[06. 클래스]

1. 객체와 클래스에 대한 설명으로 틀린 것은 무엇입니까?

➊ 클래스는 객체를 생성하기 위한 설계도(청사진)와 같은 것이다.

➋ new 연산자로 클래스의 생성자를 호출함으로써 객체가 생성된다.

➌ 하나의 클래스로 하나의 객체만 생성할 수 있다.

➍ 객체는 클래스의 인스턴스이다.

2. 클래스의 구성 멤버가 아닌 것은 무엇입니까?

➊ 필드(field )

➋ 생성자(constructor )

➌ 메소드(method)

➍ 로컬 변수(local variable )

3. 필드, 생성자, 메소드에 대한 설명으로 틀린 것은 무엇입니까?

➊ 필드는 객체의 데이터를 저장한다.

➋ 생성자는 객체의 초기화를 담당한다.

➌ 메소드는 객체의 동작 부분으로, 실행 코드를 가지고 있는 블록이다.

➍ 클래스는 반드시 필드와 메소드를 가져야 한다.

4. 필드에 대한 설명으로 틀린 것은 무엇입니까?

➊ 필드는 메소드에서 사용할 수 있다.

➋ 인스턴스 필드 초기화는 생성자에서 할 수 있다.

➌ 필드는 반드시 생성자 선언 전에 선언되어야 한다.

➍ 필드는 초기값을 주지 않더라도 기본값으로 자동 초기화된다.

5. 생성자에 대한 설명으로 틀린 것은 무엇입니까?

➊ 객체를 생성하려면 생성자 호출이 반드시 필요한 것은 아니다.

➋ 생성자는 다른 생성자를 호출하기 위해 this ( )를 사용할 수 있다.

➌ 생성자가 선언되지 않으면 컴파일러가 기본 생성자를 추가한다.

➍ 외부에서 객체를 생성할 수 없도록 생성자에 private 접근 제한자를 붙일 수 있다.

6. 메소드에 대한 설명으로 틀린 것은 무엇입니까?

➊ 리턴값이 없는 메소드는 리턴 타입을 void로 해야 한다.

➋ 리턴 타입이 있는 메소드는 리턴값을 지정하기 위해 반드시return 문이 있어야 한다.

➌ 매개값의 수를 모를 경우 “…”를 이용해서 매개변수를 선언할 수 있다.

➍ 메소드의 이름은 중복해서 선언할 수 없다.

7. 메소드 오버로딩에 대한 설명으로 틀린 것은 무엇입니까?

➊ 동일한 이름의 메소드를 여러 개 선언하는 것을 말한다.

➋ 반드시 리턴 타입이 달라야 한다.

➌ 매개변수의 타입, 수, 순서를 다르게 선언해야 한다.

➍ 매개값의 타입 및 수에 따라 호출될 메소드가 선택된다.

8. 인스턴스 멤버와 정적 멤버에 대한 설명으로 틀린 것은 무엇입니까?

➊ 정적 멤버는 static으로 선언된 필드와 메소드를 말한다.

➋ 인스턴스 필드는 생성자 및 정적 블록에서 초기화될 수 있다.

➌ 정적 필드와 정적 메소드는 객체 생성 없이 클래스를 통해 접근할 수 있다.

➍ 인스턴스 필드와 메소드는 객체를 생성하고 사용해야 한다.

9. final 필드와 상수(static final )에 대한 설명으로 틀린 것은 무엇입니까?

➊ final 필드와 상수는 초기값이 저장되면 값을 변경할 수 없다.

➋ final 필드와 상수는 생성자에서 초기화될 수 있다.

➌ 상수의 이름은 대문자로 작성하는 것이 관례이다.

➍ 상수는 객체 생성 없이 클래스를 통해 사용할 수 있다.

10. 패키지에 대한 설명으로 틀린 것은 무엇입니까?

➊ 패키지는 클래스들을 그룹화시키는 기능을 한다.

