Java课程设计——固定资产管理系统

姓名： 李润之 班级： 软件1412学号：201492036成绩：

姓名： 刘丰恺 班级： 软件1412学号：201492167成绩：

姓名： 班级： 学号： 成绩：

姓名： 班级： 学号： 成绩：

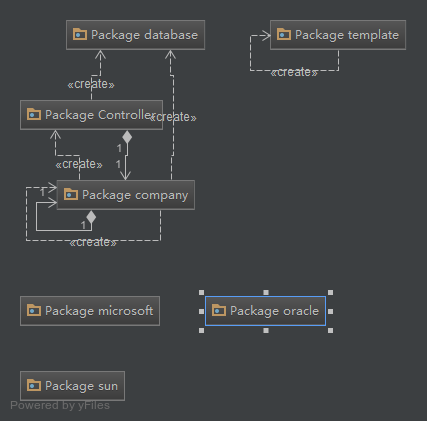
姓名： 班级： 学号： 成绩：

大连理工大学

Dalian University of Technology

1. **系统主要的类图**

1 – 1 Package



(图1 – 1 Package间关系)

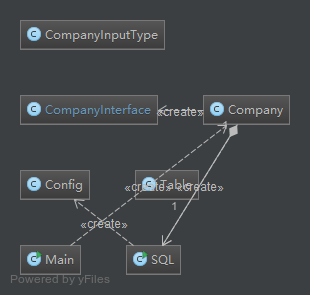
（1）：Package Company – 主调用包，包括SQL操作与View层均位于此包。

（2）：Package Controller – 控制层，包括所有的业务操作。控制、分发数据，调用数据库接口。

（3）：Package Database – 数据库模块层。将Asset数据库中所有表进行抽象封装于此包，方便进行SQL操作。

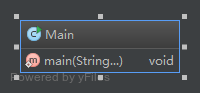
（4）：Package Template – 泛型模板包。包括自己实现的一些泛型接口的封装。

1 – 2 Package Company



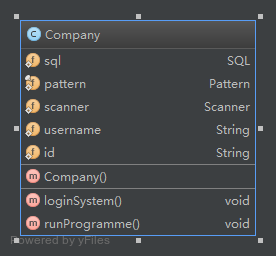
（图1 – 2 Package Company类关系）

（1）：Main – 主函数，负责启动程序



（图 1 – 3 Class Main）

（2）：Company – 主调用函数，负责管理登录操作和选项管理。



（图 1 – 4 Class Company）

1）：sql – SQL对象，操作数据库用

2）：pattern – Pattern对象，编译正则表达式，检测日期格式化

3）：scanner – Scanner对象，接收控制台输入

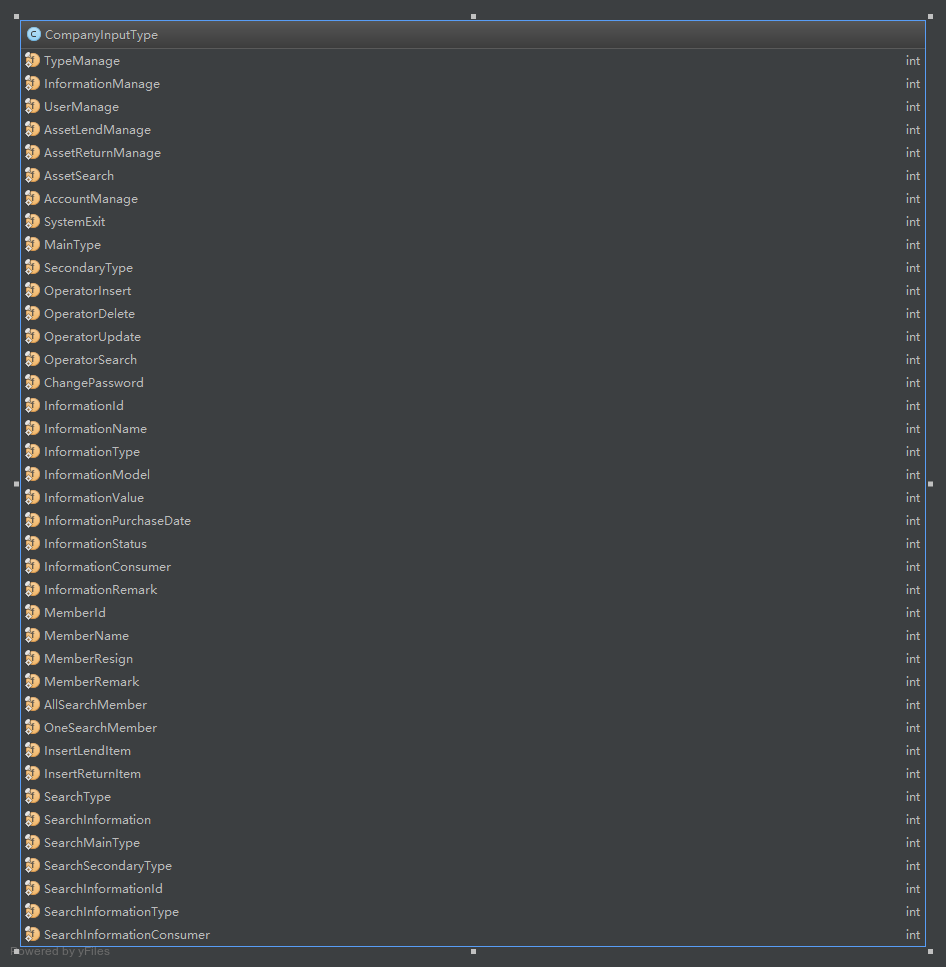
4）：username – 管理员username

5）：id – 管理员id

6）：loginSystem() – 登录系统

7）：runProgramme() – 启动程序

（3）：CompanyInputType – 输入操作定义类，内部定义大量常量，指示输入操作。



（图 1 – 5 Class CompanyInputType）

（4）：Config – 设置类，包括数据库设置

1）：name – 数据库名字

2）：driver – 驱动

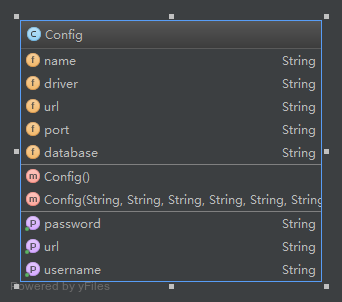
3）：url – 数据库链接

4）：port – 端口

5）database – 数据库名

6）password – 密码

7）username – 用户名



（图 1 – 6 Class Config）

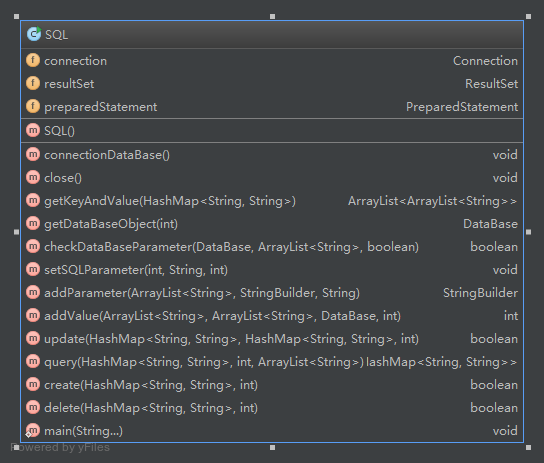
（5）：SQL – 封装的数据库操作，己实现的一个轻型ORM框架，与Package Database进行合作

