

1.jdk는 소프트웨어 개발환경 JVM과 컴파일 디버거 애플릿 및 응용프로그램 개발을위한 도구들이 포함되어있음.

JRE 자바 실행환경 라이브러리,자바 가상기계 등을제공. 개발하지않고 프로그램을 실행만하는 일반인들을위한 환경

2.공개적 정적인 메소드의반환값이 없는 메인메소드 메소드 매개변수 순서.

3. 3,6,7,8

4.byte,short,int,long,float,double,char,boolean

5.19.7 >> 자동형변환 다른 자료형중 큰타입에 따라감.

6. `public static void main(String[] args) {`

```
    int count = 1;
    for(int i = 0 ; i < 10 ; i++) {
        for(int j = 0; j < 10; j++) {
            if(j > count)
                System.out.print("*");
            else
                System.out.print(" ");
        }
        System.out.println();
        count++;
    }
}
```

}

7.`import java.util.*;`  
`public class test1 {`

```
    public static void main(String[] args) {
        System.out.println("숫자를입력: ");
        Scanner ac = new Scanner(System.in);
        int a = ac.nextInt();

        if(a % 2 == 0) {
            System.out.println("짝수");
        } else {
            System.out.println("홀수");
        }
    }
}
```

}

8.

```
public static void main(String[] args) {  
    int a = (int) (Math.random()* 45) + 1;  
    System.out.printf("%d",a);  
}  
}
```

```
9. public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("커피주문 : ");  
    String a = sc.next();  
    int b = 0;  
  
    switch(a) {  
        case "에스프레소":  
        case "카푸치노":  
        case "라떼":  
            b = 3500;  
            break;  
        case "아메리카노":  
            b = 2500;  
            break;  
        default:  
            System.out.println("메뉴에없음");  
    }  
    if(b != 0) {  
        System.out.println(a + "는" + b + "원");  
    }  
}  
}
```

10. 필드, 메소드

11. 특정한종류의 객체들을 찍어내는 형틀.

12. 클래스로부터 만들어지는 각각의 객체

13. 추상화

```
14. class car{  
    String color;  
    int speed;  
    int price;  
    void print() {  
        System.out.println("(" + color + ", " + speed + ", " + price +  
        ")");  
    }  
}
```

```

    }
    public class Car111111 {
    public static void main(String[] args) {
        car mycar = new car();
        mycar.color = "blue";
        mycar.speed = 50;
        mycar.price = 2000;
        mycar.print();
    }
}

```

15.  $11 + 93 + 85 + 134 = 323$

16. 15:30에 재부팅

17. 틀린답 정답x

18. mkdir asd1,asd2,asd3

19. A (shift a)

20. `ls -l` << 각 파일의 모드, 링크수, 소유자, 그룹, 크기, 최종수정시간을 표시  
`ls -a` << 숨겨진 파일이나 디렉터리를 표시

21. `cd /root/wonjin >> ls >> tar -zxvf 2nt5.tar.gz`

22. 메소드나 블록안에서 선언되는 변수

23. 메소드 선언에서의 변수

ex) class

```

    public int a < 필드
    ~~~
    void sibure(int a) << 매개변수
    {
        int c << 지역변수
    }

```

24. 6열에서 생성자를 선언했기 때문에 디폴트 생성자는 만들지 않기 때문에 오류가 발생한다.

25.선언한 첫 번째 생성자를 호출.

26. 인스턴스,인스턴스변수 , 정적변수 ,정적변수

27.자바 가상기계가 객체를 생성할 필요없이 main()메소드를 호출하여 실행할 수있도록 하기위해

28. this가 참조할 인스턴스가 없기 때문이다.

29.오버라이딩 >> 상속받은 조상의 메서드를 자신에게 맞게 변경을 하는 것이다

선언부가 조상 클래스의 메서드와 일치 해야 한다.

접근제어자를 조상 클래스의 메서드보다 좁은 범위로 변경할 수 없다.

예외는 조상 클래스의 메서드보다 많이 선언할 수 없다.

오버로딩 >> 한 클래스 안에 같은 이름의 메서드를 여러개 정의 하는 것을 의미한다.

메서드의 이름이 같아야한다.

매개변수의 개수 또는 타입이 달라야 한다.

반환 타입은 영향이 없다.

30. 사용 (use) : 하나의 클래스가 다른 클래스를 사용.

집합 (has-a) : 하나의 클래스가 다른 클래스를 포함한다.

상속(is-a) : 하나의 클래스가 다른 클래스를 상속한다.

