1. API 도큐먼트에 대한 설명으로 틀린 것은 무엇입니까?

➊ 자바 표준 라이브러리를 프로그램에서 어떻게 사용할 수 있는지를 설명하고 있다.

➋ 클래스의 상속 관계 및 자식 클래스들이 무엇이 있는지 알 수 있다.

➌ 생성자 선언부, 필드의 타입, 메소드의 선언부를 확인할 수 있다.

➍ public, protected, default, private 접근 제한을 가지는 멤버들을 확인할 수 있다.ㅇㅇ

3. Object 클래스에 대한 설명 중 틀린 것은 무엇입니까?

➊ 모든 자바 클래스의 최상위 부모 클래스이다.

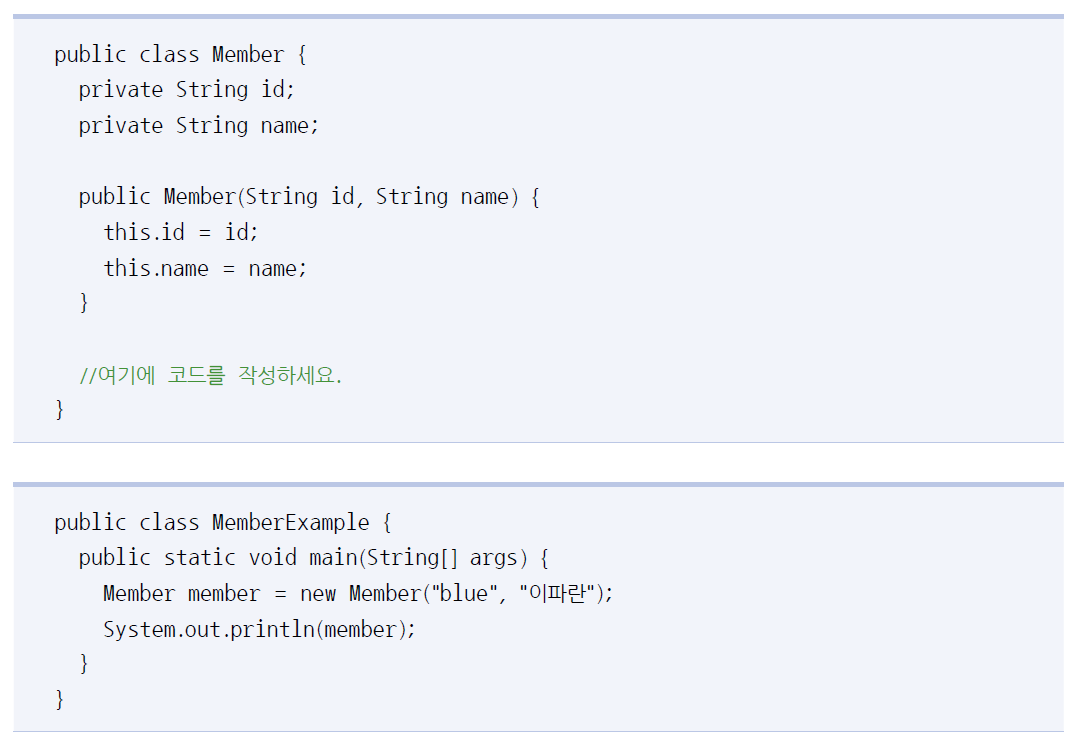
➋ Object의 equals ( ) 메소드는 == 연산자와 동일하게 번지를 비교한다.ㅇㅇ

➌ Object의 hashCode( ) 메소드는 동등 비교 시 활용된다.

➍ Object의 toString ( ) 메소드는 객체의 필드값을 문자열로 리턴한다.

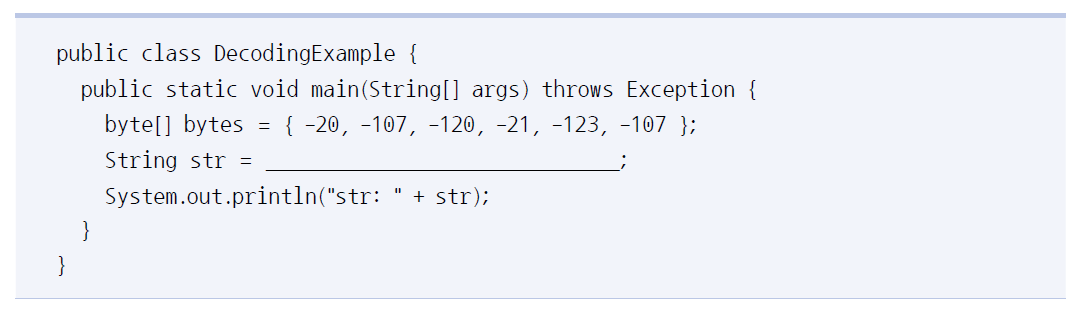
6. Member 클래스에서 Object의 toString ( ) 메소드를 오버라이딩해서 MemberExample 클래

스의 실행 결과처럼 나오도록 작성해보세요. ?????????????



9. 다음 바이트 배열은 UTF-8 문자셋으로 인코딩한 데이터로, 다시 문자열로 디코딩해서 변수 data

에 저장하려고 합니다. 밑줄 친 곳에 들어갈 코드를 작성해보세요.