1）：update – 数据库项目修改

2）：query – 数据库搜索操作

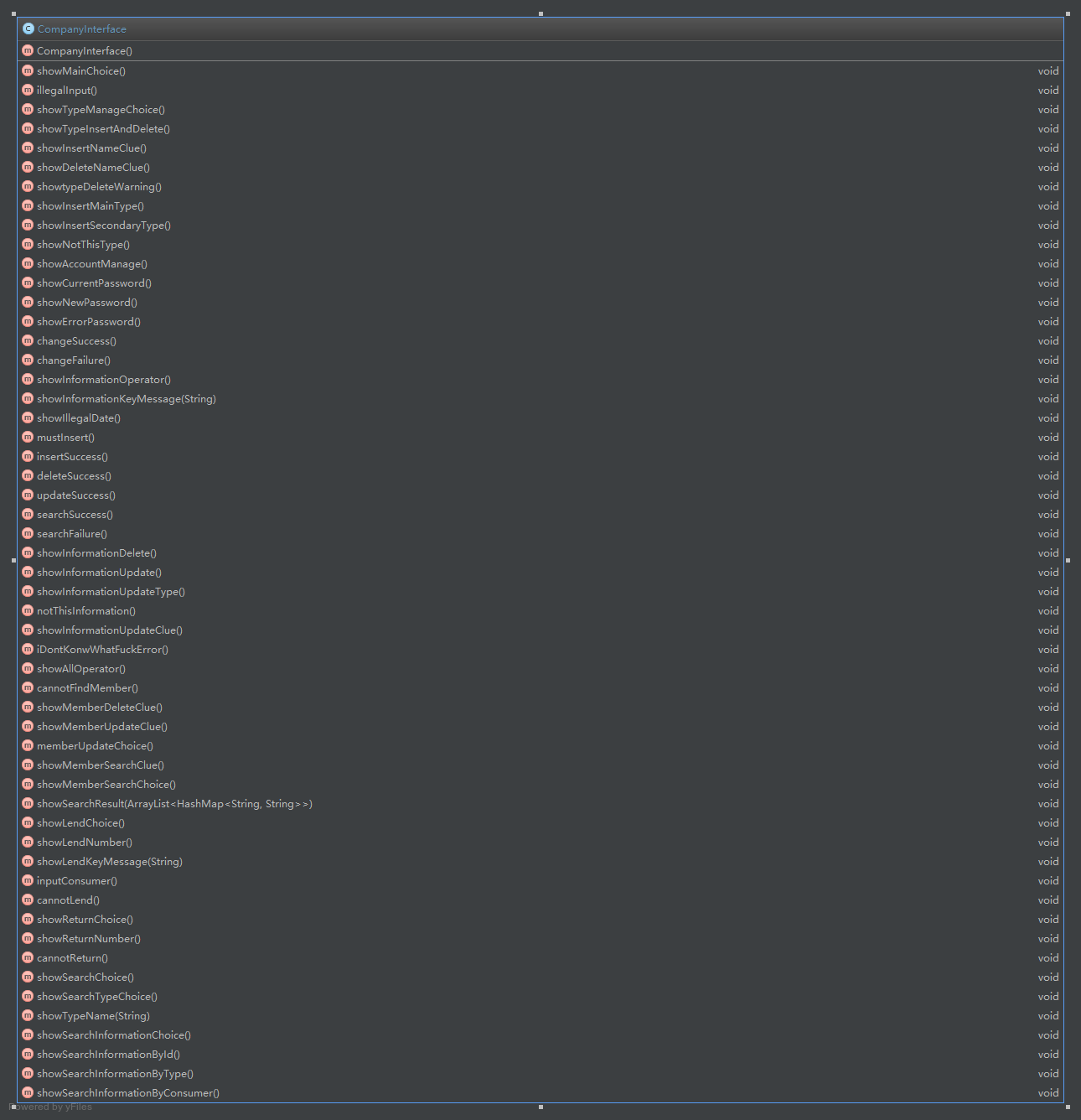
3）：create – 数据库增加操作

4）：delete – 数据删除操作



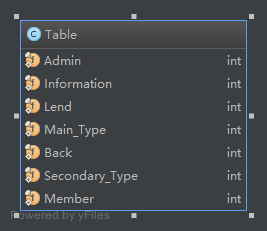
（图 1 – 7 Class SQL）

（6）：CompanyInterface – View层，管理所有控制台输出信息



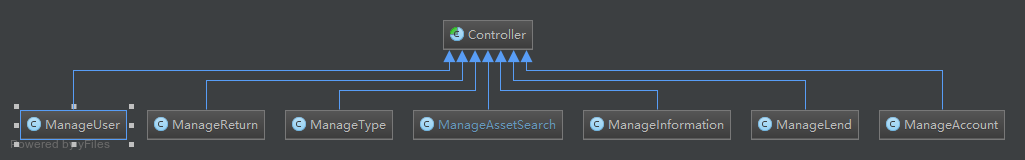
（图 1 – 8 Class CompanyInterface）

（7）：Table类 – 指示数据库表类型，ORM框架一部分，与Class SQL，与Package Database合作工作



（图 1 – 9 Class Table）

1 – 2 Package Controller



（图 1 – 10 Package Controller）

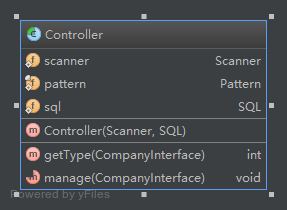
（1）：Controller类 – Controller基类，封装公共方法与虚接口