➋ 클래스가 패키지에 소속되려면 패키지 선언을 반드시 해야 한다.

➌ import 문은 다른 패키지의 클래스를 사용할 때 필요하다.

➍ com.mycom 패키지에 소속된 클래스는 com.yourcom에 옮겨 놓아도 동작한다.

11. 접근 제한에 대한 설명으로 틀린 것은 무엇입니까?

➊ 접근 제한자는 클래스, 필드, 생성자, 메소드의 사용을 제한한다.

➋ public 접근 제한은 아무런 제한 없이 해당 요소를 사용할 수 있게 한다.

➌ default 접근 제한은 해당 클래스 내부에서만 사용을 허가한다.

➍ 외부에서 접근하지 못하도록 하려면 private 접근 제한을 해야 한다.

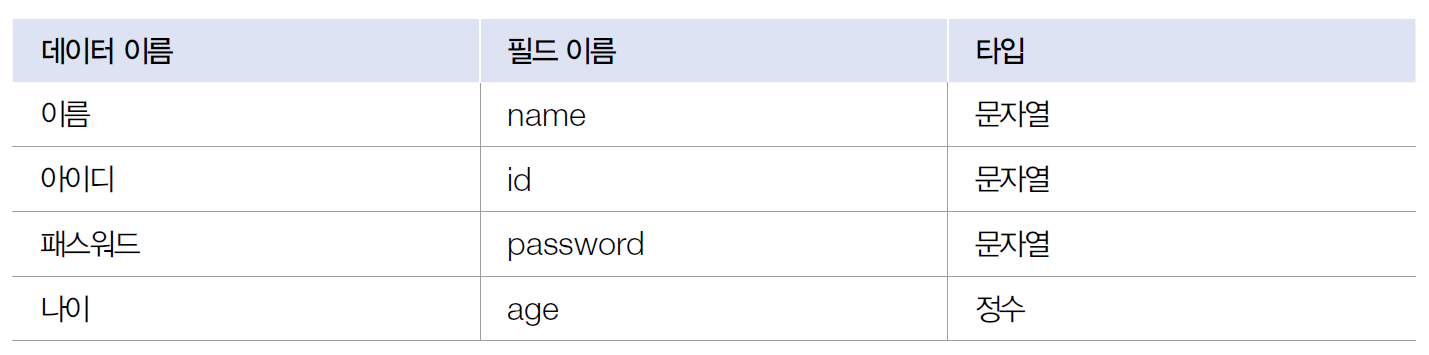
12. 다음 클래스에서 해당 멤버가 필드, 생성자, 메소드 중 어떤 것인지( ) 안에 적어보세요.

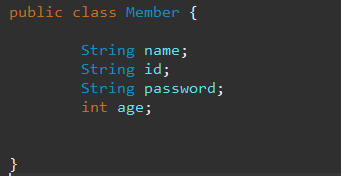
|  |
| --- |
| public class Member {  private String name; —---> ( )  public Member(String name) { … } —---> ( )  public void setName(String name) { … } —---> ()  } |

필드, 생성자, 메소드

13. 현실 세계의 회원을 Member 클래스로 모델링하려고 합니다. 회원의 데이터로는 이름, 아이디,

패스워드, 나이가 있습니다. 이 데이터들을 가지는 Member 클래스를 선언해보세요.

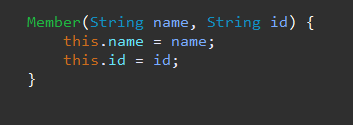




14. 13번 문제에서 작성한 Member 클래스에 생성자를 추가하려고 합니다. 다음과 같이 name

필드와 id 필드를 외부에서 받은 값으로 초기화하도록 생성자를 선언해보세요.

