1. JDK와 JRE의 차이점은?

해설: JRE -> 자바프로그램을 실행하기 위한 라이브러리, 자바가상기계, 기타 컴포넌트들을 제공. 단순히 실행만 하고 개발은 하지 않는 일반인들을 위한 환경

JDK -> JRE에 추가로 컴파일러, 디버거와 같은 명령어행 개발 도구를 추가한 것. JDK안에 JRE가 포함되어 있다.

2. JVM에 대해 아는 대로 설명해봐라

정답: 자바가상기계(소프트웨어), 다른 프로그램은 운영체제 위에서 바로 실행이 가능하지만 자바는 운영체제 위의 JVM에서 실행된다. JDK를 설치할 때 각 OS에 맞는 것을 선택해야 하는데 이는 OS마다 다른 JVM이 제공되기 때문이다.(덕분에 어떤 OS에서도 연동 가능, 독립적)

자바는 아래와 같이 두 단계로 나누어서 컴파일 되는데 그 이유는 모든 컴퓨터에서 실행되게 하기 위해서다.

자바 소스 코드(.java, src폴더에 위치) -> (컴파일) -> 바이트 코드(.class, bin 폴더에 위치) -> (JVM) -> 기계어 -> 실행

소스코드: 우리가 작성한 코드

바이트 코드: 자바 가상기계를 위한 코드. JVM이 바이트 코드를 기계어로 변환. 컴퓨터 구조에 독립적인 중간 코드.

3. API란?

- 많은 유용한 기능을 제공하는 클래스 라이브러리의 집합

4. 객체 지향 언어(자바)에서의 프로그램 개발 단위는?

- 클래스(메소드와 변수로 이루어짐)

5. 매개변수란?

- 외부의 데이터를 메소드로 전달하는 수단

6. 변수와 상수의 차이

변수: 값을 일시적으로 저장(값 변경 가능)

상수: 프로그램이 실행되는 동안 변하지 않는 수(값 변경 불가능)

7. 프로그램 개발 시 나타날 수 있는 오류 3가지에 대해 말해보시오.

컴파일 오류: 컴파일 시에 발견되는 오류. 실행X

실행 오류: 실행 도중 나타나는 오류

논리 오류: 컴파일도 되고 실행도 되지만 의도하지 않은 결과가 발생

8. 하나의 소스 파일에는 하나의 클래스만 있어야 하는가?

- 여러 개의 클래스가 있어도 된다. 하지만 public이 붙은 클래스는 하나만 있어야 한다.

- public으로 선언된 클래스의 이름이 파일 이름이 되어야한다.

9. 기초형과 참조형의 차이

기초형: 실제 값이 저장. 정수형, 실수형, 논리형, 문자형이 있다.

참조형: 객체를 가리키는 주소가 저장, 클래스 인터페이스, 배열이 있다.

10. 식별자 규칙을 말해보시오.

- 유니코드 문자와 숫자의 조합으로 이루어짐. 한글도 가능

- 첫 문자는 문자여야한다. _ 와 $도 가능

- 대문자, 소문자 구분

- 키워드 사용X

11. 기초형 자료의 종류는?

- byte, short, int, long, float, double, char, boolean

byte = -128 ~ 127

short = -32,768 ~ 32767

int = -21억 ~ 21억

long = l,L 붙여서 사용

float = 32bit, 정밀도 약 7개 유효숫자

double = 64bit, 정밀도 약 15개 유효숫자

- 금액 나타낼 때 실수형X

- 실수형은 double형이 기본이라 float형으로 선언 및 초기화할 시 F를 붙이지 않으면 double형 값으로 인식하여 오류발생

char = 하나의 문자를 ' ' 작은 따옴표로 감싸야 함. 2byte

* 구냥 정리

리터널: 소스코드에서 프로그래머에 의해 직접 입력된 값.

