



chapter6

1. StudaCard 클래스 정의

```
class SutdaCard {
    private int num;
    private boolean isKwang;

    public void setNum(int num) {
        if(1 <= num && num >= 10) {
            this.num = num;
        }
    }

    public void setIsKwang(String kwang) {
        if(kwang.equals("광")) {
            isKwang = true;
        } else {
            isKwang = false;
        }
    }
}
```

2. 두개의 생성자와 info를 추가해서 실행결과 얻기

```
public SutdaCars(int num, boolean kwang) {
    this.num = num;
    this.isKwang = kwang
}

public String info() {
    if(isKwang)
        return num+"K";
    else
        return num.toString();
}
```

3. 멤버 변수를 갖는 클래스 정의

```
class Student() {
    String name;
    int ban;
    int no;
    int kor;
    int eng;
    int math;
}
```

4. getTotal과 getAverage 메소드 생성

```
public int getTotal() {
    return kor + eng + math;
}

public float getAverage() {
```

```
return (int)(getTotal() / 3f * 10 + 0.5f) / 10f;
}
```

5. 생성자

```
public Student(String name, int ban, int no, int kor, int eng, int math) {
    this.name = name;
    this.ban = ban;
    this.no = no;
    this.kor = kor;
    this.eng = eng;
    this.math = math;
}
public String info() {
    return name + "," + ban + "," + no + "," + kor + "," + eng + "," + math + "," + getTotal() + "," + getAverage();
}
```

6. 제곱근

```
return Math.sqrt((x-x1)*(x-x1) + (y-y1)*(y-y1));
```

7. 6번을 MyPoint클래스 에서 정의

```
public double getDistance(int x1, int y1) {
    return Math.sqrt((x-x1)*(x-x1) + (y-y1)*(y-y1));
}
```

8. 변수 분류

- 클래스 - width, height
- 인스턴스 - kind, num
- 지역변수 - k, n, card

9. 변수 정의

- weapon, armor
- 공격력과 방어력이 같아야 한다고 지문에서 말하기에 공유되어야 한다.

10. 생성자의 옳지 않은것

- b - 객체를 초기화 하기위한 것 (생성은 new)
- e - 오버로딩이 가능함.

11. this의 설명

- b - static 에서는 사용이 안됨.
- d - ??

12. 오버로딩의 설명

- c - 리턴타입이 같아야함 (이름 같고, 매개변수가 달라야함)
- d - 매개변수의 타입이나 갯수가 달라야함.

13. 오버로딩 한것

- b, c, d
- a - 매개 변수의 타입과 갯수가 같음.

14. 초기화

- c - 초기화 블록이 먼저 실행된다.
- e - 클래스 변수가 먼저 초기화 된다.

15. 초기화 순서

- a - 기본 - 명시 - 초기화 블록 -생성자 순

16. 지역변수에 대한 설명

- a - 초기화가 필요하다.
- e - 힙 영역은 인스턴스가 생성되는 메모리 이며 지역변수는 call stack에 생성.

17. 호출 스택

- b → 아직 종료되지 않음.

18. 컴파일 발생 하는 부분

- A → 인스턴스 변수가 아직 초기화 되지 않음.
- B → static 메소드에서는 인스턴스 변수를 사용할 수 없다.
- D → 인스턴스 메소드를 사용할 수 없다.

19. 실행 결과 예상

- ABC123
- str은 static이 아니기에 change에서 새로운 메모리가 할당된다.

20. 배열의 순서 바꾸는 메소드

```
for(int i = 0; i < arr.length; i++) {  
    int index = (int)(Math.random() * arr.length);  
    int tmp = arr[i];  
    arr[i] = arr[index];  
    arr[index] = tmp;  
}  
return arr;
```

21. Tv클래스 완성

```
isPowerOn = !isPowerOn;

if(volume < MAX_VOLUME) {
    volume++;
}

if(volume > MIN_VOLUME) {
    volume--;
}

channel++;
if(channel > MAX_CHANNEL) {
    channel = MIN_CHANNEL;
}

channel--;
if(channel < MIN_CHANNEL) {
    channel = MAX_CHANNEL;
}
```

22. 메소드 작성

```
public boolean isNumber(String str) {
    Integer.parseInt(str)~~~
    if(str.length() < 1 || str == null) return false;
    for(int i = 0; i < str.length(); i++) {
        if(!('0' <= str.charAt(i) && str.charAt(i) <= '9')) {
            return false;
        }
    }
    return true;
}
```

23. 메소드 작성

```
public int max(int[] arr) {
    if(arr.length == 0 || arr == null) return -999999
    int maxResult = arr[0];
    for(int i = 1; i < arr.length-1; i++) {
        if(maxResult < arr[i]) maxResult = arr[i];
    }
    return maxResult;
}
```

24. 메소드 작성

```
public int abs(int value) {
    if(value < 0) {
        value *= -1;
    }
    return value;
}
```