

2024년 상반기 K-디지털 트레이닝

데이터 입출력

[KB] IT's Your Life



☑ Todo 클래스가 직렬화(Serialization)가 가능한 객체로 만드세요.

Todo.java

```
package org.scoula.todo.domain;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.io.Serializable;
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class Todo implements Serializable {
    Integer id;
    String title;
    String desc;
    boolean done;
```

◎ TodoDao 클래스에 load(), save() 메서드를 추가하세요.

- o load()
 - todo.dat 파일이 존재하는 경우, 저장된 내용을 읽어 data 멤버 초기화함.
 - 파일이 존재하지 않는 경우, 테스트 데이터로 data 멤버 초기화함
 - 생성자에서 테스트 데이터 설정하지 않음
- o save()
 - 현재 data 멤버를 todo.dat 파일로 저장함

☑ TodoDao.java

```
package org.scoula.todo.dao;
...
import java.io.*;

public class TodoDao implements CrudDao<Integer, Todo> {
    HashMap<Integer, Todo> data;
    int maxId;

    private static TodoDao instance = new TodoDao();

    private TodoDao() {
    }
}
```

TodoDao.java

```
public void load() {
   File file = new File("./todo.dat");
   if(file.exists()) { //파일이 존재하는 경우
       try(ObjectInputStream oi = new ObjectInputStream(new FileInputStream(file))) {
          maxId = oi.readInt();
          data = (HashMap<Integer, Todo>) oi.readObject();
       } catch (Exception e) {
          System.err.println("파일 읽기 에러 : " + e.getMessage());
       };
   } else { // 파일이 없는 경우
       data = new HashMap<>();
       data.put(1, new Todo(1, "ES6 학습", "ES6 심화 과정 실습하기", false));
       data.put(2, new Todo(2, "React 학습", "React 기초 공부하기", false));
       data.put(3, new Todo(3, "ContextAPI 학습", "컨텍스트 훅 공부하기", false));
       data.put(4, new Todo(4, "야구 경기 관람", "NC대 두산 야구 관람하기", true));
       data.put(5, new Todo(5, "Java 공부", "Java 컬렉션 공부하기", false));
       maxId = 6;
```

☑ TodoDao.java

```
public void save() {
    File file = new File("./todo.dat");
   try(ObjectOutputStream oo = new ObjectOutputStream(new FileOutputStream(file))) {
       oo.writeInt(maxId);
       oo.writeObject(data);
       oo.flush();
   } catch (Exception e) {
       System.err.println("파일 쓰기 에러 : " + e.getMessage());
   };
```

☑ TodoApplication이 기동할 때 TodoDao의 load()를 호출하고, 종료할 때 TodoDao의 save() 메서드를 호출하도록 TodoApplication을 수정하세요.

☑ TodoApplcation.java

```
package org.scoula.todo;
import org.scoula.todo.dao.TodoDao;
public class TodoApplication extends Application {
    @Override
    public void init() {
        super.init();
        TodoDao.getInstance().load();
    @Override
    public void cleanup() {
        super.cleanup();
       TodoDao.getInstance().save();
```