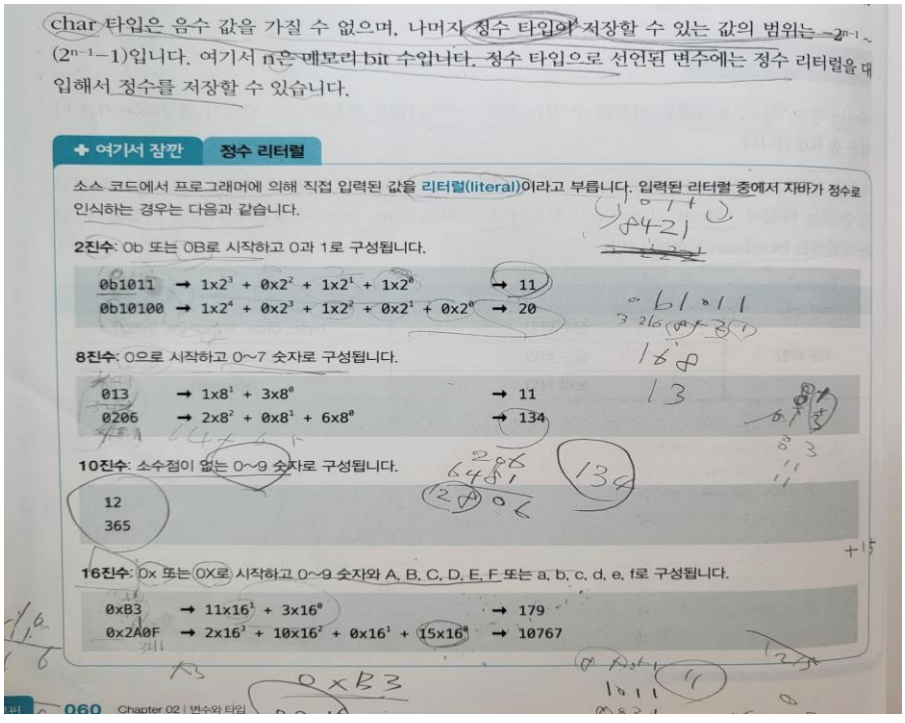
진수 리터널을 자바가 정수로 인식하게 하려면?

2진수: 0b, 0B로 시작하고 0과 1로 구성

8진수: 0으로 시작하고 0~7 숫자로 구성

10진수: 소수점이 없는 0~9 숫자로 구성

16진수: 0x, 0X로 시작하고 0~9, A~F, a~f로 구성



char 타입은 음수 값을 가질 수 없으며, 나머지 정수 타입이 저장할 수 있는 값의 범위는 -2^{n-1} ~ $(2^{n-1}-1)$입니다. 여기서 n은 메모리 bit 수입니다. 정수 타입으로 선언된 변수에는 정수 리터럴을 대입해서 정수를 저장할 수 있습니다.

✚ 여기서 잠깐 정수 리터럴

소스 코드에서 프로그래머에 의해 직접 입력된 값을 리터럴(literal)이라고 부릅니다. 입력된 리터럴 중에서 자바가 정수로 인식하는 경우는 다음과 같습니다.

2진수: 0b 또는 0B로 시작하고 0과 1로 구성됩니다.

```
0b1011  → 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0  → 11
0b10100 → 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0 → 20
```

8진수: 0으로 시작하고 0~7 숫자로 구성됩니다.

```
013  → 1x8^1 + 3x8^0          → 11
0206 → 2x8^2 + 0x8^1 + 6x8^0  → 134
```

10진수: 소수점이 없는 0~9 숫자로 구성됩니다.

```
12
365
```

16진수: 0x 또는 0X로 시작하고 0~9 숫자와 A, B, C, D, E, F 또는 a, b, c, d, e, f로 구성됩니다.

```
0xB3   → 11x16^1 + 3x16^0              → 179
0x2A0F → 2x16^3 + 10x16^2 + 0x16^1 + 15x16^0 → 10767
```

060 Chapter 02 | 변수와 타입

12. 기호 상수 만드는 법

- final 키워드 붙이기

13. for문 while, do-while문 차이점

- for문과 while문은 서로 변환이 가능하기 때문에 반복문을 작성할 때 어느 쪽을 선택해도 좋지만 for문은 반복 횟수를 알고 있을 때 주로 사용하고 while문은 조건에 따라 반복할 때 주로 사용한다. while문과 do-while문의 차이점은 조건을 먼저 검사하느냐 나중에 검사하느냐일 뿐 동작방식은 동일하다.