1）：scanner – Scanner对象

2）：pattern – Pattern对象，编译正则表达式，用来检测日期合法格式

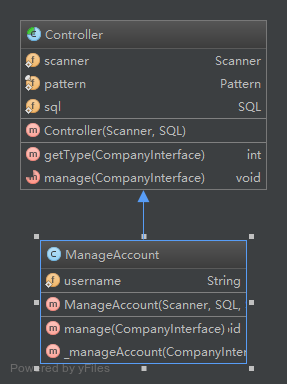
3）：sql – SQL对象

4）：manage – controller启动对象



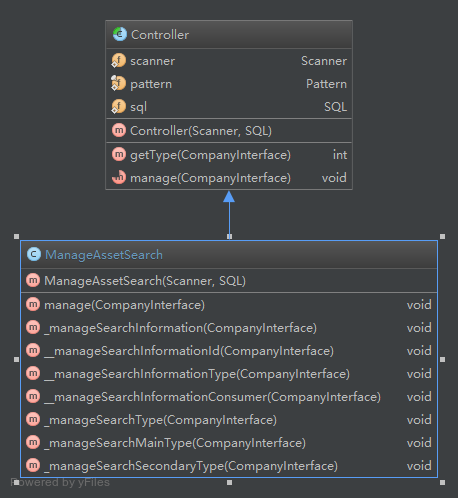
（图 1 – 11 Class Controller）

（2）：ManageAccount类 – 继承自Controller，负责管理管理员帐号方向操作



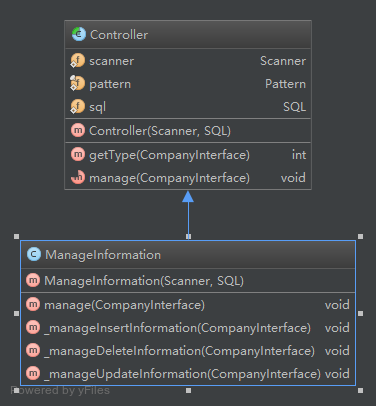
（图 1 – 12 Class ManageAccount）

（3）：ManageAssetSearch类 – 继承自Controller，负责管理信息查找方向操作



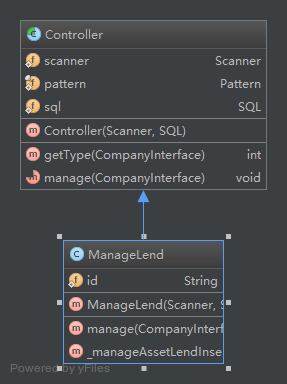
（图 1 – 13 Class ManageAssetSearch）

（4）：ManageInformation类 - 继承自Controller，负责固定资产信息管理操作



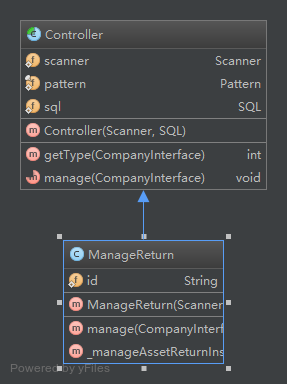
（图 1 – 14 Class ManageInformation）

（5）：ManageLend类 - 继承自Controller，负责资产的领用操作



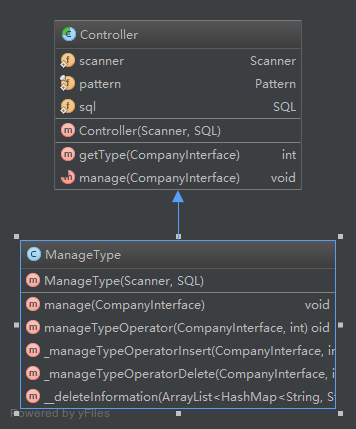
（图 1 – 15 Class ManageLend）

（6）：ManageReturn类 - 继承自Controller，负责资产的归还操作



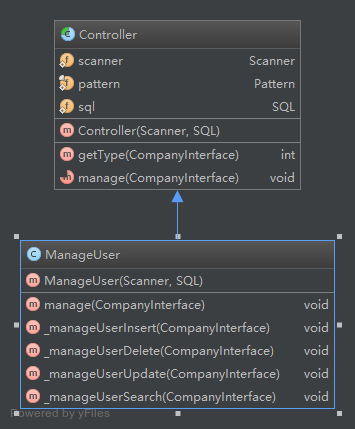
（图 1 – 16 Class ManageReturn）

（7）：ManageType类 - 继承自Controller，负责固定资产类别管理操作



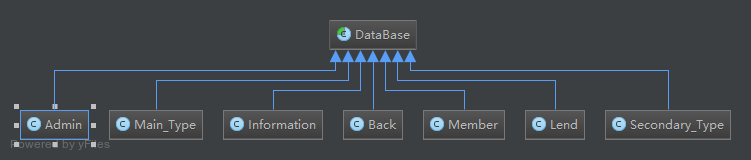
（图 1 – 17 Class ManageType）

（8）：ManageUser类 - 继承自Controller，负责人员信息管理操作



（图 1 – 18 Class ManageUser）

1 – 3 Package Database

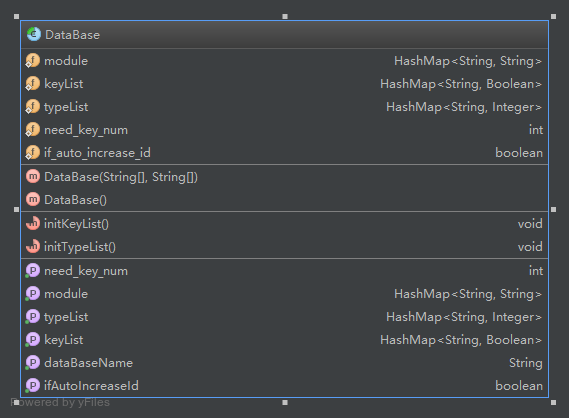


（图 1 – 19 Package DataBase）

（1）：DataBase类 – 数据库表模块基类

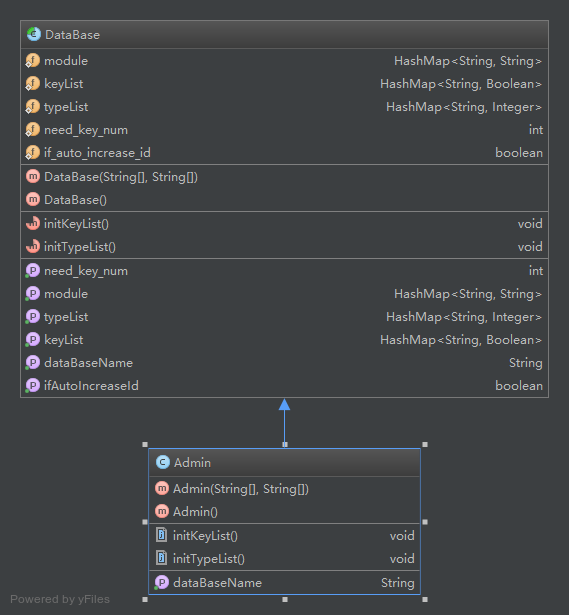
1）：keyList – 指示数据库列是否为必须项目

2）：typeList – 指示数据库列数据类型



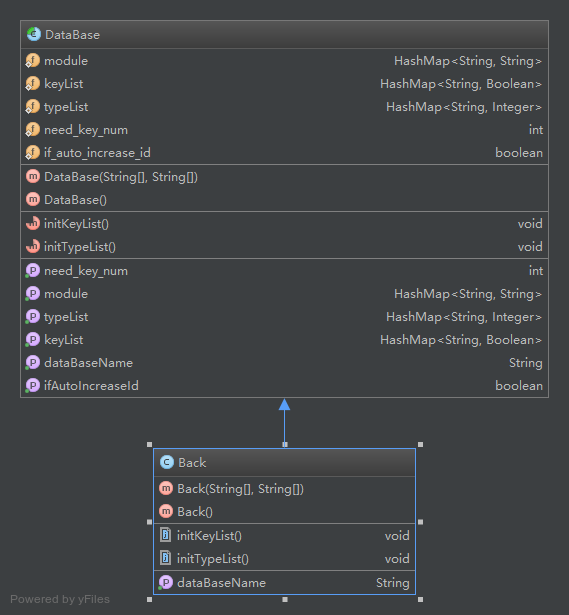
（图 1 – 20 Package DataBase）

（2）：Admin类 – 继承自DataBase类，数据库Admin表抽象模型



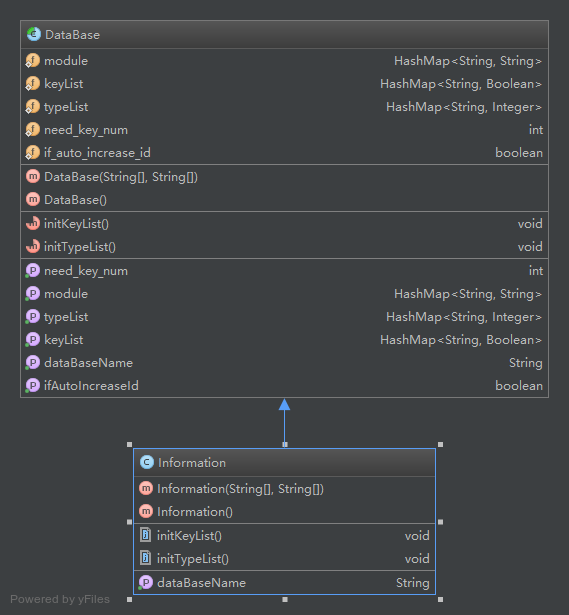
（图 1 – 21 Class Admin）

（3）：Back类 - 继承自DataBase类，数据库Back表抽象模型



（图 1 – 22 Class Back）

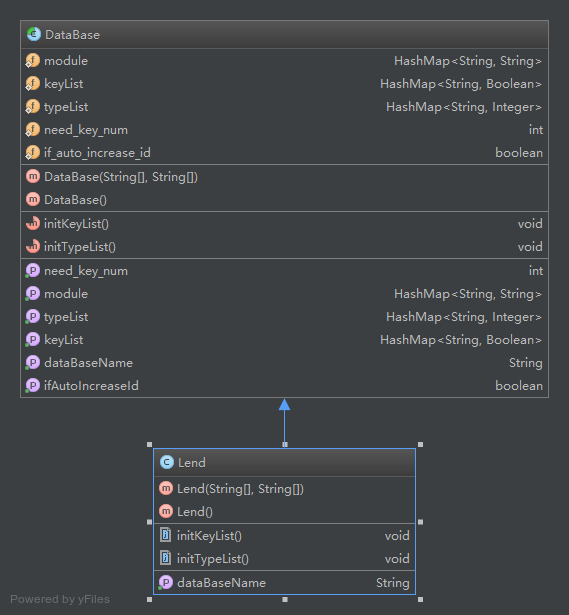
（4）：Information类 - 继承自DataBase类，数据库Information表抽象模型