|  |
| --- |
| Member user1 = new Member("홍길동", "hong"); |



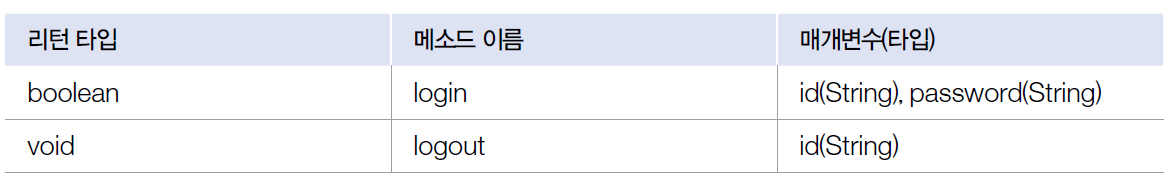
15. login ( ) 메소드를 호출할 때에는 매개값으로 id와 password를 제공하고, logout ( ) 메소드는

id만 매개값으로 제공하려고 합니다. 다음 조건과 예제 코드를 보고 MemberService 클래스에서

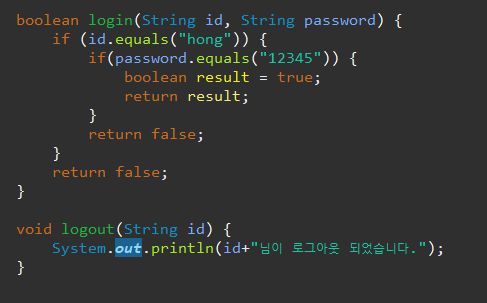
login ( ), logout ( ) 메소드를 선언해보세요.

➊ login ( ) 메소드는 매개값 id가 "hong", 매개값 password가 "12345" 일 경우에만 true로 리턴

➋ logout ( ) 메소드는 id + "님이 로그아웃 되었습니다"가 출력되도록 할 것



|  |
| --- |
| MemberService memberService = new MemberService();  boolean result = memberService.login("hong", "12345");  if(result) {  System.out.println("로그인 되었습니다.");  memberService.logout("hong");  } else {  System.out.println("id 또는 password가 올바르지 않습니다.");  } |



16. 다음은 컴퓨터 게임의 병사(marine)를 클래스로 정의한 것이다. 이 클래스의 멤버 중에 static을 붙여야

하는 것은 어떤 것들이고 그 이유는 무엇인가? (단, 모든 병사의 공격력과 방어력은 같아야 한다.)

|  |
| --- |
| class Marine {  int x=0, y=0; // Marine의 위치좌표(x,y) //static (x,y는 지역변수)  int hp = 60; // 현재 체력  int weapon = 6; // 공격력 //static (모든 병사의 공격력과 방어력이 같음)  int armor = 0; // 방어력 //static (모든 병사의 공격력과 방어력이 같음)  void weaponUp() {  weapon++;  }  void armorUp() {  armor++;  }  void move(int x, int y) {  this.x = x; // this.x는 인스턴스 변수, x는 지역변수  this.y = y; // this.y는 인스턴스 변수, y는 지역변수  }  } |

17. 다음 중 아래의 add메서드를 올바르게 오버로딩 한 것은? (모두 고르시오)

