许可证 License

1.引言

1.1 编写目的

该组件为软件复用课程的课程项目，编写此文档的目的在于使得用户能够更好的复用该组件。

2 组件简介

2.1 功能

该构件可让用户设置系统证书资源的上限，重新设置证书资源上限之后，构件使用数将重新开始计数。注：设置资源上限时，上限值只能是正数（大于 0 的正整数），若设置 0 或负数，那么将设置失败。

2.2 输入

应用程序的性能指标

2.3 输出

根据 license 中设置的的规则进行输出。

2.4 使用方法

首先我们需要获得 LicenseManager 实例，代码如下：

```
LicenseManager instance = new LicenseManager();
```

客户端需要将自身的一些信息填入 AuthManager 类中（），使得系统了解证书的申请者信息。

```
AuthManager instance = new AuthManager();
```

通过 LicenseManager 向系统发起证书申请，代码如下

```
LicenseStatus status = AuthManager.getInstance().checkLicenseStatus
```

另外我们还可以对系统证书资源的上限值做一些改变，调用方法如下：

```
public void setLicenseStore(LicenseStore store){
    if(this.store == null) {
        this.store = store;
        store.initLicense(100);
        NotifyService.startPuller();
    }
}
```

2.5 说明

LicenseManager 中的 requestManager 方法要求客户端传入一个 CallerMessage 类的实例，该实例中包含了发出证书申请客户端的一些信息。随后向系统发出证书申请，该方法会返回一个 RequestResultMessage 对象，该对象中包含了证书申请成功与否的信息。如上，可调用 LicenseStatus 来判断此次证书申请过程是否成功。若返回为 true，则表示此次证书申请成功，否则表示申请失败。

2.6 示例

```
public class LicenseTest {
    public void applyForLicenseTest(){
        LicenseManager.getInstance().setEnableWork(true);
        License license = AuthManager.getInstance().applyForLicense(1000,
TimeUnit.MILLISECONDS);
        SimpleProcess simpleProcess = new SimpleProcess(license);
        Processed processed =
LicenseManager.getInstance().decorateWithLicense(simpleProcess);
        for(int i = 0; i < 5; i++){
            try{
                processed.process();
                Thread.currentThread().sleep(1000);
            }catch (InvalidLicenseException ie){
                ie.printStackTrace();
            }catch (InterruptedException iie){
                iie.printStackTrace();
            }
        }
    }
}
```

以上代码可以完成对 license 的测试，使得其可以按照客户的需求进行修改。