31. 데이터와 절차를 하나의 덩어리로 묶어서 생각하는 방식

32. 관련된 데이터와 알고리즘이 하나의 묶음으로 정리된 것.  
용어그래도 캡슐에 넣어서 포장하는 것을 의미.

특징 > 상속, 다형성, 정보은닉

33.

Class

같은 종류의 객체들이 여러개 필요한 경우 같은 종류의 객체들은 모두 하나의 설계도로 만들어짐 이설계도를 클래스라고한다.

Instance

클래스로부터 만들어지는 각각의 객체를 그 클래스의 인스턴스라고 한다.

34. 기초변수란 int,float,char등의 기초자료형을 가지는 변수이다.

참조변수란 객체를 참조할 때 사용되는 변수이다.(배열,클래스,인터페이스)

35. Car myCar > 참조변수 선언 , new Car(); > 객체생성

36. myCar >> myCar가 참조하는 객체로부터 speed >> speed라는 필드에 접근 //

객체내의 필드는 메소드를 통해서 간접적으로 접근하는 것이 바람직하다.

37. myCar >> myCar가 참조하는 객체로부터 print(); >> print()라는 메소드 호출

38. 프로그램을 작성하다보면 우리자신의 목적에 맞는 클래스를 제작할필요가 있기 때문.

39. 메소드의 매개변수로 배열을 전달한다. getAverage가 가진 배열객체의 참조값이 전달된다.

#### 40. 수퍼클래스 , 서브클래스

41.

```
class Employee{
    public String name; < 공용멤버인 이름
    String address; < 패키지멤버인 주소
    protected int salary; < 보호 멤버인 월급
    private int RRN; < 전용멤버인 주민등록번호

    public String toString() { < toString()함수를 사용하여 매개변수없이 각데이터를
문자열로 변경
        return name + ", " + address + "," + RRN + ", " + salary;
    }
}
class Manager extends Employee { <고용 을 상속받는 매니저에전용멤버인 보너스추가
    private int bonus;

    public void printSalary() { < 이름과 (주소) (급여와 보너스를 더한값)
        System.out.println(name + "(" + address + "):" + (salary + bonus));
    }
    public void printRRN() { < 수퍼클래스에서 private로 정의했으므로
서브클래스에서는 접근할 수 없다.
        System.out.println(RRN);
    }
}
public class ManagerTest {

    public static void main(String[] args) {
        Manager m = new Manager();
        m.printRRN();
    }
}
```

#### 42. 종단클래스는 상속을 시킬수 없는 클래스를 의미한다.

종단메소드는 종단클래스가 아닌 일반 클래스에서 특정한 메소드만 재정의될 수 없게 하기위해 사용하는 것이다.

#### 43.추상메소드

44.추상클래스의 극단적인 경우, 인터페이스는 추상 메소드들로만 이루어진다. 이들 추상메소드는 특정 인터페이스를 따르는 클래스들을 위한 요구 조건의 역할을한다.

45. 상속은 다중상속이 불가능하기 때문에 같은 조상클래스를 가지고있지않은 서로 관계가없는 클래스들에게 하나의 공통적으로 구현하도록 함으로써 관계

를 맺기 때문에 사용.

46. 사용하고있던 인터페이스를 변경하였기에 이인터페이스를 구현했던 모든 클래스가 동작하지 않는다.

47. 가능하다 상향형변환을 사용하여 서브클래스 객체로 대치할수있게 만들어 준다. 역으로는 서브클래스 객체는 수퍼클래스 객체처럼 취급될수있기 때문에 가능하다.

48. true,false

49. 특정 멤버 변수를 외부에서 자주 사용한다고 할때 이것을 효율적으로 구현하기가 상당히 어렵다.하지만 특정 멤버 변수를 public 으로 선언하는 것은 캡슐화의 개념에 어긋난다. 이런 경우에 내부 클래스를 사용하게되면 특정 멤버변수를 private로 유지하면서 자유롭게 사용할 수 있다.

하나의 장소에서만 사용되는 클래스들을 한곳에 모을 수 있다. 만약클래스가 하나의 장소에서만 필요하다면 클래스를 분리하는것보다 클래스의 내부에 위치시키는 것이 가독력이 좋아진다.

보다읽기쉽고 유지보수가 쉬운 코드가 된다 .작은 클래스를 내장시키는 것은 클래스가 사용되는 위치에 더가깝게 두는 것이다.

50. 바인딩 , 동적바인딩