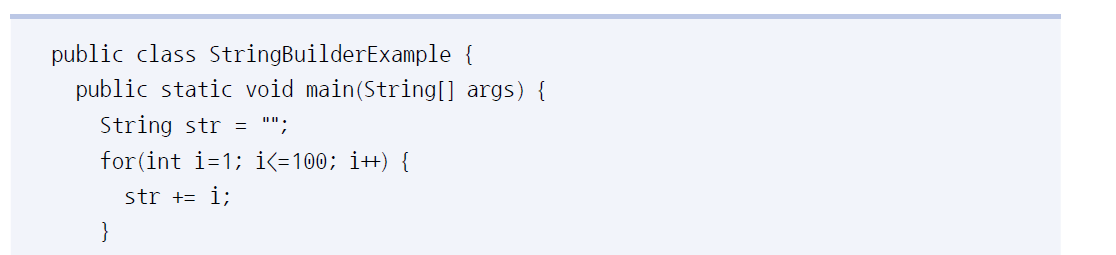


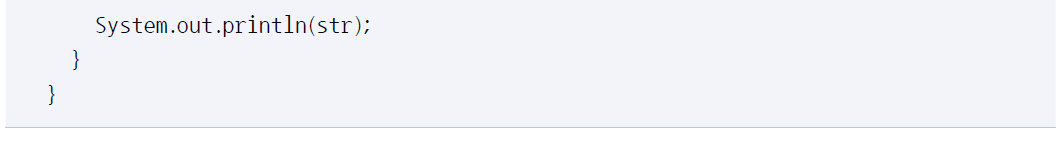
String str new String(bytes);

10. 다음 코드는 1부터 100까지의 숫자를 통 문자열로 만들기 위해 += 연산자를 이용해 100번 반

복하고 있습니다. 이것은 곧 100개 이상의 String 객체를 생성하는 결과를 만들기 때문에 좋은 코

드라고 볼 수 없습니다. StringBuilder를 사용해서 좀 더 효율적인 코드로 개선해보세요.





String str = new StringBuilder()

ArrayList<String> str = new ArrayList

str.appand(“123”)

for (int i = 0; i < 100 ; i++)

11. 다음 문자열에서 쉼표(,)로 구분되어 있는 문자열을 StringTokenizer를 이용해서 분리시키고

출력해보세요.



String data = “아이디, 이름, 패스워드”

StringTokenizer st = new StringTokenizer(data, “,”)

while (st.hasMoreTokens()) {

String token = st.nestToken();

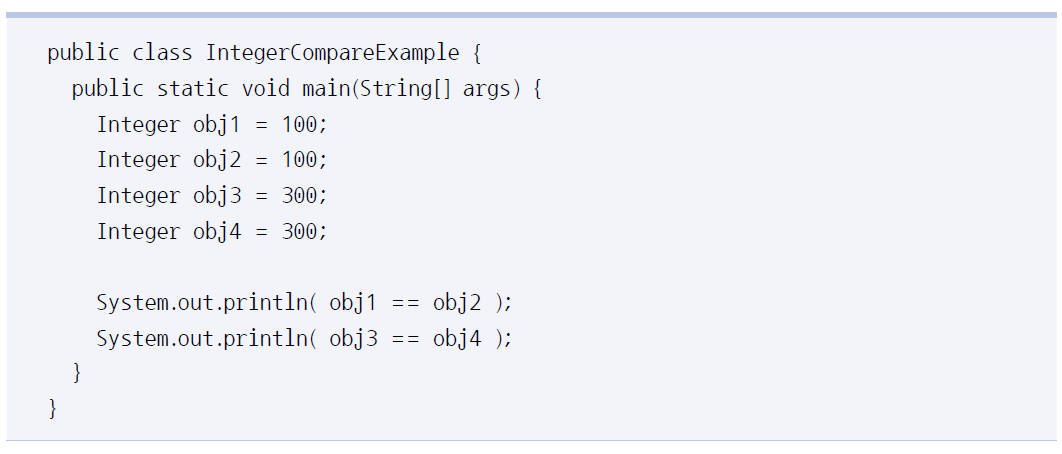
System.out.println(token);

}

12. 숫자 100과 300으로 각각 박싱된 Integer 객체를 == 연산자로 비교한 결과 100을 박싱한

Integer 객체는 true가 나오지만, 300을 박싱한 Integer 객체는 false가 나왔습니다. 그 이유를 설

명하고, 값만 비교할 수 있도록 코드를 수정해보세요.



13. Math 클래스가 제공하는 메소드의 리턴값이 잘못된 것은 무엇입니까? 4번

➊ Math.ceil (5.3) 6.0

➋ Math.floor (5.3) 5.0

➌ Math.max(5.3, 2.5) 5.3

➍ Math.round(5.7) 6.0 >> 0.06

16. SimpleDateFormat 클래스를 이용해서 오늘 날짜를 다음과 같이 출력하도록 코드를 작성해 보세요.



SimpleDataFormat sdf = new SimpleDataFormat (“yyyy+”년”+ MM+”월”+ dd+ “일” +a+”요일”+ HH+”시”+mm+”분”);

18. Class 객체에 대한 설명 중 틀린 것은 무엇입니까?

➊ Class.forName( ) 메소드 또는 객체의 getClass ( ) 메소드로 얻을 수 있다. ㅇㅇ

➋ 패키지와 클래스 이름을 알 수 있다.

➌ 클래스의 생성자, 필드, 메소드에 대한 정보를 알아낼 수 있다.

➍ getResource ( ) 메소드는 프로젝트 경로를 기준으로 리소스의 URL을 리턴한다.

19. 어노테이션(Annotation)에 대한 설명 중 틀린 것은 무엇입니까?

➊ 컴파일하거나 실행할 때 어떻게 처리해야 할 것인지를 알려주는 역할을 한다.

➋ 클래스, 필드, 생성자, 메소드를 선언하기 전에 @어노테이션을 붙일 수 있다.

➌ @어노테이션(“\*”)일 경우 value 속성값이 \*가 된다. ㅇㅇ

➍ @어노테이션(“\*”, prop=3)일 경우 value 속성값은 \*, prop 속성값은 3이 된다.