* 정리

객체지향프로그래밍

- 현실 세계에서 어떤 제품을 만들 때 부품을 먼저 개발하고 이 부품들을 하나씩 조립해서 제품을 완성하듯이 소프트웨어를 개발할 때도 부품에 해당하는 객체를 먼저 만들고 객체를 하나씩 조립해서 완성된 프로그램을 만드는 기법

- 현실 세계를 모델링

클래스

- 설계도

- 필드와 메소드로 구성

객체

- 자신의 속성을 갖고 있으면서 식별 가능한 것 ex) 자동차, 책, 사람, 강의, 주문

- 객체는 속성(필드: 색깔, 이름, 나이)과 동작(메소드: 웃다, 걷다, 달린다)로 구성된다.

- 클래스로부터 만들어지는 각각의 객체 = 인스턴스(동일한 설계도로 여려 대의 자동차를 만드는 것과 동일. 인스턴스들은 메소드는 공유하지만 데이터는 각각 가지고 있다.)

객체 모델링

- 현실 세계의 객체를 소프트웨어로 설계하는 것

- 추상화(현실 세계의 객체의 속성과 동작을 추려내어 소프트웨어의 필드와 메소드로 정의)

객체의 상호작용

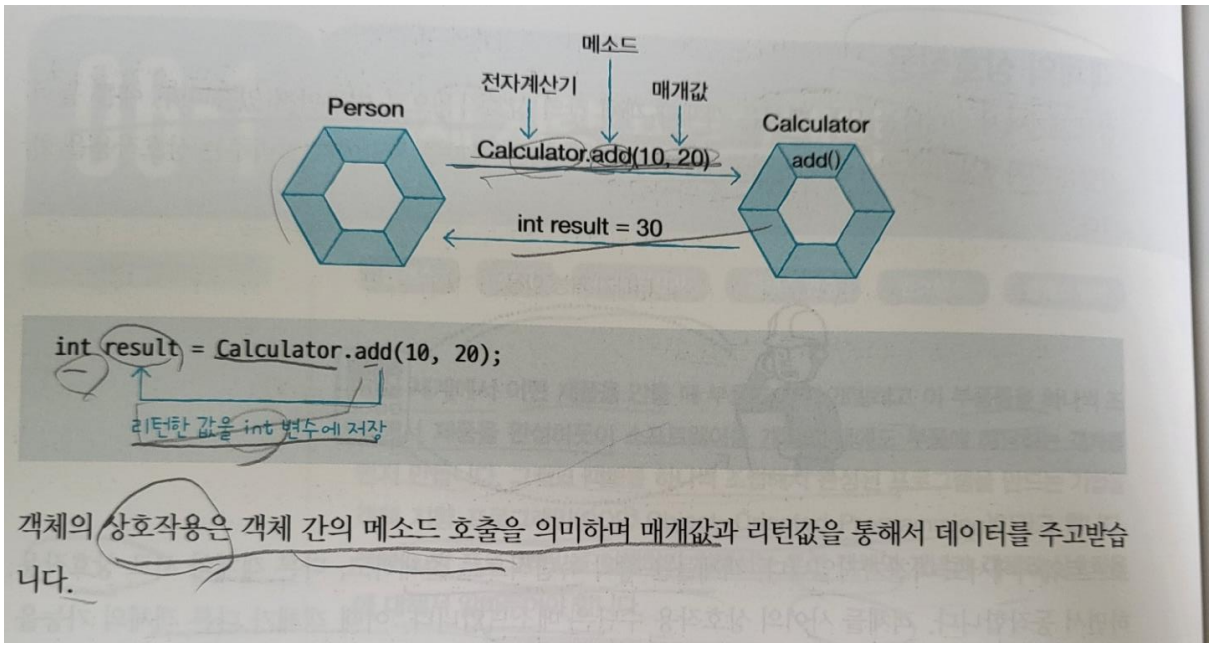
- 현실 세계에서 일어나는 모든 현상은 객체와 객체 간의 상호작용으로 이루어짐

Ex) 사람은 계산기를 사용하고 계산기는 계산 결과를 사람에게 알려줌

- 소프트웨어도 마찬가지로 다른 객체와 서로 상호작용하며 동작한다.

- 객체들 사이의 상호작용 수단은 메소드이다.

Ex)



14. 객체지향의 특징 3가지에 대해 설명해보시오.

(1) 캡슐(클래스)화

- 정보은닉(내부 데이터 보호), 알고리즘과 데이터가 묶여 있어서 재사용이 용이하고 독립성이 있다.

