

자바 저장 빈 클래스 생성 문법

1. 빈 클래스 생성 문법

형식)

```
public class 클래스명{
```

```
    private String name;
```

중략

```
    //빈클래스는 변수명을 정의할 때 반드시 private으로 정의한다. 이유는 내 자신
```

```
    //클래스내에서만 접근 할 수 있기 때문이다. 데이터 보안성
```

```
    public void setName(String name){//setter() 메서드 정의(값을 저장)
```

```
        this.name=name;
```

```
    //좌측 멤버변수와 우측 매개변수명이 같을때는 멤버변수 앞에 this.을 붙여야 값이
```

```
    //저장된다. 만약 this.을 생략하면 값이 저장 안된다.
```

```
    //this는 내 자신 인스턴스 가리키는 참조 변수이다.
```

```
    }
```

```
    public String getName(){//getter() 메서드 정의
```

```
        return name;//return 키워드로 값을 반환
```

```
    }
```

```
}
```

2. 빈클래스명을 사용하는 이유는 중간에 자료를 저장하는 역할을 하기 위해서 이다.

3. 빈클래스는 자바의 기본타입 8개에 포함 안되는 레퍼런스(참조) 타입인 클래스형에

해당한다.

클래스(정적) 메서드와 클래스(정적) 변수

1. static 예약어로 정의된 메서드를 클래스(정적)메서드라 한다. 클래스 메서드는 객체 생성 없이 클래스로 직접 접근 한다. 물론 new키워드로 생성된 모든 객체에 의해서 정적메서드는 공유되기 때문에 생성된 객체로도 접근할 수 있다.

 클래스 메서드 내에 인스턴스 변수와 this를 사용할 수 없다.

2. static 예약어로 정의된 변수를 클래스(정적)변수라 한다. 정적변수도 클래스로 직접 접근 한다.

자바의 메서드 오버로딩 개념

1. 메서드 오버로딩이란 한 클래스 내에 같은 이름의 메서드를 여러 개 정의하는 것을 말한다.
2. 메서드 오버로딩 구분 요건
 - 첫째, 메서드의 전달인자 개수를 다르게 한다.
 - 둘째, 메서드의 전달인자 타입을 다르게 한다.
 - 셋째, 메서드의 전달인자 순서를 다르게 한다.

자바의 생성자 특징

1. 생성자 이름은 클래스 이름과 같다.
2. 생성자는 new 클래스명(); 에 의해서 호출된다.
3. 생성자는 메서드의 일종이다.
4. 생성자의 리턴 타입이 없다.
5. 생성자는 객체가 생성될 때 호출되는 멤버 변수 초기화 메서드이다.
6. 생성자는 오버로딩이 가능하다.
7. 같은 클래스에서 다른 생성자 호출 법
예) this();
8. 매개변수가 없는 생성자를 기본(default) 생성자라 한다. 기본 생성자는 자바 컴파일러가 묵시적으로 제공하기 때문에 코드상에서 생략가능하다. 하지만 생성자가 오버로딩이 된 경우는 묵시적 제공을 하지 않는다. 그러므로 오버로딩이 된 경우는 기본 생성자를 명시적인 코드를 하는 습관을 가지는 경우가 좋다.