



深圳大学
Shenzhen University

深圳大学实验报告

课程名称: Java 程序设计

实验项目名称: 选实验 3 包及继承及应用

学院: 计算机与软件学院

专业: 计算机科学与技术

指导教师: 毛斐巧

报告人: 黎浩然 学号: 2018112061 班级: 01

实验时间: 2019 年 10 月 18 日 (周五) - 2019 年 10 月 24 日 (周四)

实验报告提交时间: 2019/10/16

教务部制

一、实验目的

1. 熟悉面向对象编程中 package, import 等语句的使用。。

二、实验内容与要求

1. 编写一个 CSSE 类和一个 Teacher 类。CSSE 类中包含有多个 Teacher 类的实例（所有老师的名字请查阅 <http://csse.szu.edu.cn/cn/people>）。调用 CSSE 类的实例中的 getTeacherNames 方法时，能够输出学院中所有老师的名字。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

2. 把 CSSE 类和 Teacher 类放进 cn.edu.szu 包中。编写一个测试类，在源代码中用 import 语句引入 cn.edu.szu 包中的所有类，并对它们所包含的方法进行测试。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

3. 定义一个包，在该包中定义一个类并输出 “I love SZU” 字符串。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

4. 在一个包中定义一个默认访问权限控制的类 DefaultClass，然后在该包以及另一个包中分别定义类来测试 DefaultClass 类中的 protected 和 public 数据成员的访问控制能力。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

5. 在一个包中定义一个 public 访问权限控制的类 PublicClass，然后在该包以及另一个包中分别定义类来测试 PublicClass 类中的 protected 和 public 数据成员的访问控制能力。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

6. 编写一个抽象类 Animal，它具有两个抽象方法 run 和 eat。分别实现这个抽象类的三个子类 Tiger、Cat 和 Dog。实现一个测试类，在测试类的 main 方法中分别使用这三个子类创建对象实体，然后通过它们的上转型对象变量调用 run 方法和 eat 方法。在报告中附上程序截图、运行结果截图和详细的文字说明。（15 分）

7. 一个三维向量由三个分量组成。三维向量的相加和相减等价于对应三个分量的相加和相减。比如两个三维向量 (5,2,1) 和 (3,-1,0)，它们的和为 (8,1,1)，它们的差为 (2,3,1)。编写一个接口 Computable，它具有两个抽象方法 add 和 minus。编写一个 Vector 类，通过 Computable 接口实现三维向量的相加和相减。在报告中附上程序截图、运行结果截图和详细的文字说明。（15 分）

报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰可读性强，报告整体写作较为专业。（20 分，这一项的评分，采用 20 分、10 分和 0 分三个级别）

说明：

- （1）本次实验课作业满分为 100 分，占总成绩的比例（待定）。
- （2）本次实验课作业截至时间 2019 年 10 月 31 日（周四）23:59。
- （3）报告正文：请在指定位置填写，本次实验不需要单独提交源程序文件。
- （4）个人信息：WORD 文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。
- （5）提交方式：截至时间前，请在 Blackboard 平台中提交。
- （6）发现抄袭（包括复制&粘贴整句话、整张图），该次作业记零分。
- （7）延迟提交，不得分；如有特殊情况，请于截至日期之后 48 小时内发邮件到 panweike@szu.edu.cn，并在邮件中注明课程名称、作业名称、姓名、学号等信息，以及特殊情况说明，我收到后会及时回复。

(8) 期末考试阶段补交无效。

三、实验过程及结果

1.编写一个 CSSE 类和一个 Teacher 类。CSSE 类中包含有多个 Teacher 类的实例（所有老师的名字请查阅 <http://csse.szu.edu.cn/cn/people>）。调用 CSSE 类的实例中的 getTeacherNames 方法时，能够输出学院中所有老师的名字。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

程序代码如下：

Teacher 类：

```
class Teacher{
    private String name;
    public Teacher(String name){
        this.name = name;
    }
    public void getName() {
        System.out.println(name);
    }
}
```

