

## JAVA 프로그래밍 – 중간고사 1/2

학년:            학번:

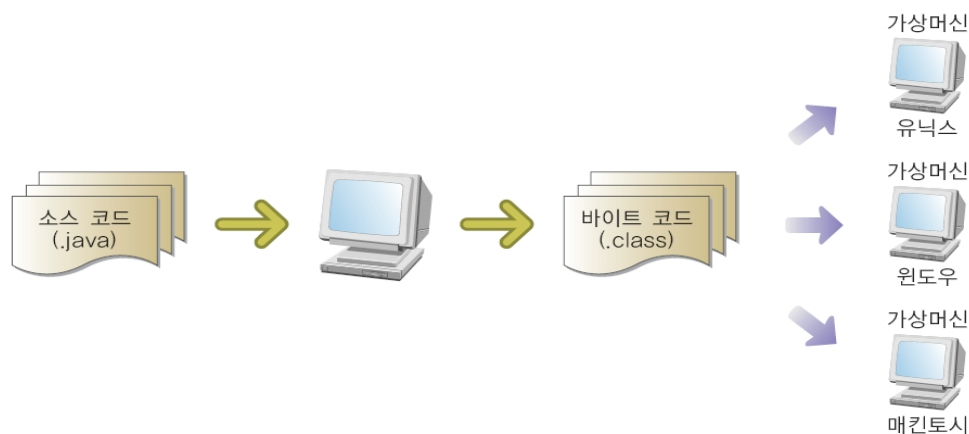
이름:            .

1. 자바 언어에 의해 생성된 프로그램의 특징을 간략히 설명하시오. (10점)

특정 플랫폼(linux, windows, os x등)에 종속된 native code를 생성하지 않고, 특정 플랫폼에 종속되지 않은 managed code를 생성한다. 자바 가상머신이 각각의 다른 플랫폼의 차이를 극복하고, 자바 프로그램에 플랫폼에 따라 동일한 환경을 제공하여 동일한 자바 프로그램이 각기 다른 플랫폼에 동작 가능하다.

문맥상 유사한 의도 및 자바에 대한 옳은 개념이 있으면 정답

2. 자바 소스코드가 각각의 다른 플랫폼에 실행되는 구조와 개발 단계에 대하여 간략히 설명하시오. (10점)



[그림 1-12] 자바 컴파일의 과정과 가상머신의 개념

운영체제->소스코드->컴파일->바이트코드->가상머신->실행결과

자바 컴파일러에 의해 자바소스코드(.java)는 바이트코드(.class)로 생성한다. 배포 된 바이트코드는 자바 가상머신에 의해 각각의 다른 플랫폼의 차이를 극복하고 각기 다른 플랫폼에 동작 가능하다.

문맥상 유사한 의도가 있으면 정답

3. 클래스(Class), 인스턴스(Instance)에 대하여 간략히 설명하시오. (10점)

클래스는 구현된 코드 즉 인스턴스를 생성하기 위한 형틀.

인스턴스는 구현된 클래스 등이 코드 내 선언에 의해 메모리 공간에 생성된 객체,

문맥상 유사한 의도가 있으면 정답, 인스턴스에 메모리 언급이 없으면 -4점

4. 절차 지향과 객체 지향에 대하여 각각 간략히 설명하시오. (10점)

절차지향 : 5점

데이터를 중심으로 구현을 한다.

각각의 기능을 수행하는 프로시저(함수)들이 데이터를 조작하거나 사용하는 방식으로 구현된다.

다양한 프로시저들이 데이터를 공유해서 동작하기 때문에 데이터의 변화는 곧 데이터를 사용하는 코드(프로시저)의 변화가 필요하다.

변화에 따른 유지보수 비용이 높다.

객체 지향 : 5점

데이터와 기능을 함께 담고 있는 객체들이 서로 대화하는 방법으로 구현된다.

객체 간에는 데이터를 (최대한) 주고 받지 않으며, 객체 상호간에 기능을 실행해 달라고 요청하는 방식으로 구현을 하게 된다.