(2) 상속

- 기존 코드 재활용(부모 클래스를 이어 받아서 자식 클래스 생성. 자식 클래스는 부모 클래스의 속성과 동작을 물려받고 추가로 자식 클래스에만 필요한 기능이 있다면 추가 또는 변경할 수 있

다.)

(3) 다형성

- 객체가 취하는 동작이 상황에 따라서 달라지는 것을 의미(다양한 실행결과)

-

15. 하나의 참조 변수가 다른 참조 변수로 대입되면 어떤 일이 발생하는가?

- 동일한 객체를 참조하게 된다.

16. 객체를 소멸시키려면 어떻게 하면 되는가?

- 참조를 잃게 한다. (null대입, 객체를 가리키는 변수가 하나도 없게 만든다.)

* 자바에서 문자열은 객체이다.(String 클래스)

* 필드(전역변수): 전체 클래스 안에서 사용가능

* 지역변수: 메소드나 블록안에서만 사용가능

* 매개변수: 메소드 선언에서의 변수

17. 문자열 "123"을 int형으로 변환하는 코드를 써봐라.

- 168p

18. 접근자와 설정자 메소드를 사용하는 이유

- 설정자에서 매개변수를 통해 잘못된 값이 넘어오는 경우 이를 사전에 차단할 수 있다.

- 필요할 때마다 필드 값을 반환할 수 있다.

- 읽기만 가능한 필드를 만들 수 있다.

* 오버로딩(중복정의)

- 여러가지 데이터 타입에 같은 처리

- 매개변수의 타입이나 개수, 순서가 달라야 한다. 리턴 값이 다른 건X, 같은 이름 사용

* 생성자

- 객체의 초기화를 담당

- 반드시 public 수식어가 있어야한다.

- 오버로딩 가능, 반환값X

-

19. 매개변수가 있는 생성자는 정의하였지만 디폴트 생성자는 정의하지 않은 상태에서 디폴트 생성자를 이용해서 객체를 생성하면 어떻게 될까?

- 오류가 난다. 클래스에 명시적으로 선언한 생성자가 1개라도 있으면 디폴트 생성자가 자동으로 추가되지 않는다.

20. this()에 대해 설명하시오.

- 생성자 오버로딩이 많아질 경우 생성자 간의 중복된 코드가 발생할 수 있다. 매개변수의 수만 달리하고 필드 초기화 내용이 비슷한 생성자에서 이런 현상을 많이 볼 수 있다. 이를 방지하고자 this()를 사용한다.

- 같은 클래스의 디른 생성자 호출(this()의 매개값은 호출되는 생성자의 매개 변수에 맞게 작성해야 한다.)

- 반드시 첫 번째 문장에서만 사용

21. this에 대해 설명하시오

- this는 객체 자신의 참조이다. 우리가 우리 자신을 '나'라고 하듯이 객체가 객체 자신을 this라고 한다.

- 통상적으로 매개변수는 필드와 비슷하거나 동일한 이름을 가짐. 이 경우 필드와 매개변수의 이름이 동일하기 때문에 사용 우선순위가 높은 매개변수에만 접근됨.

- 첫 번째 줄에서만 사용 가능

Ex)

String name;

String ssn;

public Korean(String name, String ssn){

this.name(필드의 name) = name(매개변수의 name};

this.ssn(필드의 ssn) = ssn(매개변수의 ssn)}

22. 생성자는 클래스에 반드시 하나 이상 존재하느냐?

- 넹 우리가 클래스 내부에 생성자 선언을 생략했더라도 컴파일러는 몸체가 비어 있는 기본 생성자를 바이트코드에 추가한다.

23. 정적멤버(정적변수와 정정 메소드)에 대해 설명해봐라

- 선언 시 static 키워드를 추가한다.

- 객체 생성 없이 사용 가능하다.

- 객체마다 가지고 있을 필요가 없는 공용 데이터라면 정적변수로 선언

Ex) 원의 넓이나 둘레를 구할 때 필요한 파이 = Math.PI(클래스명.변수명으로 접근)

- 정적 메소드는 클래스명.메소드(매개값)으로 접근(Math.sqrt(5))

- 정적 메소드에서는 this, 인스턴스 변수와 인스턴스 메소드에 접근할 수 없다(객체 생성을 안하고 사용하기 때문

*접근제한자

private> package > protected > public

<- 접근 제한이 강화

Public = 모든 클래스에서 사용 가능

Protected = 같은 패키지 + 자식 클래스에서 사용 가능

package = 같은 패키지에서 사용가능

private = 클래스 내에서만 사용가능

24. 클래스간의 관계에서 사용관계, 집합관계, 상속관계에 대해 설명해봐라

사용관계: 객체 간의 상호작용(메소드 호출)을 말한다.

