

2024년 상반기 K-디지털 트레이닝

컬렉션 자료구조

[KB] IT's Your Life



- TodoDao 클래스의 다음의 멤버 변수를 추가하고, 생성자를 수정하세요.
 - o HashMap 타입의 data 멤버 변수 추가
 - 키타입: Integer
 - 값 타입: Todo
 - o int maxld
 - 새로운 Todo를 추가할 때 배정할 id 값

☑ TodoDao.java

```
package org.scoula.todo.dao;
import org.scoula.lib.dao.CrudDao;
import org.scoula.todo.domain.Todo;
import java.util.HashMap;
import java.util.List;
public class TodoDao implements CrudDao<Integer, Todo> {
    HashMap<Integer, Todo> data;
    private static TodoDao instance = new TodoDao();
    private TodoDao() {
        data = new HashMap<>();
```

♥ 생성자에서 다음과 같은 내용을 가지는 테스트 데이터를 data 멤버 변수에 추가하세요.

id	title	desc	done
1	ES6 학습	ES6 심화 과정 실습하기	false
2	React 학습	React 기초 공부하기	false
3	ContextAPI 학습	컨텍스트 훅 공부하기	false
4	야구 경기 관람	NC대 두산 야구 관람하기	true
5	Java 공부	Java 컬렉션 공부하기	false

♥ 생성자에서 maxld 멤버 변수를 6으로 초기화하세요.

TodoDao.java

```
package org.scoula.todo.dao;
import java.util.HashMap;
public class TodoDao implements CrudDao<Integer, Todo> {
   HashMap<Integer, Todo> data;
   int maxId;
   private TodoDao() {
       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;
```

☑ TodoDao의 메서드를 실제로 동작하도록 정의하세요.

☑ TodoDao.java

```
package org.scoula.todo.dao;
import org.scoula.lib.dao.CrudDao;
import org.scoula.todo.domain.Todo;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
public class TodoDao implements CrudDao<Integer, Todo> {
   @Override
    public List<Todo> list() {
        List<Todo> todoList = new ArrayList<>(data.values());
        return todoList;
```

☑ TodoDao.java

```
@Override
public Todo get(Integer key) {
    return data.get(key);
@Override
public void create(Todo todo) {
   todo.setId(maxId);
   maxId++;
   data.put(todo.getId(), todo);
@Override
public void update(Integer key, Todo todo) {
   data.put(key, todo);
```

✓ TodoDao.java

```
@Override
public void delete(Integer key) {
    data.remove(key);
}
```

♥ 앞에서 만든 TodoDao를 이용해서 다음 Command에서 Todo CRUD 기능을 완성하세요.

- o 패키지: org.scoula.todo.command
- o TodoListCommand
- o TodoDetailCommand
- o TodoCreateCommand
- o TodoUpdateCommand
- o TodoDeleteCommand
- o 각 Command 객체는 TodoDao 멤버 변수를 가짐

TodoListCommand.java

```
package org.scoula.todo.command;
import org.scoula.lib.cli.command.Command;
import org.scoula.todo.dao.TodoDao;
import org.scoula.todo.domain.Todo;
import java.util.List;
public class TodoListCommand implements Command {
   TodoDao dao = TodoDao.getInstance();
   @Override
   public void execute() {
      System.out.println("[[Todo 목록 보기]]");
      List<Todo> list = dao.list();
      System.out.println("========");
      System.out.println("ID 제목");
      System.out.println("=========");
      for(Todo todo: list) {
         System.out.printf("%3d %s\n", todo.getId(), todo.getTitle());
      System.out.println("========");
```

☑ TodoDetailCommand.java

```
package org.scoula.todo.command;
import org.scoula.lib.cli.Input;
import org.scoula.lib.cli.command.Command;
import org.scoula.todo.dao.TodoDao;
import org.scoula.todo.domain.Todo;
public class TodoDetailCommand implements Command {
   TodoDao dao = TodoDao.getInstance();
   @Override
    public void execute() {
       System.out.println("[[Todo 상세 보기]]");
       int id = Input.readInt("ID: ");
       Todo todo = dao.get(id);
       if(todo == null) {
           System.out.println("해당 항목이 없습니다.");
           return ;
```