CSSE 类：

```
public class CSSE {
    Teacher ZhangZhihong = new Teacher("张志宏");
    Teacher Basker = new Teacher("Basker");
    Teacher BaiJiancong = new Teacher("白鉴聪");
    Teacher CaiShubing = new Teacher("蔡树彬");
    Teacher LiYanhong = new Teacher("李延红");
    Teacher LiZhi = new Teacher("李志");
    Teacher LinShaocong = new Teacher("林少聪");
    Teacher LiuGang = new Teacher("刘刚");
    Teacher MaoFeiqiao = new Teacher("毛斐巧");
    Teacher RuanYuan = new Teacher("阮元");
    Teacher XiaoZhiqiao = new Teacher("肖志娇");
    Teacher YangFang = new Teacher("杨芳");
    Teacher ZhangXiaojian = new Teacher("张小健");
    Teacher ZhangYan = new Teacher("张艳");
    Teacher ZhuangHua = new Teacher("庄华");
}
```

getTeacherNames 方法：

```

        public void getTeacherNames() {
            ZhangZhihong.getName();
            Basker.getName();
            BaiJiancong.getName();
            CaiShubing.getName();
            LiYanhong.getName();
            LiZhi.getName();
            LinShaocong.getName();
            LiuGang.getName();
            MaoFeiqiao.getName();
            RuanYuan.getName();
            XiaoZhiqiao.getName();
            YangFang.getName();
            ZhangXiaojian.getName();
            ZhangYan.getName();
            ZhuangHua.getName();
        }

        public static void main(String[] args) {
            // TODO Auto-generated method stub
            CSSE college = new CSSE();
            college.getTeacherNames();
        }
    }
}

```

程序运行结果：

```

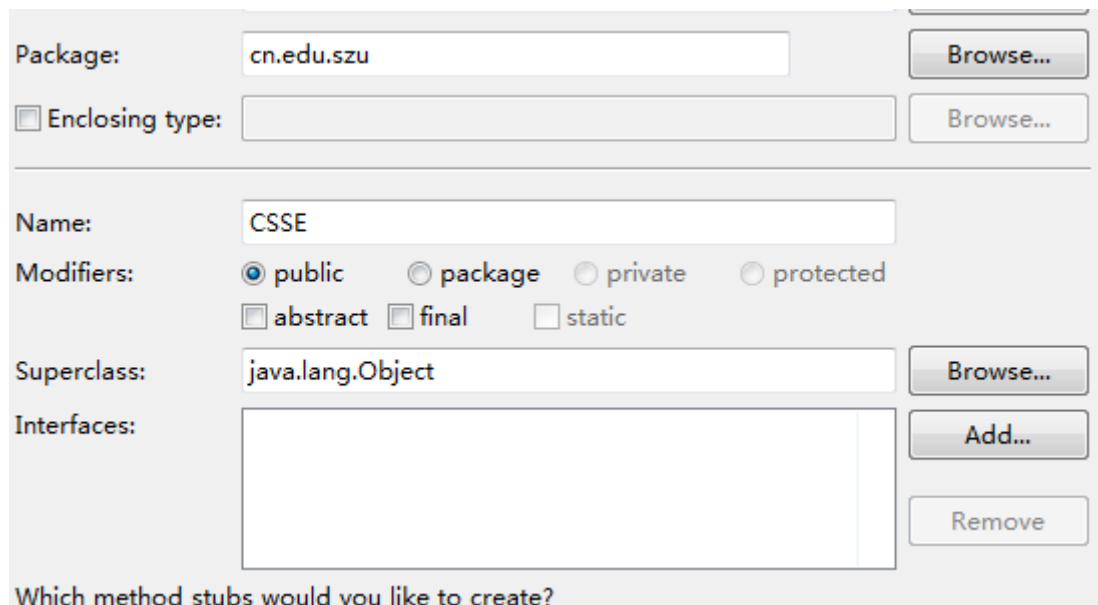
张志宏
Basker
白鉴聪
蔡树彬
李延红
李志
林少聪
刘刚
毛斐巧
阮元
肖志娇
杨芳
张小健
张艳
庄华

```

2.把 CSSE 类和 Teacher 类放进 cn.edu.szu 包中。编写一个测试类，在源代码中用 import 语句引入 cn.edu.szu 包中的所有类，并对它们所包含的方法进行测试。在报告中附上程序截

图、运行结果截图和详细的文字说明。(10 分)

在 cn.edu.szu 包中存放 Teacher 类和 CSSE 类:



Package: cn.edu.szu Browse...