|  |
| --- |
| long add(int a, int b) { return a+b;} |

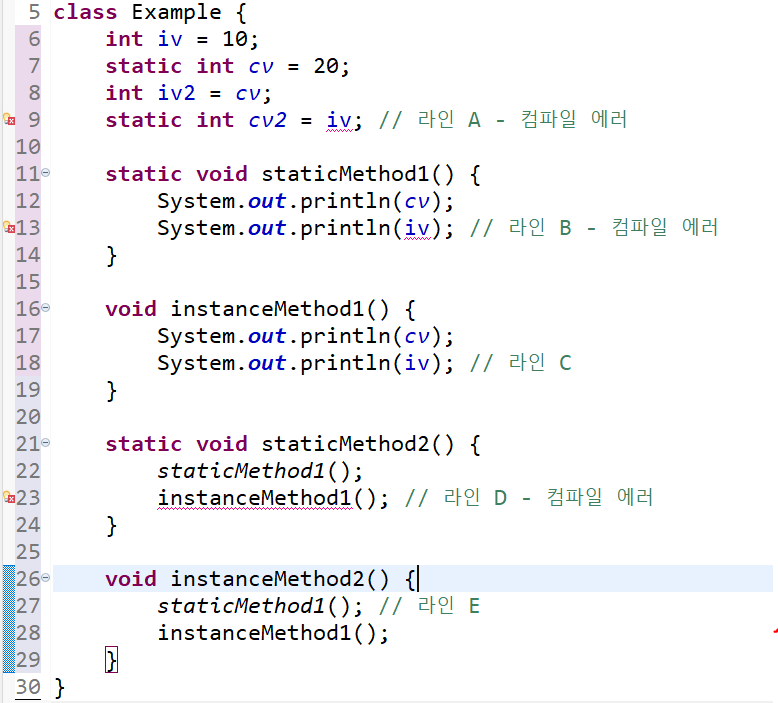
➊ long add(int x, int y) { return x+y;}

➋ long add(long a, long b) { return a+b;}

➌ int add(byte a, byte b) { return a+b;}

➍ int add(long a, int b) { return (int)(a+b);}

18. 다음 코드를 실행하면 컴파일 에러가 발생한다. 에러가 발생하는 이유를 설명하시오.



라인 A – static으로 선언한 변수에 인스턴스 변수를 넣을 수 없다.

라인 B - static으로 선언한 메서드에 인스턴스 필드를 넣을 수 없다.

라인 D - static으로 선언한 메서드에 인스턴스 메서드를 넣을 수 없다.

19. Tv클래스를 주어진 로직대로 완성하시오. 완성한 후에 실행해서 주어진 실행결과와 일치하는지

확인하라.

|  |
| --- |
| class MyTv {  boolean isPowerOn;  int channel;  int volume;  final int MAX\_VOLUME = 100;  final int MIN\_VOLUME = 0;  final int MAX\_CHANNEL = 100;  final int MIN\_CHANNEL = 1;  void turnOnOff() {  // (1) isPowerOn의 값이 true면 false로, false면 true로 바꾼다.  }  void volumeUp() {  // (2) volume의 값이 MAX\_VOLUME보다 작을 때만 값을 1증가시킨다.  }  void volumeDown() {  // (3) volume의 값이 MIN\_VOLUME보다 클 때만 값을 1감소시킨다.  }  void channelUp() {  // (4) channel의 값을 1증가시킨다.  // 만일 channel이 MAX\_CHANNEL이면, channel의 값을 MIN\_CHANNEL로 바꾼다.  }  void channelDown() {  // (5) channel의 값을 1감소시킨다.  // 만일 channel이 MIN\_CHANNEL이면, channel의 값을 MAX\_CHANNEL로 바꾼다.  }  } // class MyTv  class Example {  public static void main(String args[]) {  MyTv t = new MyTv();  t.channel = 100;  t.volume = 0;  System.out.println("CH:" + t.channel + ", VOL:" + t.volume);  t.channelDown();  t.volumeDown();  System.out.println("CH:" + t.channel + ", VOL:" + t.volume);  t.volume = 100;  t.channelUp();  t.volumeUp();  System.out.println("CH:" + t.channel + ", VOL:" + t.volume);  }  } |

20. 다음과 같이 정의된 메서드를 작성하고 테스트하시오.

-메서드명 : isNumber

-기능 : 주어진 문자열이 모두 숫자로만 이루어져있는지 확인한다.

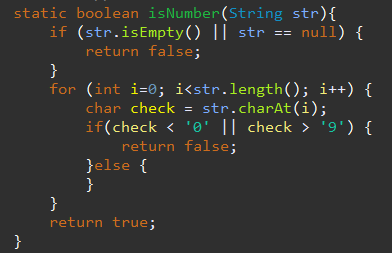
모두 숫자로만 이루어져 있으면 true를 반환하고, 그렇지 않으면 false를 반환한다.