집합관계: 한 객체는 부품, 한 객체는 완성품

Ex) 엔진, 타이어, 핸들 객체 <-> 자동차 객체(자동차 객체는 엔진, 타이어, 핸들 등의 객체를 가지고 있다.)

상속관계: 부모 객체 <-> 자식 객체

Ex) 동물 객체 <-> 강아지 객체

24. for-each문과 for문을 비교하라

- for-each문은 배열의 처음부터 끝까지 순차적으로 값을 꺼내서 처리하기 때문에 원소의 값을 변경하거나 역순으로 처리하는 경우, 전체가 아닌 일부 원소만 처리하는 경우 등에는 전통적인 for문을 사용해야한다.

25. 2차원 배열이 메소드로 전달되면 무엇이 전달되는가?

- 행과 열로 전달된다.?

26. 상속에 대해 설명해라

- 이미 잘 개발된 클래스로부터 멤버들을 물려받아 새로운 클래스를 만드는 것

- 단일상속

- 부모 클래스로부터 멤버를 물려받기 때문에(상속된 멤버를 자식 클래스에서 추가, 교체, 상세화 시킬 수 있다.) 코드의 중복을 줄이고 개발과 유지보수가 쉽다.

27. 서브클래스에서 추가된 멤버를 부모 클래스에서 사용 가능한가?

- X 상속은 일방향성이다. 상속 관계에서 부모 클래스를 변경하면 자식 클래스에 영향을 주지만 자식 클래스를 변경하면 부모 클래스에는 영향이 없다.

28. 오버라이딩의 개념에 대해 말해봐라

- 메소드 재정의(자식 클래스에서 부모 클래스의 메소드를 다시 정의하는 것을 말한다)

- 부모의 메소드와 동일한 시그니처(리턴 타입, 메소드 이름, 매개변수 목록)을 가져야한다(메소드

의 몸체만 변경. 헤더는 동일)

- 접근 제한을 더 강하게 재정의할 수 없다. 반대로 더 약하게는 가능

29. 자식 클래스에서 부모 클래스로부터 상속받은 메소드를 중복 정의(오버로딩)할 수 있을까?

- 못한다. 서브 클래스가 작성한 새로운 클래스로 인식한다.

30. super 키워드 사용이유는?

- 부모를 가리키는 키워드

- 자식 클래스에서 부모 클래스 메소드를 재정의하게 되면 부모 클래스의 메소드는 숨겨지고 재정의된 자식 메소드만 사용된다. 그러나 자식 클래스 내부에서 부모 클래스의 메소드를 호출해야 하는 상황이 온다면 super 키워드를 붙여서 부모 메소드를 호출할 수 있다.

31. 부모 생성자에 디폴트 생성자는 없고 매개변수를 가진 생성자가 정의되어 있다. 이때 자식 클래스에서 부모 클래스 생성자를 호출하지 않으면 어떻게 되는가?

- 항상 자식 클래스는 부모 클래스의 생성자를 호출하기 때문에 부모 클래스에 기본 생성자가 아닌 생성자가 정의되어 있다면 자식 클래스에서 부모 클래스 생성자를 명시적으로 호출해줘야 한다.

32. final 클래스, final 메소드, 변수에서 사용되는 final 키워드에 대해 각각 설명해라

- final 키워드는 해당 선언이 최종 상태이고 수정될 수 없음을 뜻한다.

- final class: 상속x

- final 메소드: 재정의X

- final 변수(필드): 값변경X 상수

33. 추상 클래스의 의미 특징과 용도에 대해 말해봐라

- 몸체가 없는(추상) 메소드를 가지고 있는 클래스(상속의 용도, 일반 메소드도 가질 수 있음)

- 추상 메소드가 하나도 없어도 abstract로 선언되어 있으면 추상 클래스이다.

- 추상 클래스를 상속받는 자식 클래스에서는 반드시 추상 메소드를 재정의해야 한다.

