

10

- 다양한 입출력 클래스

최 문 환



1. 객체 직렬화

객체의 직렬화는 데이터들이 한 줄로 나열해서 스트림을 통해서 전송된다는 말입니다.

지금까지 살펴본 입출력 방식과 같이 데이터들이 개별적으로 전송되는 것이 아니고 클래스 내부에 설계된 멤버들이 객체 단위로 파일에 기록하거나 쓴다는 의미입니다.

2. ObjectOutputStream과 ObjectInputStream

<예제> ObjectOutputStream을 이용한 객체단위로 출력

```
001:import java.io.*;
002:import java.util.*;
003:class ObjectOutputStreamTest01{
004:    public static void main(String [] args) throws IOException {
005:        String name = new String("성윤정");
006:        Date birthDay = new Date();
007:
008:        FileOutputStream fos = new FileOutputStream("test02.txt");
009:        ObjectOutputStream oos = new ObjectOutputStream(fos);
010:        oos.writeObject(name);
011:        oos.writeObject(birthDay);
012:        oos.flush();
013:        oos.close();
014:    }
015:}
```

<예제> ObjectInputStream을 이용한 객체 단위로 입력

```
001:import java.io.*;
002:import java.util.*;
003:class ObjectInputStreamTest02{
004:    public static void main(String [] args) throws Exception {
005:        String name=null;
006:        Date birthDay = new Date();
007:        FileInputStream fis = new FileInputStream("test02.txt");
008:        ObjectInputStream ois = new ObjectInputStream(fis);
009:        name = (String)ois.readObject();
010:
011:        birthDay = (Date)ois.readObject();
012:
013:        System.out.println(name);
014:        System.out.println( birthDay.toString() );
015:        ois.close();
016:    }
017:}
```

<예제> ObjectOutputStream을 이용한 객체 단위로 출력

```
001:import java.io.*;
002:class ObjectOutputStreamTest03{
003:    public static void main(String[] args) {
004:        int ID=1;
005:        String name="성윤정";
006:        int age=25;
007:        double height=165.6;
008:        try{
009:            FileOutputStream fos=new FileOutputStream("iotest2.txt") ;
010:            ObjectOutputStream oos=new ObjectOutputStream(fos);
011:
012:            oos.writeObject(new Integer(ID));
013:            oos.writeObject(name);
014:            oos.writeObject(new Integer(age));
015:            oos.writeObject(new Double(height));
016:
017:            oos.close();
018:            fos.close();
019:        }catch(IOException e){
020:            e.printStackTrace();
021:        }
022:    }
023:}
```

<예제> ObjectInputStream을 이용한 객체 단위로 입력

```
001:import java.io.*;
002:class ObjectStreamTest04{
003: public static void main(String[] args) {
004:     int ID;
005:     String name="";
006:     int age;
007:     double height;
008:     try{
009:         FileInputStream fis=new FileInputStream("iotest2.txt") ;
010:         ObjectInputStream ois = new ObjectInputStream(fis);
011:         ID =(Integer)ois.readObject();
012:         name=(String)ois.readObject();
013:         age=(Integer)ois.readObject();
014:         height=(Double)ois.readObject();
015:
016:         System.out.println("ID Wt name Wt age Wt height");
017:         System.out.println(ID + "Wt " + name + "Wt " + age + "Wt " + height);
```

<예제> ObjectInputStream을 이용한 객체 단위로 입력

```
019:    ois.close();
020:    fis.close();
021: }catch(ClassNotFoundException ce){
022:     ce.printStackTrace();
023: }
024:     catch(IOException ie){
025:         ie.printStackTrace();
026:     }
027: }
028: }
```

<예제> 객체 직렬화를 위한 클래스 설계

```
001:import java.io.*;
002:class Customer implements Externalizable{
003:    int ID;
004:    String name ;
005:    int age;
006:    double height;
007:
008:    public Customer( ) { }
009:
010:    public Customer(int ID, String name, int age, double height){
011:        this.ID      = ID;
012:        this.name     = name;
013:        this.age      = age;
014:        this.height   = height;
015:    }
016:    public void writeExternal(ObjectOutput oos) throws IOException{
017:        oos.writeObject(new Integer(ID));
018:        oos.writeObject(name);
019:        oos.writeObject(new Integer(age));
020:        oos.writeObject(new Double(height));
021:    }
022:}
```


<예제> 객체 직렬화를 위한 클래스 설계

```
023:    public void readExternal(ObjectInput ois)
        throws ClassNotFoundException, IOException{
024:        ID =(Integer)ois.readObject();
025:        name=(String)ois.readObject();
026:        age=(Integer)ois.readObject();
027:        height=(Double)ois.readObject();
028:    }
029:
030:    public String  toString(){
031:        String temp;
032:        temp = ID + "Wt " + name + "Wt " + age + "Wt " + height;
033:        return temp;
034:    }
035:}
```

<예제> ObjectOutputStream을 이용한 객체 단위로 출력

```
001:import java.io.*;
002:class ObjectOutputStreamTest05{
003:    public static void main(String[] args) {
004:        Customer cus=new Customer(2, "전수빈", 9, 130);
005:        try{
006:            System.out.println(cus);
007:            FileOutputStream fos=new FileOutputStream("iotest3.txt") ;
008:            ObjectOutputStream oos=new ObjectOutputStream(fos);
009:            oos.writeObject(cus);
010:            oos.close();
011:            fos.close();
012:        }catch(IOException e){
013:            e.printStackTrace();
014:        }
015:    }
016:}
```

<예제> ObjectInputStream을 이용한 객체 단위로 입력

```
001:import java.io.*;
002:class ObjectStreamTest06{
003:  public static void main(String[] args) {
004:    try{
005:      FileInputStream fis=new FileInputStream("iotest3.txt") ;
006:      ObjectInputStream ois=new ObjectInputStream(fis);
007:      Customer cus=(Customer)ois.readObject( );
008:      System.out.println("ID Wt name Wt age Wt height");
009:      System.out.println(cus);
010:      ois.close();
011:      fis.close();
012:    }catch(ClassNotFoundException ce){
013:      ce.printStackTrace();
014:    }
015:    catch(IOException ie){
016:      ie.printStackTrace();
017:    }
018:  }
019:}
```