각 객체들은 내부적으로 데이터를 담고 있지만 다른 객체에게 데이터를 노출하지는 않는다.

변화에 따른 유지보수 비용이 상대적으로 낮다.

문맥상 유사한 의도가 있으면 정답

5. 자바에서 추상클래스와 인터페이스의 공통점(1가지)과 차이점(3가지 이상)에 대해 간략히 설명하시오. (10점)

정답 :

공통점 : 4점

둘 다 추상 메소드를 포함, 자식 클래스가 추상 메소드(*abstract method*)를 구현

차이점 : 6점 , 한 가지=2점

추상 클래스는 추상 메소드가 아닌 메소드를 가질 수 있음

추상 클래스는 단일 상속만 가능 : 클래스이므로

인터페이스의 변수는 반드시 *static final* (상수)

인터페이스는 반드시 *public* 메소드 만을 포함

추상 클래스는 생성자를 정의할 수 있음

6. 아래 소스 코드가 주어진 출력 결과로 동작 하도록 \_\_\_\_ (밑줄)의 코드를 완성 하시오.(10점)

파일명	Hello_Test.java
소스코드	package java_exam;  public class ____{

	<pre> public static void main(String[] args) {     _____ } </pre>
출력결과	Hello Java!

정답

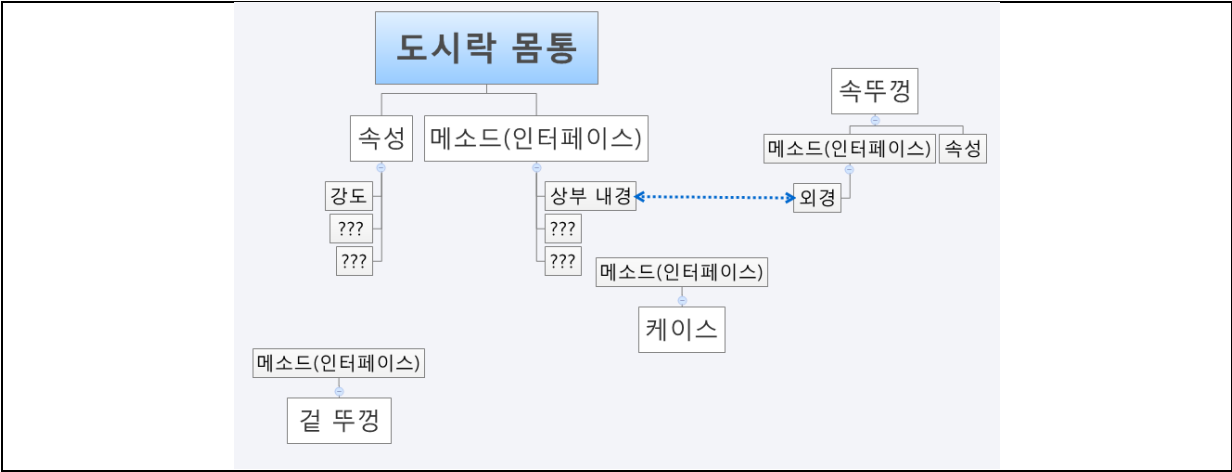
파일명	Hello_Test.java
소스코드	<pre> package java_exam;  public class Hello_Test{     public static void main(String[] args) {         System.out.println("Hello Java!");     } } </pre>
출력결과	Hello Java!

Hello\_Test : 5점 , 대소문자 구분 잘못 -3

System.out.println("Hello Java!"); : 5점, 세미콜론 없음 -3, 대소문자 잘못 -1,

7. 아래의 보온도시락과 그림의 맵을 참고하여 o표한 "도시락 몸통"의 객체에 대하여 속성과 메소드(or 인터페이스)를 구분하여 각각 2가지 이상 더 제시하여 아래 표를 완성하시오. (10점)

 <p>케이스 및 전체 구성품의 체결 상태</p>	 <p>전체 구성품( 케이스, 속 뚜껑, 겉 뚜껑 도시락 몸통 )</p>
 <p>속 뚜껑과 도시락 몸통</p>	 <p>속 뚜껑과 도시락 몸통의 체결 상태 및 겉 뚜껑</p>