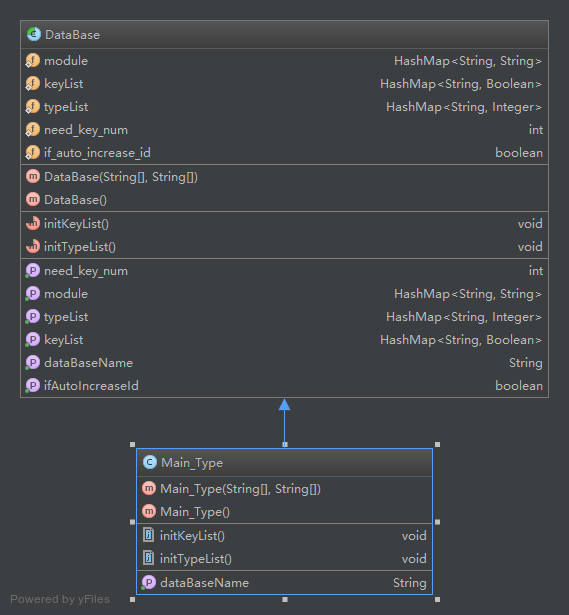
（图 1 – 23 Class Information）

（5）：Lend类 - 继承自DataBase类，数据库Lend表抽象模型

（图 1 – 24 Class Lend）

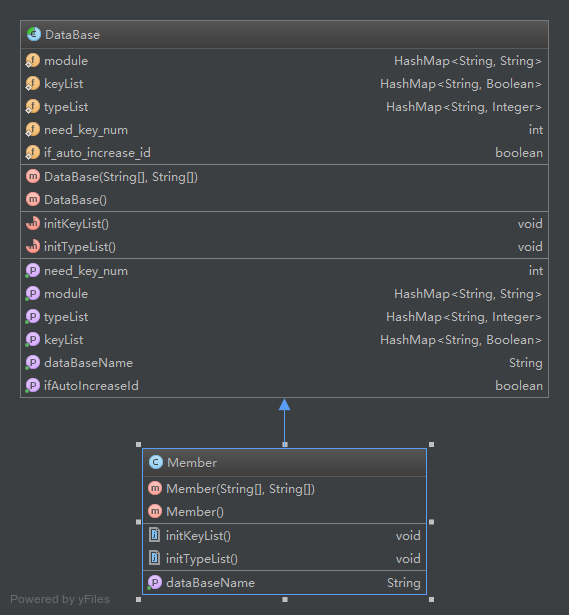


（6）：Main\_Type类 - 继承自DataBase类，数据库Main\_Type表抽象模型



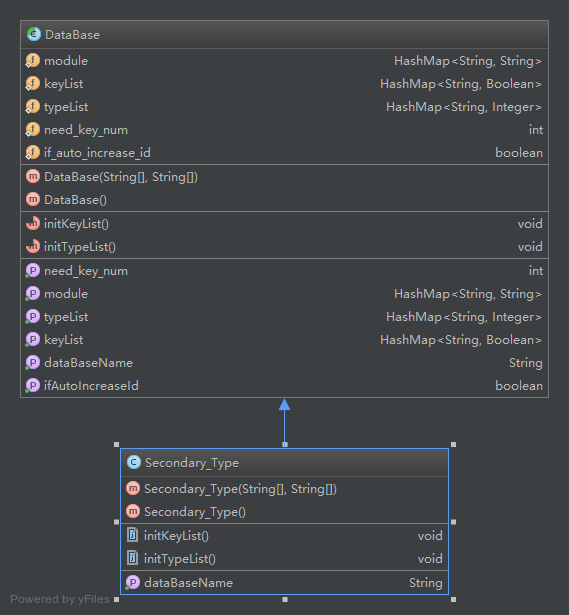
（图 1 – 25 Class Main\_Type）

（7）：Member类 - 继承自DataBase类，数据库Member表抽象模型



（图 1 – 25 Class Member）

（8）：Secondary\_Type类 - 继承自DataBase类，数据库Secondary\_Type表抽象模型

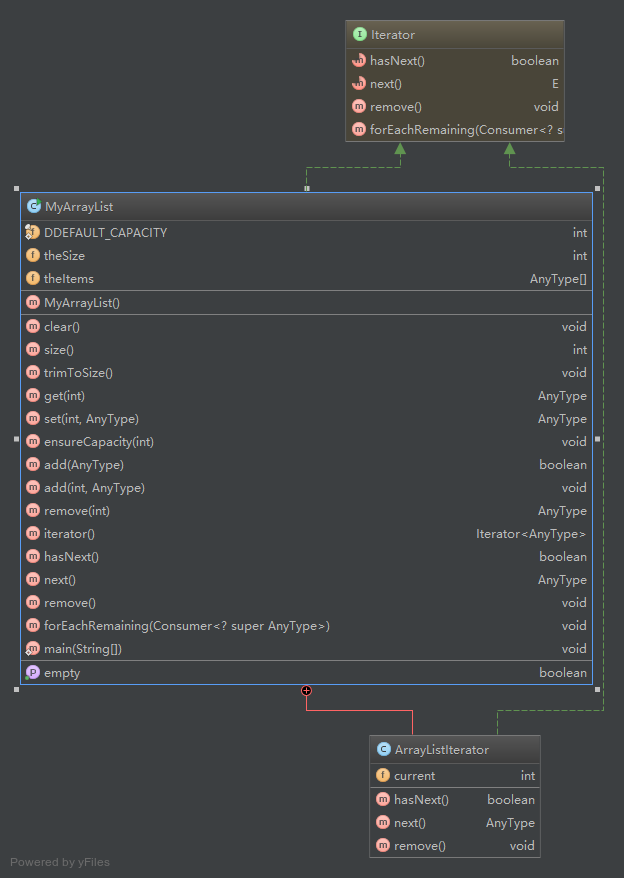


（图 1 – 25 Class Secondary\_Type）

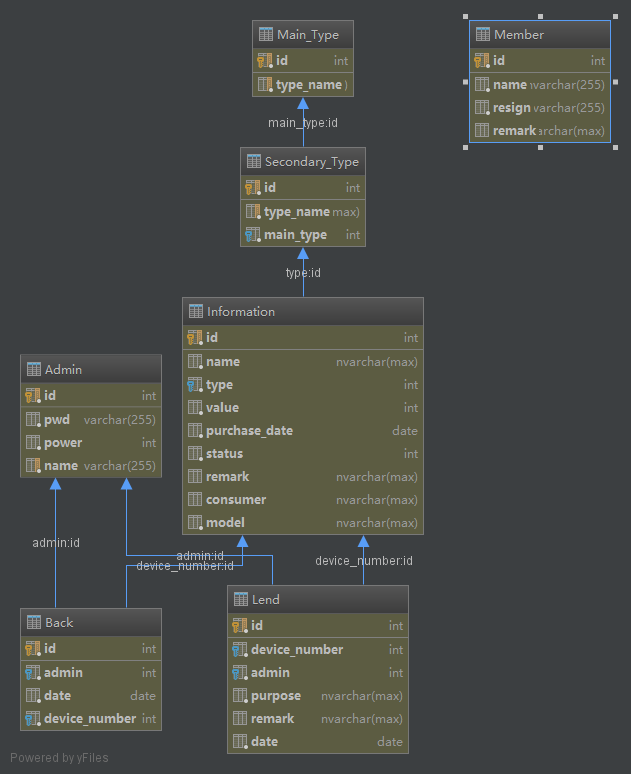
1 – 3 Package Templete

1. MyArrayList类 – 自己实现的ArrayList类

（图 1 – 26 Class MyArrayList）



1. **系统数据库设计**



（图 2 – 1 数据库表关系图）

1. Admin表 （管理员信息）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | Admin id |
| name | varchar(255) | 否 | 无 | 用户类型 |
| pwd | varchar(255) | 否 | 无 | 用户密码 |
| power | int | 否 | 无 | 管理员权力 |

1. Back表 （归还记录）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| admin | int | 否 | Admin.id | 操作管理员id |
| date | date | 否 | 无 | 操作日期 |
| device\_number | int | 否 | Information.id | 设备项目id |

1. Information表 （设备记录）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| name | nvarchar(max) | 否 | 无 | 设备名id |
| purchase\_date | date | 否 | 无 | 操作日期 |