☑ TodoDetailCommand.java

☑ TodoCreateCommand.java

```
package org.scoula.todo.command;
                                                                 @Override
                                                                 public void execute() {
                                                                     System.out.println("[[Todo 생성]]");
import org.scoula.lib.cli.Input;
import org.scoula.lib.cli.command.Command;
                                                                     String title = Input.read("TITLE: ");
import org.scoula.lib.dao.CrudDao;
                                                                     String desc = Input.read("DESCRIPTION: ");
import org.scoula.todo.dao.TodoDao;
import org.scoula.todo.domain.Todo;
                                                                     Todo todo = Todo.builder()
                                                                             .title(title)
import java.util.List;
                                                                             .desc(desc)
                                                                             .build();
public class TodoCreateCommand implements Command {
                                                                     dao.create(todo);
    TodoDao dao = TodoDao.getInstance();
                                                                     System.out.println("해당 항목을 생성했습니다.");
```

☑ TodoUpdateCommand.java

```
package org.scoula.todo.command;
                                                                    String title = Input.read("TITLE", todo.getTitle());
import org.scoula.lib.cli.Input;
                                                                    String desc = Input.read("DESCRIPTION",
import org.scoula.lib.cli.command.Command;
                                                                                             todo.getDesc());
                                                                     boolean done = Input.confirm("DONE", todo.isDone());
import org.scoula.todo.dao.TodoDao;
import org.scoula.todo.domain.Todo;
                                                                    todo.setTitle(title);
public class TodoUpdateCommand implements Command {
                                                                    todo.setDesc(desc);
    TodoDao dao = TodoDao.getInstance();
                                                                    todo.setDone(done);
   @Override
                                                                    dao.update(id, todo);
    public void execute() {
                                                                    System.out.println("해당 항목을 수정했습니다.");
       System.out.println("[[Todo 수정]]");
       int id = Input.readInt("ID: ");
       Todo todo = dao.get(id);
       if(todo == null) {
           System.out.println("해당 항목이 없습니다.");
           return ;
```

TodoDeleteCommand.java

```
package org.scoula.todo.command;
import org.scoula.lib.cli.Input;
import org.scoula.lib.cli.command.Command;
import org.scoula.todo.dao.TodoDao;
import org.scoula.todo.domain.Todo;
public class TodoDeleteCommand implements Command {
   TodoDao dao = TodoDao.getInstance();
   @Override
   public void execute() {
       System.out.println("[[Todo 삭제]]");
       int id = Input.readInt("ID: ");
       Todo todo = dao.get(id);
       if(todo == null) {
           System.out.println("해당 항목이 없습니다.");
           return ;
       dao.delete(id);
       System.out.println("해당 항목을 삭제했습니다.");
```

- ☑ TodoApplication을 작성하고 운영하세요.
 - o 패키지: org.scoula.todo
 - o main 메서드를 정의하고 운영

TodoApplication.java

```
package org.scoula.todo;
import org.scoula.todo.command.*;
public class TodoApplication extends Application {
   @Override
    public void createMenu(Menu menu) {
           menu.add(new MenuItem("목록", new TodoListCommand()));
           menu.add(new MenuItem("보기", new TodoDetailCommand()));
           menu.add(new MenuItem("생성", new TodoCreateCommand()));
           menu.add(new MenuItem("수정", new TodoUpdateCommand()));
           menu.add(new MenuItem("삭제", new TodoDeleteCommand()));
           menu.add(new MenuItem("종료", new ExitCommand(this)));
    public static void main(String[] args) {
       TodoApplication app = new TodoApplication();
       app.run();
```