

SC2002 第一次讨论

pufanyi

目录

1 Overview	3
1.1 Techniques We Use	3
1.2 本次讨论目的	3
2 Codes	3
2.1 Location	3
2.2 Model	3
2.2.1 <code>getID()</code>	4
2.2.2 <code>toMap()</code>	4
2.2.3 <code>fromMap(Map<String, String> map)</code>	4
2.2.4 <code>Supervisor(Map<String, String> map)</code>	4
2.3 Repository	4
2.3.1 <code>getFilePath()</code>	5
2.3.2 <code>seeAll(List<Map<String, String>> data)</code>	5
2.3.3 <code>FacultyRepository()</code>	5
2.3.4 <code>FacultyRepository getInstance()</code>	6
2.3.5 <code>getByID</code>	6
2.3.6 <code>contains</code>	6
2.3.7 <code>add</code>	6
2.3.8 <code>remove</code>	6
2.3.9 <code>isEmpty</code>	6
2.3.10 <code>size</code>	6
2.3.11 <code>update</code>	6
2.3.12 <code>updateAll</code>	7
2.3.13 <code>load</code>	7
2.3.14 <code>save</code>	7
2.3.15 <code>findByRules</code>	7
2.3.16 遍历 Repository	7
2.3.17 调用 Repository	8
2.4 JUnit	8

1 Overview

1.1 Techniques We Use

- GitHub
- JUnit 5.8.1
- GitHub Copilot & ChatGPT
- JetBrains IntelliJ IDEA

1.2 本次讨论目的

- 介绍一下我们的 project 的目的
- 介绍一下我们的 project 的大致架构
- 进行 Model 和 Repository 部分的分类与分工
- 大致理解 Controller 部分，完整读完题目

2 Codes

2.1 Location

/src/main/util/config/Location.java

每个人都需要修改一下

代码可以复制 /src/main/util/config/Location.java.example

具体的，需要把其中的 LOCATION 改成自己整个 project 的位置。

```
package main.utils.config;

public class Location {
    public static final String LOCATION = "D:\\NTU\\Y1S2\\SC2002\\FYPMS";
}
```

2.2 Model

/src/main/model/Model.java

具体详细的例子可以参考 /src/main/model/user/Supervisor.java

需要实现的 method:

- @Override public String getID()
- @Override public Map<String, String> toMap()
- @Override public void fromMap(Map<String, String> map)
- public Supervisor(Map<String, String> map)

2.2.1 getID()

返回的是 Model 的 ID，用于在 Repository 中进行查找。

对于同一类型的 Model，ID 必须是唯一的。

2.2.2 toMap()

返回的是 Model 的 Map，用于在 Repository 中进行存储。

Map 类似于 python 中的 dict，可以通过 key 来获取 value。

其中我们把 Model 的 attribute 的 name 作为 key，attribute 的 value 作为 value。

```
@Override
public Map<String, String> toMap() {
    Map<String, String> ansMap = new HashMap<>();
    ansMap.put("supervisorID", this.supervisorID);
    ansMap.put("supervisorName", this.supervisorName);
    ansMap.put("email", this.email);
    ansMap.put("password", getPassword());
    return ansMap;
}
```

2.2.3 fromMap(Map<String, String> map)

从 Map 中读取 Model 的 attribute。

```
@Override
public void fromMap(Map<String, String> map) {
    this.supervisorID = map.get("supervisorID");
    this.supervisorName = map.get("supervisorName");
    this.email = map.get("email");
    setPassword(map.get("password"));
}
```

2.2.4 Supervisor(Map<String, String> map)

通过 Map 来构造 Model。

2.3 Repository

具体详细的例子可以参考 /src/main/repository/user/FacultyRepository.java

Repository<ModelType> 中的 ModelType 是要存储的 Model 的类型，必须继承于 Model 类型。

```
import main.repository.Repository;
```

```
public class FacultyRepository extends Repository<Supervisor> {  
    ...  
}
```

需要实现的 method:

- @Override public String getFilePath()
- @Override public void seeAll(List<Map<String, String>> data)
- public FacultyRepository()
- public static FacultyRepository getInstance()