| type | int | 否 | Secondary\_Type.id | 设备小类id |
| value | int | 否 | 无 | 设备价值 |
| model | nvarchar(max) | 否 | 无 | 设备型号 |
| consumer | nvarchar(max) | 否 | 无 | 使用者 |
| status | int | 否 | 无 | 状态 |
| remark | nvarchar(max) | 否 | 无 | 备注 |

1. Lend表 （领用记录）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| admin | int | 否 | Admin.id | 操作管理员id |
| date | date | 否 | 无 | 操作日期 |
| device\_number | int | 否 | Information.id | 设备项目id |
| purpose | nvarchar(max) | 否 | 无 | 用途 |
| remark | nvarchar(max) | 否 | 无 | 备注 |

1. Main\_Type表 （大类）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| type\_name | nvarchar(max) | 否 | 无 | 大类名 |

1. Secondary\_Type表 （小类）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| type\_name | nvarchar(max) | 否 | 无 | 小类名 |
| main\_type | int | 否 | Main\_Type.id | 从属大类id |

1. Member表 （用户）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 关联外键 | 意义 |
| id | int | 是 | 无 | item id |
| name | nvarchar(max) | 否 | 无 | 名字 |
| resign | nvarchar(max) | 否 | 无 | 职务 |
| remark | nvarchar(max) | 否 | 无 | 备注 |

**3、泛型集合的使用**

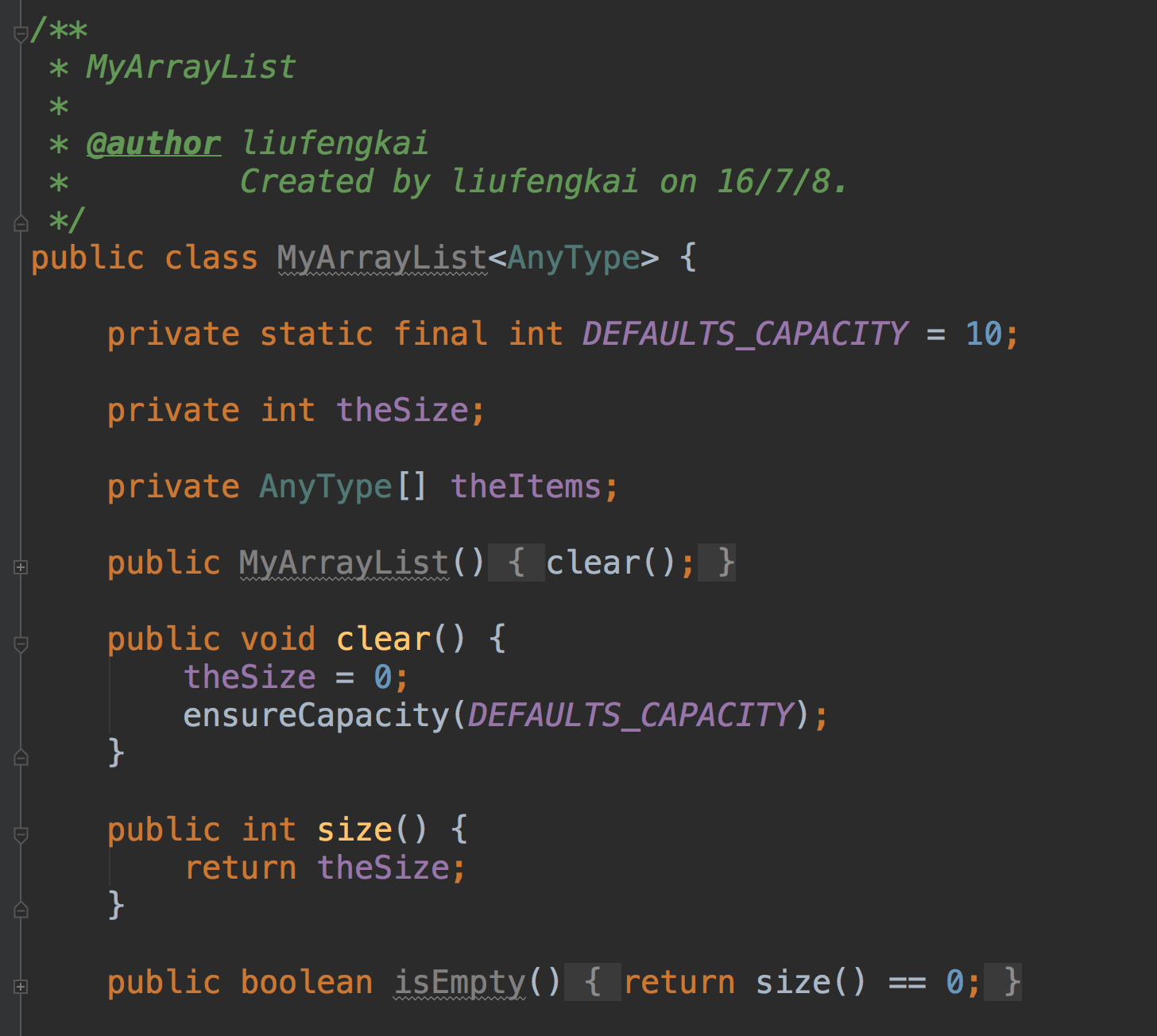
­­­­­­在我们的固定资产管理系统中中我们主要的是用了ArrayList和HashMap两种泛型结构。针对这两种泛型结构，我们进行了仔细的学习和研究，认真阅读了JDK6.0版本这部分的源码，以便于更方便的去理解集合类型和泛型结构。

