# 个人报告

# 朱海童 3130100660

## 工作内容：

1. 整个工程架构的管理、.pro文件依赖关系的处理
2. ViewModel层Commander的设计和实现
3. Model层Network部分
4. Model层Music、OnMusic、OffMusic类继承关系的设计和基类Music类的实现

VIewModel层：

我负责的是软件的ViewModel部分和Model中的Network部分。一开始觉得要实现的功能都比较简单，各个模块直之间也比较独立，所以采用了MVP模式设计。随着项目的高强度展开，我发现View层和Model层出现了直接进行交互的现象，而且各个模块之间的耦合程度有增加的趋势，意识到设计的模式上存在缺陷。几次向老师请教后，才明白了MVVM模式的概念以及具体应用到项目上的方法。

ViewModel层的具体设计已经在报告的“内部实现”中详细阐释，这里就不加赘述了。下面我想谈谈个人对面向接口编程和传统面向对象编程的一些体会。

## 继承与接口的区别：

对象的继承表现的是一种特殊化的过程。每个子类都是更特殊的父类，所以在具有父类性质的基础上，进一步发展出自己特殊的性质，在程序中的体现就是每个子类实际上都包含了一个父类对象的数据，然后通过增加成员方法或者多态重载父类方法，实现代码的重用和扩展。然而正是因为这种继承关系，使得子类和父类之间存在一种依赖，即父类功能的改变会影响到子类的功能，而子类一些功能需求又不得不在父类中进行修改，这会给程序的实际开发过程，尤其是多人合作开发带来不少麻烦。

而接口代表的是一组规则，每个实现了接口的对象都具有这个规则。接口只是用来给调用者提供一个具体实现了某种规则的对象，从其实现方式——抽象类（C++）即可看出，它和对象之间并没有太多的联系。这种模式对开发的好处是，各个接口相互独立，实现接口的对象之间也相互独立。功能增删只需要通过某个接口/对象的增加/删除即可实现，并不需要影响其他类。于是多人合作的情况下，每个人都可以独立开发自己的模块，然后进行单元测试，互不影响地同步开发，极大地提高了项目开发的秩序性和效率。

Model层的Music类和Network部分也是我负责的，开发完ViewModel层后再回头看这两个类，发现Music类下面OffMusic和OnMusic两个子类确实实现了代码重用，然而在Network类中一些搜索相关的Search类的关系显得比较混乱，继承带来的代码重用效果并不好，反而使得程序结构较为混乱。

有了这次开发的经验，希望以后在对象继承和接口之间能够做出更好的设计选择。

Network部分的实现、测试及思考

NetWork部分主要实现的是歌曲名称、歌手名称mp3链接、专辑图像链接、lrc链接等相关信息的抓取和分析。

一个Search类用来表示整个网络搜索的类，下面继承的MusicSearch和LrcSearch类分别实现在线歌曲搜索和歌词搜索的功能。

向网易云音乐提交http请求的api链接、post data、header等信息，由子类各自实现的MakePostData函数实现，其中调用了父类的SetApiFor…函数。搜索的执行统一由 Dosearch函数完成，期间具体实现：

用QNetworkManager类提交QNetworkRequest类的请求，调用QLoop执行循环，知道收到服务器返回的QReply才推出循环，表示请求已经结束并获得结果，再讲接收到的Json数据存到QJsonObject类型的searchObj成员变量中，留给不同的子类进行对应的分析。MusicSearch和LrcSearch中的GetMusicList和SearchLrc就进行了相应分析。（lrc的分析由于需要根据歌曲id再调用一次API进行搜索，所以需要在现有结果上再进行一次搜索分析。

此外因为需要对抓取下来的图片链接进行处理，所以还设置了一个ImageDownload类来提供处理图片的功能。

关于测试：

为MusicSearch和LrcSearch类都实现了测试模块，进行单元测试。对提交的request和返回的结果（code和json数据）进行检验，看是否符合预期，并对分析出来的歌曲和歌词结果与测试模块内的结果进行了比对，确保了类功能的正确性。

本项目使用了QT框架，所以免不了对QT库进行较多调用。在这个过程中，发现QT的Network库存在不少缺陷。比如当用post发送http请求时需要的cookies头信息，需要QVariant的参数，然后QNetworkCookie类的对象却无法转换为QVariant类型，经过若干尝试我只能放弃cookie相关的部分，所以最后登录的部分没能够实现。

这次调试各种库的艰辛过程让我体会到了能够自己编写小型库函数能力的重要性，虽然现在方便的API很多，但关键的时候需要的功能还是要自己实现的。

附：ViewModel和Network部分的UML图

## ViewModel



## Network