其他方法不需要再实现，直接用即可。

2.3.1 getFilePath()

返回的是存储仓库的绝对路径。

```
final private static String FILE_PATH = "/data/user/faculty.txt";  
  
@Override  
public String getFilePath() {  
    return Location.LOCATION + FILE_PATH;  
}
```

2.3.2 seeAll(List<Map<String, String>> data)

给你的的是一个 List<Map<String, String>>, 你需要把它转换成你需要的类型。
具体的转法见 Model.toMap(Map<String, String>)。

```
@Override  
public void setAll(List<Map<String, String>> listOfMappableObjects) {  
    for (Map<String, String> map : listOfMappableObjects) {  
        getAll().add(new Supervisor(map));  
    }  
}
```

2.3.3 FacultyRepository()

仿照以下方法写就好了。

```
public FacultyRepository() {  
    super();  
    load();  
}
```

2.3.4 FacultyRepository getInstance()

仿照以下方法写就好了。

```
public static FacultyRepository getInstance() {  
    return new FacultyRepository();  
}
```

2.3.5 getByID

```
public ModelObject getByID(String modelObjectID) throws NoSuchElementException
```

2.3.6 contains

```
public boolean contains(String modelObjectID)
```

2.3.7 add

```
public void add(String modelObjectID) throws IllegalArgumentException
```

抛出 `IllegalArgumentException` 的原因是因为 `modelObject` 的 ID 已经存在了。

2.3.8 remove

```
public void remove(String modelObjectID) throws NoSuchElementException
```

抛出 `NoSuchElementException` 的原因是因为 `modelObject` 的 ID 不存在。

2.3.9 isEmpty

```
public boolean isEmpty()
```

2.3.10 size

```
public int size()
```

2.3.11 update

```
public void update(ModelObject) throws NoSuchElementException
```

将原来 ID 为 `modelObject` 的 `Model` 替换成 `modelObject`。

抛出 `NoSuchElementException` 的原因是因为 `modelObject` 的 ID 不存在。

2.3.12 updateAll

```
public void updateAll(List<ModelObject> modelObjects)
```

2.3.13 load

```
public void load()
```

从仓库中重新加载数据。

2.3.14 save

```
public void save()
```

将数据保存到仓库中。

需要注意的是，凡是已完成的方法，如果数据库中的数据发生了变化，都已经自动调用 `save()` 方法。

2.3.15 findByRules

不给出定义了，只说一下怎么用：

```
facultyRepository.findByRules(  
    supervisor -> supervisor.getID().equals("A1234567A")  
);  
facultyRepository.findByRules(  
    supervisor -> supervisor.getID().equals("A1234567A"),  
    supervisor -> supervisor.getID().equals("12345")  
);
```

传入参数是 `modelObject -> expression` 或者 `modelObject -> { statements; }`，返回值是 `List<ModelObject>`。

`expression` 是一个 `boolean` 类型的表达式，`statements` 是一系列的语句，最终需要 `return` 一个 `boolean` 类型的值，类似于 `method`。

返回的东西均满足 `expression` 或者 `statements`。

如果传入多个参数，那么返回的东西均满足所有的 `expression` 或者 `statements`。

2.3.16 遍历 Repository

```
for (Supervisor supervisor : facultyRepository) {  
    ...  
}
```

2.3.17 调用 Repository

使用 `getInstance()` 方法来获取 `Repository` 的实例。

该实例已经被初始化和数据库同步好了，可以直接使用。

```
FacultyRepository facultyRepository = FacultyRepository.getInstance();
```

例如，在代码中要找到所有叫 `Li Yi` 的 `Supervisor`，可以这样写：

```
List<Supervisor> manyLiYi = FacultyRepository.getInstance().findByRules(
    supervisor -> supervisor.getName().equals("Li Yi");
));
for (Supervisor supervisor : manyLiYi) {
    System.out.println(supervisor.getID());
}
```

2.4 JUnit

重点：一切交给 GitHub Copilot / ChatGPT!!!

例子看 `test` 目录下的代码。