.

스트림의 종류(상대방 종류)

1. 타겟 스트림(기본 스트림)
   ==> 외부장치에 직접 연결되는 스트림
2. 필터 스트림
   ==> 스트림의 성능 향상, 개발자 편의를 위해서
   스트림에 연결되는 스트림

참고 반드시 기본 스트림은 존재해야 하고
필요에 따라서 필터 스트림을 연결해서 사용해야 한다.

Byte 단위 스트림

1. InputStream 의 기본 함수

int read()
==> 오직 한문자(1byte)만 읽어들이는 함수
반환값 읽은 문자

int read(byte[] b)
==> 여러 바이트를 읽어들이는 함수
읽은 결과는 byte[] b가 기억하게 되고
반환값 읽은 데이터 개수(바이트 수)

int read(byte[] b, int off, int len)
==> 여러 바이트를 읽어들이는 함수
다만, 배열의 지정한 위치부터 지정한 개수만큼만 읽어준다.
반환값 읽은 데이터 개수(바이트 수)

2. FileInputStream

==> 파일에 연결되는 입력용 스트림

생성방법]
FileInputStream(String name)
==> 스트림을 연결할 파일의 이름을 이용해서 스트림을 연결한다.

3. FileOutputStream

==> 파일로 연결된 보내는 방향의 스트림이다.

기본함수

void write(int b)

==> 한 글자만 내보내는 함수

void write(byte[] b)

==> 여러글자를 내보내는 함수

void write(byte[] b, int off, int len)

==> 여러 글자 중에서 지정한 부분만 내보내는 함수

int off, 내보낼 시작 위치

int len 내보낼 데이터 개수

File

이것은 스트림은 아니다.(데이터를 입력, 출력하는 기능이 없다.)
다만 우리가 외부장치 중에서 가장 많이 사용하는 외부장치가 파일이다보니.... 그 파일에 대한 정보를 처리하기 위한 클래스.

생성방법]

File(String pathname)

==> 파일의 경로와 이름을 지정해서 만든다.

File(String parent, String child)

==> 파일의 경로와 이름을 따로 지정해서 만든다.

File(File parent, String child)

==> 파일의 정보를 이용하고 파일의 이름만 다시 지정해서 만든다.

참고]

자바에서의 파일은 폴더 자체만도 파일로 인정한다.

예] File f = new File("D:\\");

File(URI uri)
==> 네트워크에서 다른 시스템에 있는 파일의 정보를 만든다.

참고함수]

1. 정보를 알아내는 함수
   - getName() 파일의 이름만 알아내는 함수
   - getParent() 폴더이름만 알아내는 함수
   - exists() 파일의 존재 여부를 알아내는 함수
   - length() 파일의 크기를 알아내는 함수
   - .....

2. 외부적인 작업을 하기 위한 함수
   ==> 데이터입출력 이외에 파일 전체에 대해서 필요한 작업을 할 수 있는 함수들....
   - delete() 파일 삭제
   - mkdir() 폴더 생성(하나만 만들수 있고)
   - mkdirs() 폴더 생성(계층으로 만들 수 있다.)
   - renameTo 파일이름 변경
   - list() 폴더에 있는 파일의 목록을 구하는 함수
   - ...

참고]

String[] list() 이름만 알아낸다.
File[] listFiles() 정보를 알아낸다.
==> 일반적인 파일의 목록을 알아낸다.

String[] list(FilenameFilter filter)
File[] listFiles(FileFilter filter)
File[] listFiles(FilenameFilter filter)
==> 필터링을 해서 리스트를 구할 수 있다.
즉, 필요한 파일만 골라서 목록을 구할 수 있다.

Filter Stream(보조스트림)

1. 기본 스트림에 기능을 추가하거나,
2. 사용자의 편의를 제공하기 위한 보조적인 기능을 하는 스트림이다.

주의]

반드시 기본 스트림은 존재해야 하며
필터 스트림은 필요에 따라서 연결해서 사용하면된다.

참고]

필터 스트림 역시 방향성을 가지고 있으며....
반드시 같은 방향끼리만 연결해야 한다.

★★★

1. BufferedInputStream/BufferedOutputStream

사용자 편의 기능은 없고
대신
스트림의 성능을 향상시키는 역활을 한다.
버퍼에 데이터를 모았다가 한번에 처리하는 기능이 추가된 스트림이다.

★★★

주요함수]

flush()
==> 강제로 버퍼를 비우는 함수

2. DataInputStream/DataOutputStream

많이 사용하는 스트림은 아니다.
기능 향상은 없고
대신
보통 스트림은 반드시 byte[]로만 데이터를 처리한다.
문자 데이터가 아닌 다른 형태의 데이터는 문제가 있을 수 있다.
==> **자바의 데이터형을** byte[] 변환하지 않고
직접 외부장치로 입출력하는 기능을 추가한 것

잘 사용하지 않는 이유?

DataOutput을 이용한 경우는 DataInput으로 받아야 한다.
순서가 지켜져야 한다.
(저장할 때 int, float -> 받을때도 int, float의 순서대로 받아야 한다.)

