



메소드 오버로딩과 String 클래스

메소드 오버로딩과 String 클래스

● 메소드 오버로딩

- 메소드의 이름과 메소드의 매개변수 정보를 참조하여 메소드 호출
- 따라서 메소드의 이름이 같더라도 메소드의 매개변수 정보(자료형, 개수)가 다르다면 문제가 되지 않음
- 반환형은 메소드 오버로딩에 관여하지 않음
- 예시

```
public class MyHome {  
    public void mySimpleRoom(int n) { }  
    public void mySimpleRoom(int n1, int n2) { }  
    public void mySimpleRoom(int n1, int n2, int n3) { }  
}
```

```
public class Solution {  
    public static void main(String[] args) {  
        MyHome home = new MyHome();  
        home.mySimpleRoom(3, 5);  
    }  
}
```

메소드 오버로딩과 String 클래스

- 생성자에 대한 메소드 오버로딩

```
public class Person {  
    private int regiNum;  
    private int passNum;  
    public Person(int regiNum, int passNum) {  
        this.regiNum = regiNum;  
        this.passNum = passNum;  
    }  
    public Person(int regiNum) {  
        this.regiNum = regiNum;  
        this.passNum = 0;  
    }  
    void showPersonallInfo() {  
        System.out.println("주민등록 번호 : " + this.regiNum);  
        if (this.passNum != 0) {  
            System.out.println("여권번호 : " + this.passNum);  
        } else {  
            System.out.println("여권을 가지고 있지 않습니다.");  
        }  
    }  
}
```

메소드 오버로딩과 String 클래스

```
public class ConOverloading {  
    public static void main(String[] args) {  
        Person jung = new Person(335577, 112233);  
  
        Person hong = new Person(775544);  
  
        jung.showPersonalInfo();  
        hong.showPersonalInfo();  
    }  
}
```

- Person 클래스에서 파라미터가 다른 생성자 2개
- Solution 클래스에서 Person의 객체를 생성할 때 2가지 방법으로 생성 가능
- 반환형은 메소드 오버로딩에 관여하지 않는다.

메소드 오버로딩과 String 클래스

● String 클래스

```
public class StringInst {  
    public static void main(String[] args) {  
        String str1 = new String("The Best String");  
        String str2 = "The Best String";  
        String str3 = "The Best String";  
        System.out.println(str1);  
        System.out.println(str1.length());  
        System.out.println(str2);  
        System.out.println(str2.length());  
        if (str1 == str2) {  
            System.out.println("str1과 str2는 같은 곳을 참조합니다.");  
        }  
        if (str2 == str3) {  
            System.out.println("str2와 str3는 같은 곳을 참조합니다.");  
        }  
    }  
}
```

- String 인스턴스 생성은 2가지 방법으로 가능
- new 키워드를 사용하여 생성하는 방법과 =를 사용하여 생성하는 방법은 참조의 차이 존재
- 문자열의 비교는 equals 메소드 이용