1. ArrayList：

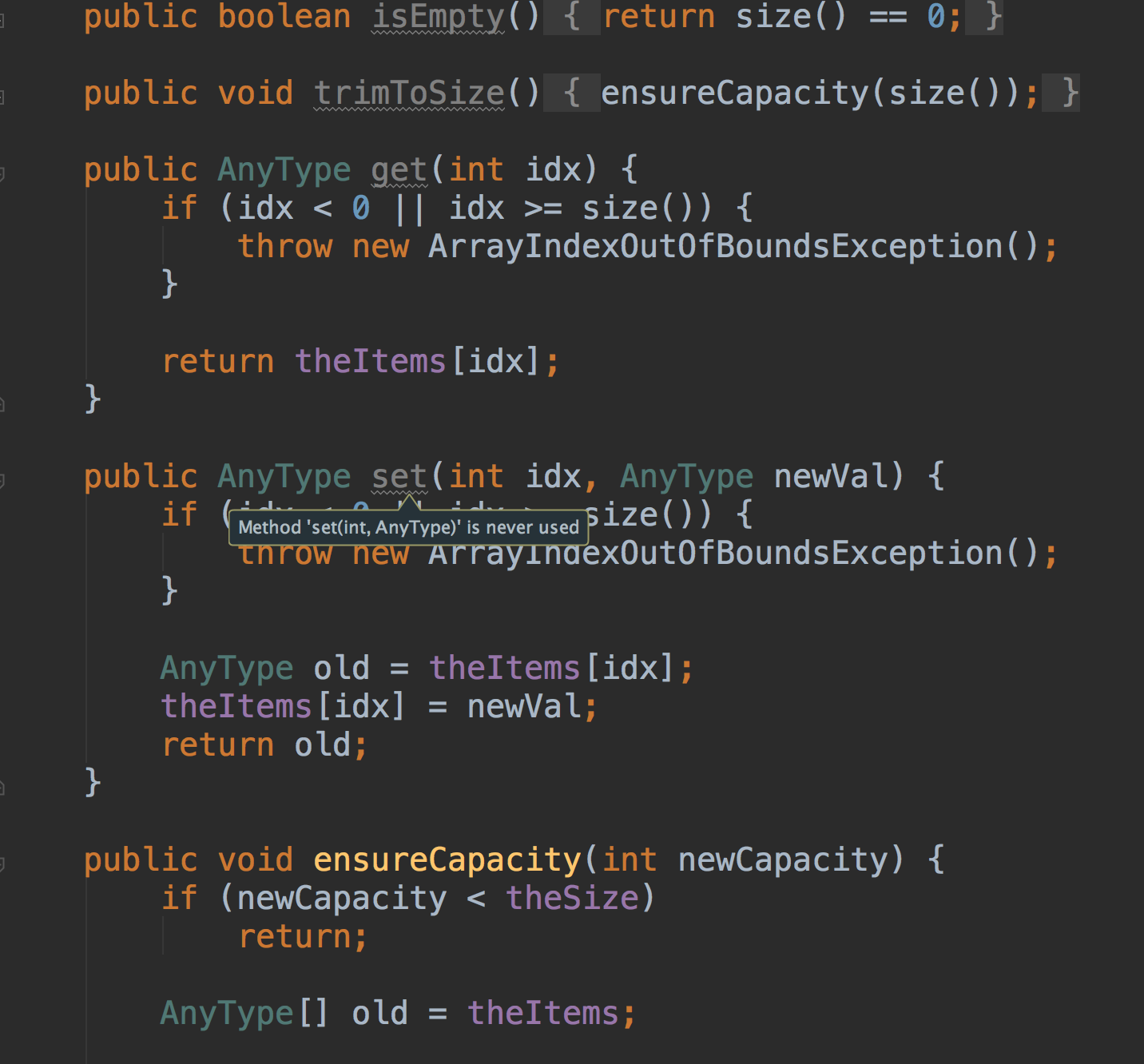
ArrayList应该是我们最常见的泛型结构了，在实际的开发中，ArrayList经常可以代替普通的数组，当成一个自增的数组来用，

这为我们的开发提供了很多的便捷。我们阅读了JDK6.0中ArrayList的实现方式，原来ArrayList是使用了数组来实现自身的，每当数组用完时，数组容量翻倍，并将原数据复制过去。

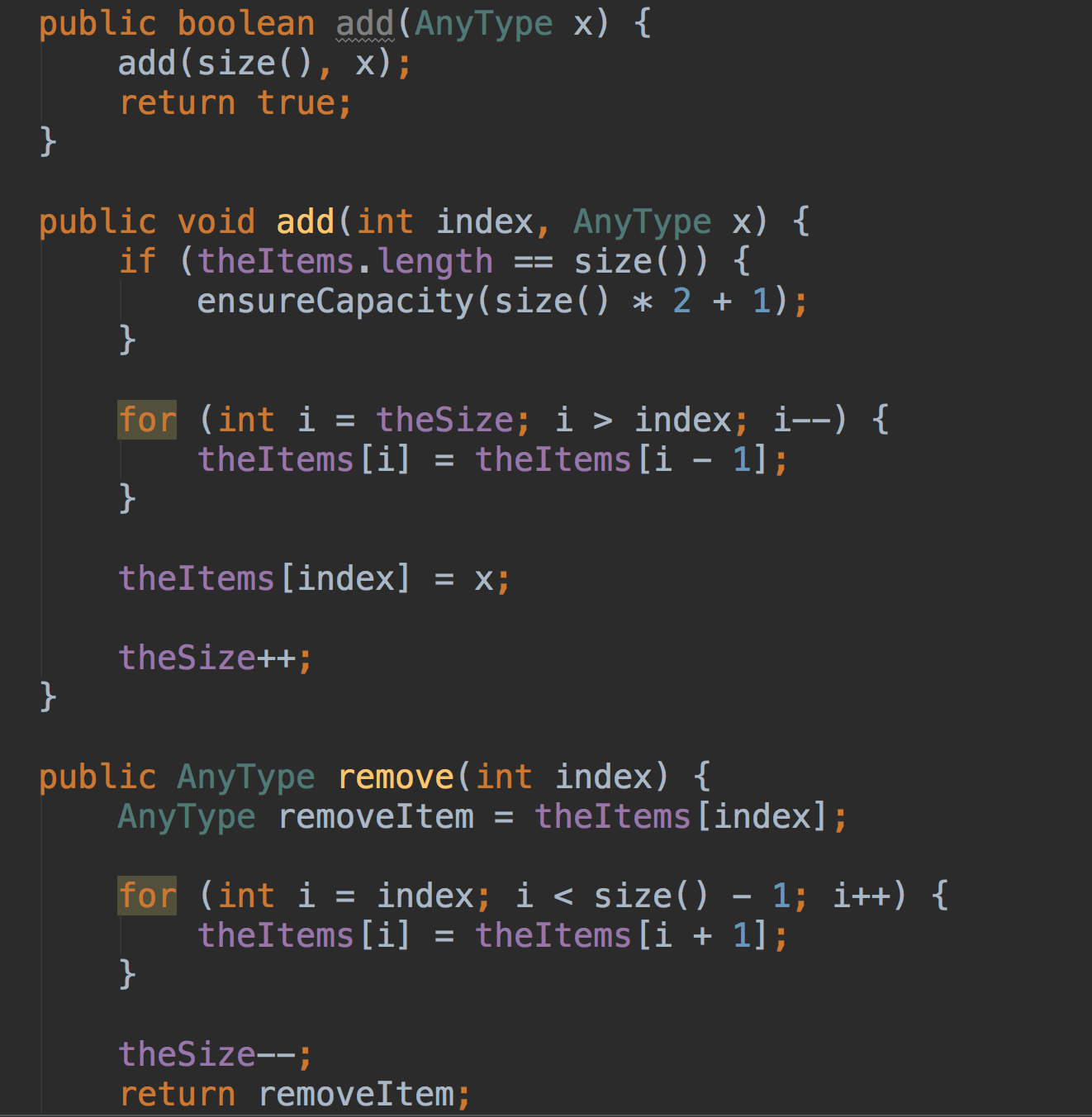
我们学习了源码之后，自己也实现了一次ArrayList的功能，在我们系统的一些部分中，就是用来我们自己写的MyArrayList。



