Object

Object

Object

Object Class

- 자바에서 상속은 필수
- 아무것도 상속하지 않더라도 암묵적으로 Object클래스를 상속
- 결국 모든 클래스는 Object클래스의 후손클래스

```
public class Test{

}
```

- 아무 내용 없이 클래스를 만든 후 객체를 생성해보면, 작성한 적 없는 메소드들이 있는 것이 확인 가능
- 작성한 적 없는 메소드들은 모두 Object 클래스의 메소드로 기본상속
- 클래스가 공통적으로 포함하고 있어야 하는 기능을 포함

Object

toString() 메소드

- 내용이 없는 클래스를 생성
```
public class TestObject{
}
```

- 실행메소드
```
TestObject to = new TestObject();
System.out.println(to);
System.out.println(to.toString());
```

두개의 출력 결과가 동일
레퍼런스 주소 호출 시 자동으로 toString메소드 호출

출력결과

패키지명.클래스명@인스턴스식별값

※ toString()메소드는 overriding하여 사용 가능

Object

clone() 메소드

- 객체 자체를 복사하여 다른 객체에 사용할 때 사용하는 메소드
  → 배열의 깊은복사 시 clone()을 이용하여 복사

- clone()메소드의 접근제어 지시자는 protected이므로 다른 패키지에서는 사용 불가

- 작성한 클래스에서 clone() 메소드를 사용하기 위해서는 overriding을 해야함

- Overriding을 위해서는 Colneable interface를 상속해야함
  → 상속이유는 복제 가능한 Class임을 JVM에게 알려주기 위함

Object

equals() 메소드

- 객체가 같은지 비교하는 메소드

```
String str1 = new String(“Hello”);
String str2 = new String(“Hello”);
if(str1 == str2){
    System.out.println(“같습니다.”);
}else{
    System.out.println(“다릅니다.”);
}
```

→ String객체의 값은 동일하지만 두 객체는 다른인스턴스이므로 “다릅니다” 출력

같은지 비교하기 위해서

```
if(str1.equals(str)){
    System.out.println(“같습니다.”);
}else{
    System.out.println(“다릅니다.”);
}
```

Object

clone() 메소드

```
public classTestClass implements Cloneable{
    private int data;
    //기본생성자, 매개변수생성자, getter, setter 작성
    @Override
    public Object clone(){
        try{
            //부모클래스의 clone메소드를 호출하여 리턴함
            //clone메소드의 실제 코드는 볼 수 없음
            return super.clone();
        }catch(CloneNotSupportedException e){
            e.printStackTrace();
        }
        return null;
    }
}
```

try ~ catch 는 뒤에서 따로 다룸