- 객체 생성 불가능(new X), 상속을 위한 부모 클래스로만 사용된다.

- 공통된 필드와 메소드의 이름을 통일할 목적으로 사용된다.

(자식 클래스를 여러 명이서 작성하면 데이터와 기능이 동일함에도 이름이 달라 객체마다 사용방법이 달라짐을 방지 ex) turnOn, powerOn)

-

34. 부모 클래스에서 몸체가 없는 메소드를 만든 후 자식 클래스에서 오버라이딩 하는 경우와 추상 메소드의 차이점은?

- 실행결과는 같으나 추상 메소드를 사용하면 자식 클래스에서 무조건 재정의하여야 한다.

35. 인터페이스의 특징은?

- 추상 메소드들로만 이루어져 있다.(자식 클래스에서 무조건 몸체 구현해야함)

- 객체 생성이 불가능하다.

- 다중 인터페이스 구현이 가능하다.

- implements 키워드를 사용해서 구현한다.

36. 인터페이스의 필요성 및 주된 용도는?

- 돼지코

-다형성, 여러 다른 개체에 대해 동일한 작업을 수행하는 능력이라고 말할 수 있습니다.

- 개발 코드를 수정하지 않고, 사용하는 객체를 변경할 수 있도록 하기 위해서이다. 인터페이스는 하나의 객체가 아니라 여러 객체들과 사용이 가능하므로 어떤 객체를 사용하느냐에 따라서 실행 내용과 리턴 값이 다를 수 있다. 즉, 코드 변경 없이 실행 내용과 리턴값을 다양화할 수 있는 장점이 있다.

* 인스페이스 참조 변수를 통해서는 그 인터페이스 안에 정의된 메소드 만을 호출할 수 있다. 다른 메소드나 필드에는 접근할 수 없다.

37. 인터페이스 안에 인스턴스 변수를 선언할 수 있는가?

- 안된다. 상수와 몸체 없는 메소드만 사용 가능하다.

* 상향형변환

자식 클래스는 부모 클래스의 객처처럼 취급되어질 수 있다. 왜냐하면 자식 클래스 객체 안에는 부모 클래스의 객체가 포함되어 있다고 생각할 수 있기 때문이다.

Ex) 동물(부모) 클래스 <-> 강아지(자식) 클래스

강아지는 동물이다O

동물은 강아지다X

Animal a;

Dog d;

a = new Animal; 가능

a = new dog; 가능 단, Animal로부터 상속받은 부분만 사용가능

d = new dog; 가능

d = new Animal; 불가능

*하향형변환

- 자식 클래스인데 형변환에 의해 일시적으로 부모 클래스 참조 변수에 의해 참조되고 있는 경우 하향 형변환을 통해 원래 상태로 되돌릴 수 있다. 이때 반드시 명시적으로 형변환 연산자를 적어주어야 한다.

38. 자식 클래스를 부모 클래스 객체로 취급하는 것이 어디에 사용되는가?

- 메소드의 매개변수를 선언할 때(부모 클래스에서 파생된 모든 클래스의 객체를 받을 수 있다.)

- 자바는 동적 바인딩을 사용하므로 재정의된 메소드로 객체들이 동일한 메시지를 받더라도 각 객체의 타입에 따라 다른 동작을 하게 할 수 있다.

39. 동적바인딩이란?

- 실행 단계에서 객체의 타입을 보고 적절한 메소드를 호출하는 것

40. instanceof 연산자가 하는 연산은 무엇인가?

- 객체의 타입이 맞으면 true 아니면 false를 반환한다.

리눅스

1. user01 사용자를 생성하고 사용자 ID를 2022로 지정하는 명령어는?

답: adduser --uid 2022 user01

2. tar 명령어의 동작 중 c, x 와 옵션 f 에 대해 말해보시오

c: 새로운 묶음 파일을 만든다.

x: 묶음 파일을 푼다.

f: 묶음 파일명을 지정한다.

3. 해당 컴퓨터가 네트워크상에서 응답하는지 테스트하기 위한 네트워크 명령어와 DNS 서버의 작동을 점검하기 위한 명령어를 순서대로 적으시오.

답: ping, nslookup

4. 명령모드에서 입력모드로의 전환 키는?

- i, a, o, s, I, A, O, S

5. 시스템을 종료하는 명령어는?

- poweroff, shutdown –P now, halt –p, init 0 등