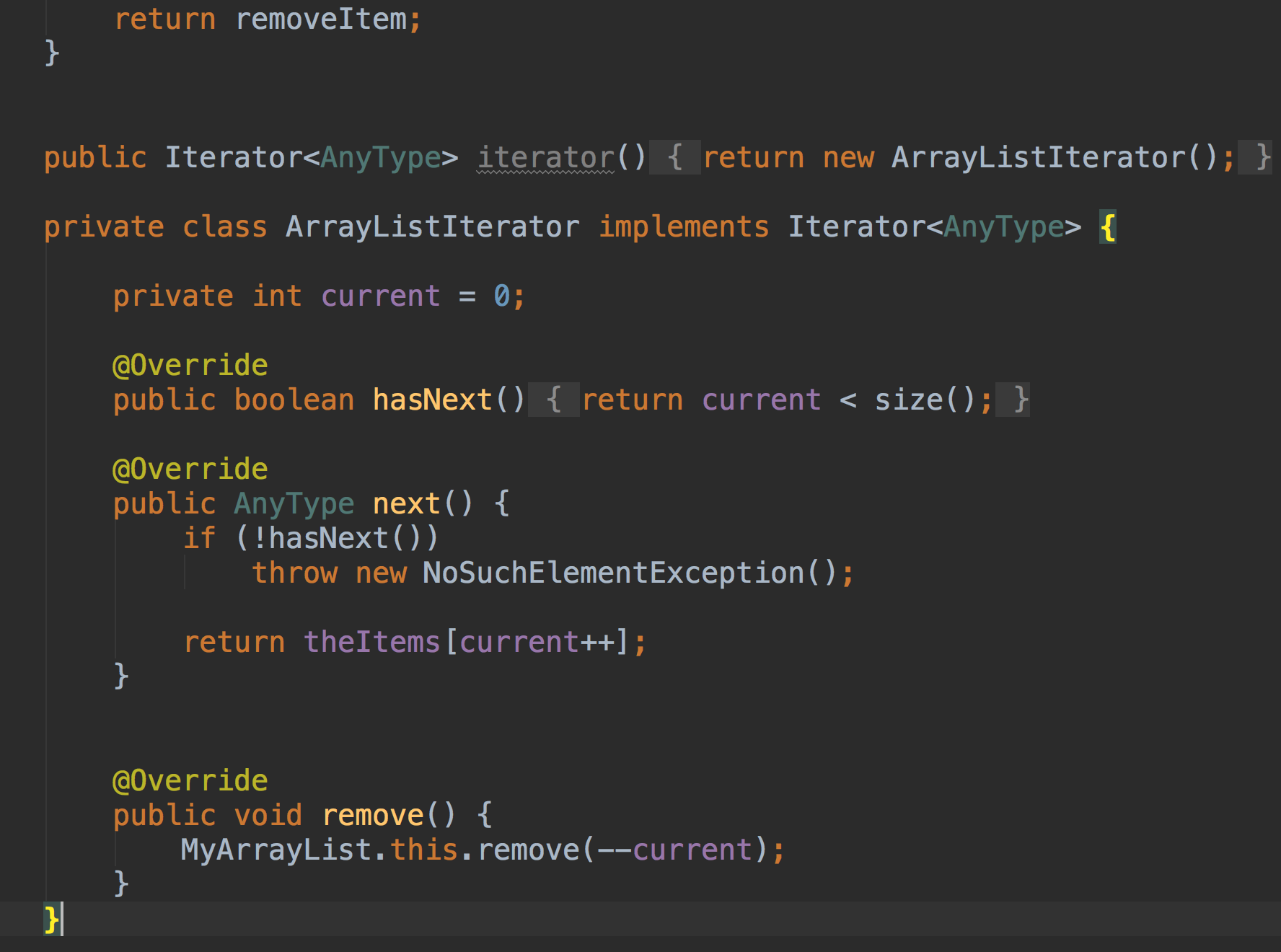
（图 3 - 1）



（图 3 - 2）



（图 3 - 3）



（图 3 - 4）

同时我们注意到和Java中的另一种列表形式的LinkedList的区别，LinedList的实现是使用了链表进行存储，所以在实现上两者各有

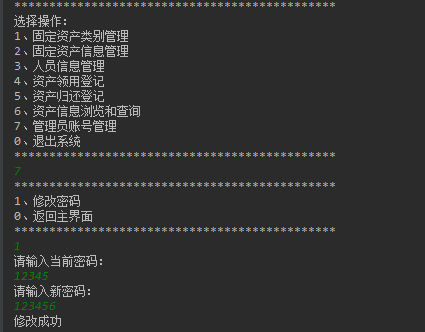
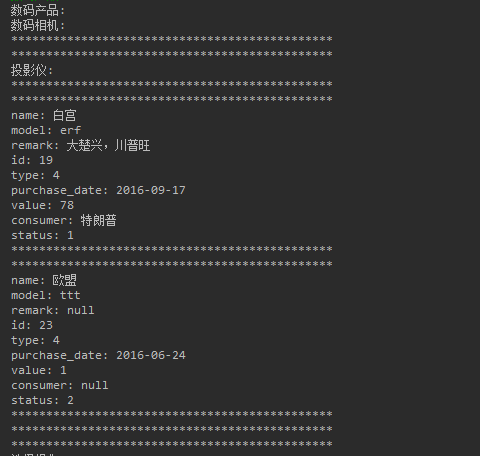
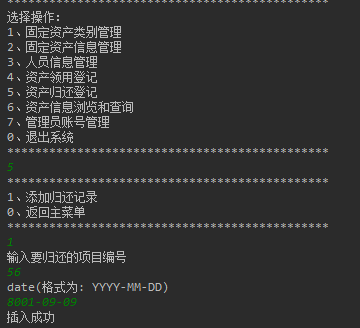
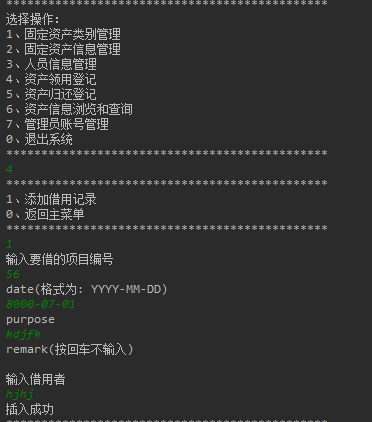
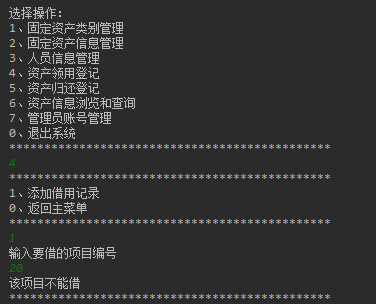
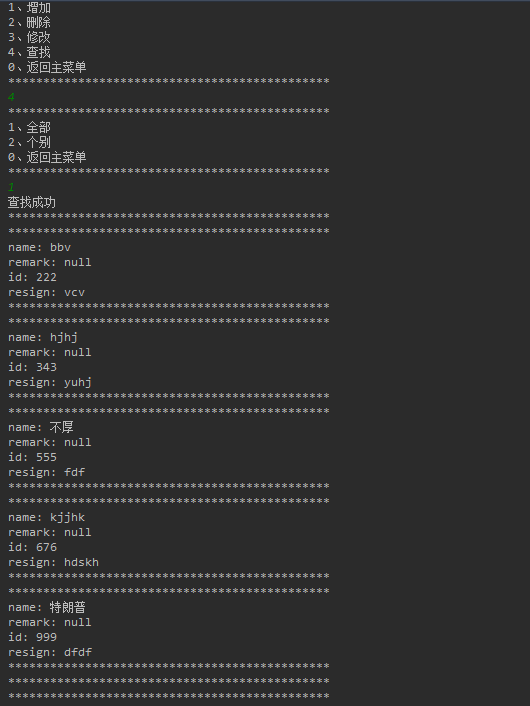
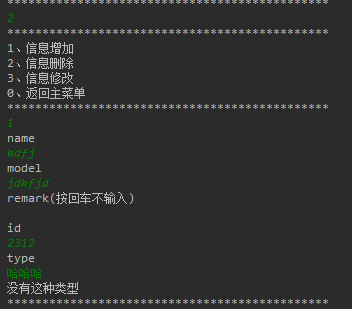
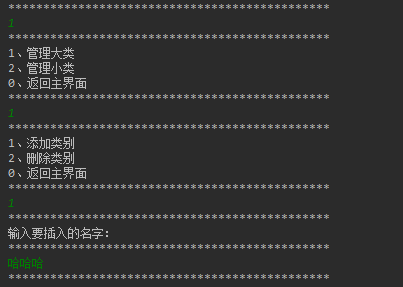
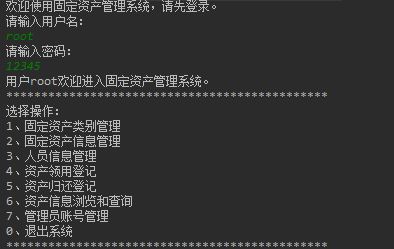
利弊，ArrayList强于索引，LinkedList强于存储，LinkedList不需要连续的数据结构申请。