객체 이름	속성	메소드(인터페이스)
도시락 몸통	강도	상부 내경 -> 속 뚜껑 ( 외경 )

정답

객체 이름	속성	메소드(인터페이스)
도시락 몸통	강도 보온 능력 진공 층의 두께 저장 용량 ...	상부 내경 -> 속 뚜껑 ( 외경 ) 상부 외경 -> 겉 뚜껑 ( 내경 ) 지름 및 길이 -> 케이스 ( 내부 크기 ) ...

문맥상 유사한 의도가 있으면 정답

8. 아래코드의 예상 결과를 작성하시오. (10 점)

```
package java_exam;

interface IUserMath{
    abstract void sum( );
    abstract void sub( );
}

class Exception6 implements IUserMath
{
    void sum( ){ }
    void sub( ){ };
}
```

```

public static void main(String args[]) {
    int i=2;
    int j=1;
    int nAry[] = { 6, 7, 8, 9 };
    int nTmp = 0;

    try{
        if( (i++/(j)--) > 1){
            i++;
        }
        nTmp = nAry[i];
    }

    catch(ArithmeticException e){
        System.out.println("arithmetic error.");
    }

    catch(ArrayIndexOutOfBoundsException e){

        System.out.println("Array index out of bounds");
    }

    catch(Exception e){
        System.out.println("generic exception.");
    }

    finally{
        System.out.println("In the finally block");
    }
}

```

정답

Array index out of bounds  
In the finally block

정답이 아니면 부분 배점 없음.

9. 아래 소스 코드를 보고 물음에 답하시오. (20점)

- 아래 소스 코드가 주어진 출력 결과로 동작 하도록 \_\_\_\_ (밑줄)의 코드를 완성 하시오.(10점)
- main()함수에서 찾을 수 있는 오류를 세가지를 찾고 수정할 수 있는 방법에 대하여 기술하시오(10점).

```

package java_exam;

abstract class AbstractClass {
    abstract void abstractMethod();
}

interface IBicycle{

    int FrameSize_L = 18;

```

```

int FrameSize_M = 16;
int FrameSize_S = 15;

abstract void prtInfo( );
}

interface IMotor{
    int Motor_Size_Small = 20;
    int Motor_Size_Large = 100;

    abstract void start( );
    abstract void stop( );
}

class MotorBicycle _____{
    private int id;
    private String brand;

    public void prtInfo( ) {
        System.out.println("ID : " + id);
        System.out.println("Brand : " + brand);
        System.out.println("Frame Size : " + IBicycle.FrameSize_M);
        System.out.println("Motor Size : " + IMotor.Motor_Size_Large);
    }

    public void start( ) {
        System.out.println("## 모터 start!!");
    }
    public void stop( ) {
        System.out.println("## 모터 stop!!");
    }
}

class FindMistakes{
    public static void main(String[] args) {
        MotorBicycle mb = new MotorBicycle( );
        mb.id = 0x6FFFFFF0 * 0x000FFFFF;
        mb.brand = "Yamaha";
        mb.prtInfo( );
        mb.start ( );
        mb.stop ( );
    }
}

```

```

ID : -1895825392
Brand : Yamaha
Frame Size : 16
Motor Size : 100
## 모터 start!!
## 모터 stop!!

```

정답 :

a. \_\_\_\_ (밑줄)의 코드 : **implements** IBicycle, IMotor 정답이 아니면 부분 배점 없음.

b. 오류 1 : private 멤버의 접근, private→public 교체, `mb.id` = 0x6FFFFFF0 \* 0x000FFFFF;

오류 2 : private 멤버의 접근, private→public 교체, `mb.brand` = "Yamaha";

오류 3 : 오버 플로우, 오버 플로우되지 않는 낮은 수의 id발급, `mb.id` = 0x6FFFFFF0 \* 0x000FFFFF;

1개 : 3점 , 2개 : 6점, 3개 : 10점