만일 주어진 문자열이 null이거나 빈문자열“”이라면 false를 반환한다.

-반환타입 : boolean

-매개변수 : String str ->검사할 문자열

|  |
| --- |
| class Example {  //isNumber 메소드 작성  public static void main(String[] args) {  String str = "123";  System.out.println(str + "는 숫자입니까? " + isNumber(str));  str = "1234o";  System.out.println(str + "는 숫자입니까? " + isNumber(str));  }  } |



21. 다음과 같이 정의된 메소드를 작성하고 테스트 하시오.

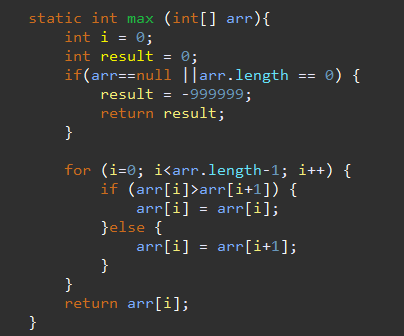
-메서드명 : max

-기능 : 주어진 int형 배열의 값 중에서 제일 큰 값을 반환한다. 만일 주어진 배열이 null이거나 크기가 0인 경우, -999999를 반환한다.

-반환타입 : int

-매개변수 : int[] arr -> 최대값을 구할 배열

|  |
| --- |
| class Example {  //max 메소드 작성  public static void main(String[] args) {  int[] data = { 3, 2, 9, 4, 7 };  System.out.println(java.util.Arrays.toString(data));  System.out.println("최대값:" + max(data));  System.out.println("최대값:" + max(null));  System.out.println("최대값:" + max(new int[] {})); // 크기가 0인 배열  }  } |



22. 은행 계좌 객체인 Account 객체는 잔고(balance) 필드를 가지고 있습니다. balance 필드는

음수값이 될 수 없고, 최대 백만 원까지만 저장할 수 있습니다. 외부에서 balance 필드를 마음대로

변경하지 못하도록 하고, 0 <= balance <= 1,000,000 범위의 값만 가질 수 있도록 Account 클래

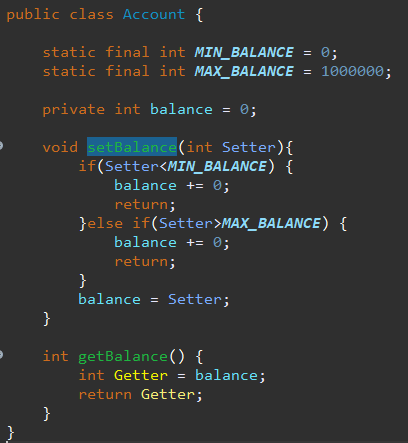
스를 작성해보세요.

➊ Setter와 Getter를 이용

➋ 0과 1,000,000은 MIN\_BALANCE와 MAX\_BALANCE 상수를 선언해서 이용

➌ Setter의 매개값이 음수이거나 백만 원을 초과하면 현재 balance 값을 유지

|  |
| --- |
| Account account = new Account();  account.setBalance(10000);  System.out.println("현재 잔고: " + account.getBalance()); //현재 잔고: 10000  account.setBalance(-100);  System.out.println("현재 잔고: " + account.getBalance()); //현재 잔고: 10000  account.setBalance(2000000);  System.out.println("현재 잔고: " + account.getBalance()); //현재 잔고: 10000  account.setBalance(300000);  System.out.println("현재 잔고: " + account.getBalance()); //현재 잔고: 300000 |



23. 다음은 키보드로부터 계좌 정보를 입력받아 계좌를 관리하는 프로그램입니다. 계좌는 Account

객체로 생성되고 BankApplication에서 길이 100인 Account[ ] 배열로 관리됩니다. 실행 결과를

보고, Account와 BankApplication 클래스를 작성해보세요(키보드로 입력받을 때는 Scanner

의 nextLine ( ) 메소드를 사용).





