

CellPhone 클래스

-변수

String model // 핸드폰 모델 번호

double battery; // 남은 배터리 양

-생성자

CellPhone(String model) // 모델 번호를 인자로 받는 생성자

-메서드1

void call(int time) // 통화 시간(분)을 출력하고, 통화 시간에 따라 배터리 양을 감소시킨다.

// 선언부에 IllegalArgumentException 전달 표현을 추가한다

// 감소되는 배터리 양 = 통화시간(분) * 0.5

// 배터리 양은 0보다 작아지지 않는다.

// 통화 시간(분)이 0보다 작은 경우에는

// 배터리 양은 변화없고

// IllegalArgumentException("통화시간입력오류") 발생시킨다

- 메서드2

void charge(int time) // 충전한 시간(분)을 출력하고, 충전한 시간에 따라 배터리 양을 증가시킨다.

// 선언부에 IllegalArgumentException 전달 표현을 추가한다

// 충전되는 배터리 양 = 충전시간(분) * 3

// 배터리 양은 100까지만 증가한다.

// 충전 시간(분)이 0보다 작은 경우에는

```
// 배터리 양은 변화없고
```

```
// IllegalArgumentException("충전시간입력오류") 발생시킨다
```

- 메서드3

```
void printBattery()      // 남은 배터리 양을 출력한다.
```

- 메서드4

```
boolean isSame(CellPhone other)
```

```
// CellPhone 타입의 객체를 입력받고, 입력받은 객체의
```

```
// 모델 번호가 같은 경우에 true를 리턴한다.
```

*<<CellPhoneMain 실행결과>>

충전 시간 : 20분

남은 배터리 양 : 60.0

통화 시간 : 300분

남은 배터리 양 : 0.0

충전 시간 : 50분

남은 배터리 양 : 100.0

통화 시간 : 40분

남은 배터리 양 : 80.0

통화시간입력오류

동일 모델입니다.

*/

```
public class CellPhoneMain {

    public static void main(String[] args) {

        CellPhone myPhone = new CellPhone("GALAXY-7");

        //charge, call 메소드 호출 포함하여 try-catch 블록 정의
        // throws 하고 있는 예외 처리한다.
        // 각 예외 발생시 어떤 메소드에서 어떤 원인으로 발생했는지
        // 출력한다

        myPhone.charge( 20 ); //20분간 충전을 한다.
        myPhone.printBattery();

        myPhone.call( 300 ); //300분간 통화를 한다.
        myPhone.printBattery();

        myPhone.charge( 50 ); //50분간 충전을 한다.
        myPhone.printBattery();

        myPhone.call( 40 ); //40분간 통화를 한다.
        myPhone.printBattery();

        myPhone.call( -20 ); //통화시간입력오류

        CellPhone yourPhone = new CellPhone("galaxy-7");
```

```
        if( myPhone.isSame(yourPhone) ) {  
            System.out.println("동일 모델입니다.");  
        } else {  
            System.out.println("다른 모델입니다.");  
        }  
    }  
}
```

```
public class CellPhone {  
  
}
```