

重构程序文档

PM模块

项目组：Team3

成员：1352837 倪雨婷

1352866 罗晓丹

1352873 王刚

1352895 叶坤宇

1352918 刘旭东

一.PM模块重构组建的选取

我们选用了team2的PM构件。原因是其程序框架与我们组相似且构件功能丰富。

该PM构件提供以下接口：

1. First, you should do the instantiation.

```
/**
 * instantiation
 * @param _filePath
 * @param _delayMinute
 */
PMMManager pmManager = new PMMManager(String _filePath,int
_delayMinute);
```

2. Second,start the Manager

```
public void startRecord( )
```

3. Third, change the amount of the record.

```
/**
 * Log in successfully
 */
public void LogSuccess()
```

```
/**
 * Log in failed
 */
public void LogFail()
```

4. Besides,we can change the filepath.

```
/**
 * reset the filepath
 * param filepath
 */
public static void setFilepath(String filepath)
```

二.PM模块重构的感受

1. team2的PM构件功能丰富，除了提供PM基本功能，还提供了setFilepath函数。接口简洁清晰，便于使用。
2. team2的PM构件将记录登录成功次数与记录登录失败次数封装为两个函数，而我们组的PM构件并没有将此操作接口化，而是仍采用在实体化后直接调用其中登录成功次数参数和登录失败次数参数的方式，不便于用户的使用，不利于安全性，是需要我们改进的地方。
3. 使用他人构件的时候需要根据构件提供接口更改自己程序的部分逻辑，这是这次作业比较繁琐的地方，但由于我们选择的构件简洁且便于使用，我们的代码也更加简短清晰。

三.PM模块重构过程中的问题

1. 在使用该构件后，发现每分钟生成的性能文档仅记录该分钟内登录成功与失败次数，而不是全过程中的总次数。在经过沟通后，我们与team2共同完善了此功能。
2. 我们组的应用程序界面上有显示登录成功及失败次数的模块，在使用我们自己构件时，我们通过直接调用参数来实现此功能。在更改为team2的构件后，发现team2将参数successTime / failTime封装在了构件里，导致我们的此功能无法得到实现。因此，提供getter是此构件可改进的一个部分。在经过沟通后，我们与team2共同完善了此功能。
3. 在使用他人构件过程中，我们不断与team2组员进行沟通，不但完善了自己的程序，也提升了该构件的完成度，同时也明白了自己构件的优缺点。