

实验名称：人员信息管理系统（v2.2）

设计要求：

利用 java 中的集合类 ArrayList 实现 v1 版本的所有功能。

功能要求：

- 1 可以对学生和职工信息进行增删改查
 - 手工输入 ID
 - 根据 ID 进行精确查询
 - 根据姓名进行模糊查询
 - 可以更新 ID 以外的信息
 - 根据 ID 进行删除
- 2 以命令行菜单形式供用户使用。
- 3 数据不需要持久化。
- 4 对程序进行**重构**，将不同功能的类组织到不同的包中，例如 Student 类属于值对象（Value Object）类别。
- 5 引入表示增删改查等操作的 PersonOperate 接口，该接口接收或返回相关人员对象，发生异常时抛出自定义异常 PersonOpException。并对学生和职工定义实现该接口的操作类，例如 StudentOperate。

```
public class PersonOpException Exception{
    public PersonOpException () {
        this("Person Operation Exception");
    }
    public PersonOpException (String msg) {
        super(msg);
    }
}
```

```
public interface PersonOperate {
```

```
    /**
     * 插入数据的操作
     *
     * @param person
     *         插入的是一个人员信息
     * @throws PersonOpException
```

```

        *          如果有错误，则把错误抛给调用处处理
    */
    public void add(Person p) throws PersonOpException;

    /**
     * 按id删除信息
     *
     * @param id
     *          人员编号
     * @throws PersonOpException
     *          如果有错误，则把错误抛给调用处处理
    */

    public void delete(String id) throws PersonOpException;

    /**
     * 更新数据的操作
     *
     * @param person
     *          更新的具体人员信息
     * @throws PersonOpException
     *          如果有错误，则把错误抛给调用处处理
    */

    public void update(Person p) throws PersonOpException ;

    /**
     * 因为查询是多个，所以要返回ArrayList集合
     *
     * @return 全部的查询结果，一个ArrayList中包含了多个Person对象
     * @throws PersonOpException
    */

    public ArrayList<Persion> findAll() throws PersonOpException;

    /**
     * 按id进行查询
     *
     * @param id
     *          人员编号
     * @return 具体的人员信息
     * @throws PersonOpException
     *          如果有错误，则把错误抛给调用处处理
    */

    public Person findById(String id) throws PersonOpException;

```

```
/**
 * 按关键字对姓名进行模糊查询
 *
 * @param key 输入的关键字
 * @return 全部的查询结果，一个ArrayList中包含了多个Person对象
 * @throws PersonOpException
 */
public ArrayList<Persion> findByName(String key) throws
PersonOpException;

}
```