PrintStream

필터 계열의 스트림이다.
사용자 편의성 + 기능 향상을 동시에 만들어 놓는 보조 스트림

기능

1. 내부적으로 Buffered와 연결해 놓았다.
2. 사용자 편의를 위해서 자바의 데이터형태를 그래도 출력하도록 해 놓았다.
3. 파일과 직접 연결하도록 해 놓았다.
   (다른장치와도 연결이 가능)

단점]

한쌍을 이루는 입력쪽 스트림이 없다.

char 단위의 스트림

한번 입출력할 때 2바이트 단위로 입출력할 수 있는 스트림

주의]

기계어 상태의 내용은 절대로 char 단위 입출력을 사용하면 안된다.
이것은 오직 텍스트 문서에 한해서만 사용해야 한다.

기계어 문서는 반드시 byte 단위의 입출력을 사용해야 한다.
왜? 2byte가 1char가 될때 바이트의 순서가 역전이 된다.

예> 가

ㄱ 0011

ㅏ 0101 이라고 가정하면

가 00110101 이라고 생각하겠지만
01010011 로 바뀐 순서로 처리된다.

1. FileWriter/FileReader

==> 파일에 직접 연결된 타겟 스트림이다.
단위는 char 단위가 된다.

내부적으로 char 단위로 처리한다는 것만 변경되었을 뿐
사용자 입장에서는 byte 단위로 별 차이없이 사용하면 된다.

2. 관련된 필터 클래스

1) BufferedReader/BufferedWriter

==> 중간에 버퍼 기능을 추가해서 스트림의 성능 향상을 위한 보조 스트림 클래스

참고]

readLine() ==> 한줄 단위로 문자열을 그대로 읽어서 사용할 수 있는 함수

참고]

이 함수를 사용할 때 주의사항
이 함수는 줄 단위로 한줄씩 읽는 함수이다.
(이 것은 줄 단위(\r\n)를 발견하면 그곳에서 읽는것을 멈춘다.)

참고]

필요하면 줄단위를 강제로 읽어야 한다.
이 함수는 줄 단위를 삭제한다.
(즉, 줄 단위(\r\n) 까지 읽은 후 이 줄 단위 기호를 버린다.)

2. PrintWriter

PrintStream처럼 필터 계열의 스트림

1. 버퍼의 기능을 가지고 있다.
2. 직접 파일에 연결할 수 있다.
3. 사용자 편의를 도모한다(자바의 데이터형을 그대로 출력한다.)
4. 다른 byte[]의 스트림과도 연결할 수 있다.

참고

스트림을 연결할 때 주의사항

1. 반드시 같은 방향끼리만 연결할 수 있다.
2. 반드시 같은 크기만 연결할 수 있다.

예>

```
FileInputStream     fin = new FileInputStream("????");
BufferedReader      br = new BufferedReader(fin);
==>     틀린 경우이다.
```

참고 스트림

InputStreamReader/OutputStreamWriter

크기가 다른 두개의 스트림을 연결할 때 사용하는 보조 스트림의 일종이다.

예]

```
FileInputStream         fin = new FileInputStream("????");
InputStreamReader       temp = new InputStreamReader(fin);
BufferedReader          br = new BufferedReader(temp);
```

==> 올바른 경우

참고]

타겟이 char 이고 보조가 byte이면 연결이 되지 않는다.
반드시
타겟이 byte이고 보조가 char인 경우에만 연결할 수 있다.

객체의 직렬화

데이터는 자바의 기본형 이외에도 클래스 전체를 데이터로 상대방(외부장치)와 입출력할 수 있다.

이때는 함수는 제외되고
그 클래스안에 있는 변수의 내용만 입출력이 된다.

이처럼 클래스가 다른 장치에 전달되는 상태를
"직렬화"
라고 표현한다.

참고]
직렬화라고 부르는 이유
클래스에 있는 변수의 내용이 순서대로 입출력된다.. 라는 의미에서 붙여진 이름이다.

★★★
주의]
아무 클래스나 입출되는 것은 아니고
그 클래스가 반드시 직렬화 가능 클래스 이어야 한다.
직렬화 가능 가능 클래스는
Serializable 를 상속 받은 클래스이다.

문제 직렬화는 변수의 내용만 입출력되므로
네트워크 같이 서로 떨어진 경우에는
양쪽에 같은 클래스가 존재해야 한다.
(왜냐하면 클래스 구조는 전달되지 않고 변수 내용만
전달되기 때문에 양쪽 모두 클래스의 구조는 가지고 있어야 한다.)
이때 양쪽에 있는 클래스는 클래스이름 + 패키지이름이
모두 동일해야한다.

이때 사용하는 보조 스트림
ObjectInputStream/ObjectOutputStream
이것은 주로 네트워크에서 많이 사용
Reader/Writer 계열은 없다.

IO쪽에서 최소한 해야할 공부 ==> 필요한 데이터를 파일에 저장할 수 있고 파일에 저장된 내용을 불러오기 할 수 있도록 공부할것