1. HashMap

我们还用了HashMap这种泛型结构，HashMap其实是一种哈希表的实现，哈希表使用了对应的KEY的哈希值来存储VALUE。每个Java对象都有hashCode（）方法，哈希表就是每次调用KEY的这个方法。如果有哈希值相撞，就会形成一个列表跟在对应节点的后面。

当 HashMap 的每个 bucket 里存储的 Entry 只是单个 Entry ——也就是没有通过指针产生 Entry 链时，此时的 HashMap 具有最好的性能：当程序通过 key 取出对应 value 时，系统只要先计算出该 key 的 hashCode() 返回值，在根据该 hashCode 返回值找出该 key 在 table 数组中的索引，然后取出该索引处的 Entry，最后返回该 key 对应的 value 即可。

4 实现界面



5、对课程的体会

班级： 软件1412班 姓名：刘丰恺 学号：201492167

这个小学期有幸选择了马瑞新老师的Java开发课程，这门课程是我受益匪浅，我们从中不但学会了很多编程的知识，还过程中和最后都进行了丰富而有趣的实践，特别是最后和队友进行一个实例项目的编写，让我们第一次感受到了组队进行项目编写的感觉，这一切的一切都让我们学到了很多东西，下面我想来谈谈我对这门课的切身感受。

我们选择的Java开发课程是每天的上午8:00-11:30，选择这个时间是因为我觉得早上起床能提升我一天的效率，地点在C区的大教室，人很多，也很热，但这些困难我觉得都是可以克服的。因为Java这门课程确实生动而有趣，让我们都忽略了这些客观因素，能够全心投入的去进入这门课程的学习和训练之中。软件工程有别于传统CS课程的地方可能就在于我们有很多真实实践的机会，我个人认为这是弥足珍贵的，因为很多时候我们在学校学习的都是干巴巴的理论（当然理论知识也是很重要的，实践也要在理论的基础上），缺乏独具匠心设计的训练都是一个重大的问题。但是我们的小学期的课设就完美的解决了这个问题，让我们在重点理论知识的学习之余，也能切身的进行一些充足的实践。用马老师的话说：“让我们的能力和公司接轨”。

马老师讲过为什么要开设小学期这个问题，因为根据好几年前的学长们在公司实习的反馈结果来看，我们的学生的实际动手能力居然不如东软信息学院的学生，就我个人而言这种情况真的是令人咋舌，我们堂堂一所985高校的学生居然不如一所专科院校的学生的动手能力。我想归根结底是因为我们平时重视基础，充实思维，但是忽略了我们切身的动手能力，这是很不应该的。所以我们有了Java这门训练课。

我自己之前自问也对Java有一些了解，因为我自己动手参与的项目其实也是挺多的，Android平台的App也写过不少。说起来还参与过很多次马老师当评委的比赛，但是我想说，虽然我有一定的基础，但是我自己的能力也不可能覆盖Java这门语言的方方面面，有所疏漏也在所难免，所以说马老师的这门课就对我起到了很大的帮助，我使用过Java开发过Android甚至还用这个写数据结构作业，自己写解释器，但是我没有用过JDBC的很多功能，这次的训练很好的弥补了我这方面的缺失。我想这门课程是适合每个软件工程专业的学生，无论是有基础的还是没基础的，都能有所获益。

我想说的其他方面就是有关合作的东西，之前其实很少有人合作，开发也基本都是前端和后端之间的合作，没有写相同部分的合作，那我想我之前的合作可能是不深入的，因为合作本身就是个复杂的东西，这次的课程大作业就让我和同学有了一次深入合作的机会，这种合作的机会本事也是弥足珍贵的，总之，希望这门课程能一直在软件学院开设下去，使更多的学生从中受益，提升自己的知识水平。

班级： 软件1412班 姓名：李润之 学号：201492036

我是软英1412班的李润之，这个小学期我参与了Java开发这门课程，使我感受颇深，下面我想谈谈我对这门课的收获和感悟。

这门Java开发课程让我学到了很多东西，我感触最深的就是与人合作这点，我其实很喜欢写程序的，也希望能从事有趣的程序开发的工作的，但是在这之前其实我都没太合别人一起合作过，不过独木不成林，一花不成春，大型软件工程的项目不可能是一个人独立完成的，Linux系统虽然是由Linus开发的，但这其中开源社区的努力也起到了很大的作用。所以说合作是很重要的。

但是《人月神话》这本书里面提到了很多关于软件工程管理的内容，其中就有很多关于提高程序员工作效率的很多的建议。书中说：“不成熟的合作甚至不如一个人单干”。我想现实中也大概就是这样的，人月神话中的“人月”指的是一个人一个月的工作量，研究者经过研究发现，不成熟的合作者合作消耗的人月甚至多过一个人的单独工作。但是没有认识能天生就合作愉快的。每个人都需要不断的适应团队并且磨合。我们难道要到公司才能有机会磨合么？这样会不会显得太晚了，我想这门课的意义也就一方面的体现在这个地方。

我们在这门课上就可以进行类似真实的项目的合作，我想这对于我是很重要的。我和我的队友刘丰恺同学，我们本身就很热爱变成，加上本身比较熟悉，有了这门课提供的机会去练习合作。可以说在合作的过程中，是比较完美的，在设计数据库，实现功能上面虽有一定的分歧，但也很快就打成立一致。

另外，我们这次使用的由华为企业云提供的项目管理软件，也对于我们的工程管理提供了很大的帮助，在这个过程中，我们还得到了来自华为的高级工程师的细心而又专业的讲解，这一切都是我学习编程路上的一块很重要的基石。

我个人写过一些JavaScript的项目，个人主要也是朝着前端攻城师的方向发展，但是这门课程对我的帮助也不是一般的大，因为不但接触了一门有趣且强大的语言，而且比语言更重要的是由老师传授的关于编程的思想性的内容，我想这才是我需要珍而重之的瑰宝。因为我想程序员是解决问题的人，而不是只单单是用某一种语言的人，如果项目有需要，我们就要去学越来越多的新知识，而不单是某种固有的旧知识。

感谢这门课程，能让我亲炙这么多有趣的项目和工程，学到如此多的知识！