

上机实验 (六)

(2023 年 2 月 6 日)

实验内容：面向接口编程与异常处理，实现学位颁发

编写一个程序，通过面向接口编程思想，实现学位颁发功能。具体设计要求如下。

- 1、设计一个抽象类 Student，描述学生的基本信息。该抽象类所包含的属性为：

```
String name;
char gender;
Date birthDay;
String school;
String stuId;
String category;
double[] score;
boolean status;

public abstract String getDegree();
```

分别对应学生的姓名、性别、出生日期、学校、学号、学生类别、课程成绩、学位颁发状态、以及获得学位的抽象方法。变量要求包含对应的 set/get 方法。

- 2、设计一个接口 Manageable，提供学位管理功能。该接口定义如下的三个抽象方法：

```
public abstract double computeAverage() throws Exception;
public abstract boolean getPass();
public abstract void printInfo();
```

分别用于计算给定一组数据的平均值、识别是否满足用户给定的约束条件、以及打印一组信息。

- 3、设计一个继承 Exception 的异常类 UndergraduateException，用于捕获大学生异常，定义如下：

```
public class UndergraduateException extends Exception
```

该异常类中定义一个变量 message，用于判断当大学生的某门成绩不在 0-100 范围内，捕获异常信息，通过 warnMess() 方法返回收集的异常信息。

- 4、设计一个继承 Exception 的异常类 MasterException，用于捕获硕士研究生异常，

定义如下：

```
public class MasterException extends Exception
```

该异常类中定义一个变量 `message`，用于判断当大学生的某门成绩不在 0-100 范围内，或者当硕士研究生学位论文等级不在 A-E 范围内，捕获异常信息，通过 `warnMess()` 方法返回收集的异常信息。

5、设计一个继承 `Student` 抽象类、实现 `Manageable` 接口的具体类 `Undergraduate` 类，具体要求如下：

- (1) 定义一个 `double` 型变量 `aver`，用于保存该类大学生课程成绩的平均值；
- (2) 实现 `Manageable` 接口中定义的三个抽象方法。
- (3) 在实现 `computeAverage()` 方法过程中，要求通过算术平均值计算大学生课程平均分。若某门课程不在 0-100 范围，抛出和该分数相关的异常对象。
- (4) 在实现 `getPass()` 方法过程中，根据大学生课程平均分判定，若课程成绩平均分 `aver` 大于等于 60 分，则满足学位颁发要求，改变学位颁发状态。
- (5) 在实现 `getDegree()` 方法过程中，根据学生的学位颁发状态，确定是否授予“学士学位”的决定。
- (6) 在实现 `printInfo()` 方法过程中，分别输出大学生的基本信息（姓名、性别、出生日期、学校、学号、学生类别、各门课程成绩、颁发学位状况）。

6、设计一个继承 `Student` 抽象类、实现 `Manageable` 接口的具体类 `Master` 类，具体要求如下：

- (1) 定义一个 `double` 型变量 `aver`，用于保存该类硕士研究生课程成绩平均值；
定义一个字符型变量 `thesisLevel`，用于保存硕士研究生学位论文等级
(取值范围为‘A’，‘B’，‘C’，‘D’，‘E’)，并提供对应 `get/set` 方法。
- (2) 实现 `Manageable` 接口中定义的三个抽象方法。
- (3) 在实现 `computeAverage()` 方法过程中，要求通过几何平均值计算硕士研究生课程平均分，同时判定学位论文等级的合法性。若某门课程不在 0-100 范围，抛出和该分数相关的大学生异常对象；若学位论文等级不在 A-E 范围内，抛出和学位论文等级相关的硕士生异常对象。
- (4) 在实现 `getPass()` 方法过程中，根据硕士研究生课程平均分和学位论文两

个方面判定,若课程成绩平均分 aver 大于等于 80 分且学位论文等级为'C'及其以上, 则满足学位颁发要求, 改变学位颁发状态。

- (5) 在实现 getDegree() 方法过程中, 根据学生的学位颁发状态, 确定是否授予“硕士学位”的决定。
- (6) 在实现 printInfo() 方法中, 输出硕士生基本信息(姓名、性别、出生日期、学校、学号、学生类别、各门课程成绩、论文等级、颁发学位状态)。

7、设计一个具体类 StudentDegree, 用于管理各类学生的学位授予。具体要求如下:

- (1) 该类包含一个方法:

```
public void issueDegree(Manageable manage)
```

通过调用接口的打印信息方法 (printInfo()) 和计算平均值方法 (computeAverage()) 分别输出学生基本信息和学生课程成绩平均分 (不同类别学生对应的学生基本信息不同)。

- (2) 提示: 在调用计算平均值方法 (computeAverage()) 过程中, 需要捕获可能的异常信息, 包括大学生的课程成绩异常和硕士研究生的学位论文等级异常。
- (3) 提示: 在调用以上方法过程中, 利用了面向接口编程思想, manage 接受实现了该接口的任何类创建的对象, 则调用接口方法时实际上是调用传入类对象所实现的方法。

8、设计一个测试类 StudentApplication, 测试学生学位授予情况。具体要求如下:

- (1) 创建一个 Undergraduate 类对象 undergraduate, 采用如下赋值:

```
String name="王小二";
char gender='男';
Date birthday="2001-6-1";
String school="上海大学";
String stuId="11128981";
String category="大学本科生";
double[] score={89.5, 82, 87, 73};
boolean status=false;
```

- (2) 创建一个 Master 对象 master, 采用如下赋值:

```
String name="李燕";
```

```

char gender="女";
Date birthday="1997-6-12";
String school="上海大学";
String stuId="10306";
String category='硕士研究生';
double[] score={70, 52.5, 95, 88, 89, 91};
char thesisLevel='B';
boolean status=false;

```

- (3) 创建一个 StudentDegree 类的对象 studentDegree，用于测试学位授予。
- (4) 调用 studentDegree 的成员方法 issueDegree(Manageable manage)，并分别使用 undergraduate 和 master 类作为实参传入，测试大学生对象和硕士生对象的学位授予情况。

9、创建一个类 Constants，里面定义各类在此程序中使用到的常量，分别如下：

```

public class Constants {
    public final static String UNDERGRADUATE = "大学本科生";
    public final static String MASTER = "硕士研究生";
    public final static String ISSUE_UNDERGRADUATE_DEGREE="颁发学士学位";
    public final static String ISSUE_MASTER_DEGREE="颁发硕士学位";
    public final static String UNSATISFACTORY_ISSUE_UNDERGRADUATE_DEGREE="
        不满足颁发学士学位要求";
    public final static String UNSATISFACTORY_ISSUE_MASTER_DEGREE="不满足
        颁发硕士学位要求";
}

```

注:	1.	本次实验提交实验内容代码;
	2.	将内容打包，选择第 6 次实验作业对应编号，提交至课程管理平台。