☐ Enclosing type: Browse...

Name: CSSE

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

将第一题中的代码拷贝过去:

```
package cn.edu.szu;

class Teacher{
    private String name;
    public Teacher(String name){
        this.name = name;
    }
    public void getName() {
        System.out.println(name);
    }
}

public class CSSE {
    Teacher ZhangZhihong = new Teacher("张志宏");
    Teacher Basker = new Teacher("Basker");
    Teacher BaiJiancong = new Teacher("白鉴聪");
    Teacher CaiShubing = new Teacher("蔡树彬");
    Teacher LiYanhong = new Teacher("李延红");
    Teacher LiZhi = new Teacher("李志");
    Teacher LinShaocong = new Teacher("林少聪");
    Teacher LiuGang = new Teacher("刘刚");
    Teacher MaoFeiqiao = new Teacher("毛斐巧");
    Teacher RuanYuan = new Teacher("阮元");
    Teacher XiaoZhiqiao = new Teacher("肖志娇");
    Teacher YangFang = new Teacher("杨芳");
    Teacher ZhangXiaojian = new Teacher("张小健");
}
```

建立测试类:

```
package testpackage;

import cn.edu.szu.*;

public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        CSSE college = new CSSE();
        college.getTeacherNames();
    }

}
```

运行结果如下:

```
张志宏
Basker
白鉴聪
蔡树彬
李延红
李志
林少聪
刘刚
毛斐巧
阮元
肖志娇
杨芳
张小健
张艳
庄华
```

3.定义一个包,在该包中定义一个类并输出“I love SZU”字符串。在报告中附上程序截图、运行结果截图和详细的文字说明。(10分)

程序如下:

```
package cn.edu.szu.csse;

class Confession{
    public static void Says_to_szu() {
        System.out.println("I love SZU");
    }
}

public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Confession.Says_to_szu();
    }

}
```

运行结果如下：

```
I love SZU
```

4. 在一个包中定义一个默认访问权限控制的类 `DefaultClass`，然后在该包以及另一个包中分别定义类来测试 `DefaultClass` 类中的 `protected` 和 `public` 数据成员的访问控制能力。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）

程序如下：

`DefaultClass` 类：

```
package cn.edu.szu.csse;

class DefaultClass{
    protected int a = 1;
    public int b = 2;
    protected void forTest_pro() {
        System.out.println("This is a protected method");
    }
    public void forTest_pub() {
        System.out.println("This is a public method");
    }
}
```

与 `DefaultClass` 在同一个包中的测试类：

```
package cn.edu.szu.csse;

import cn.edu.szu.csse.DefaultClass;

public class Test{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DefaultClass test = new DefaultClass();

        System.out.println(test.a);
        System.out.println(test.b);
        test.forTest_pro();
        test.forTest_pub();
    }

}
```

编译器没有报错，运行结果如下：

```
1
2
This is a protected method
This is a public method
```

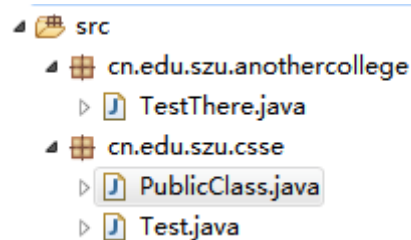
可见，在同一个包中，测试类对 DefaultClass 类的 protected 和 public 数据成员均可访问

在另一个包中的测试类：

```
1 package cn.edu.szu.anothercollege;
2
3 import cn.edu.szu.csse.DefaultClass;
4
5 public class TestThere {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         DefaultClass test = new DefaultClass();
10
11         System.out.println(test.a);
12         System.out.println(test.b);
13         test.forTest_pro();
14         test.forTest_pub();
15     }
16
17 }
18
```

编译器在引入 DefaultClass 类时已经报错，可见，一个类不能访问另一个包中的 DefaultClass

5. 在一个包中定义一个 public 访问权限控制的类 PublicClass，然后在该包以及另一个包中分别定义类来测试 PublicClass 类中的 protected 和 public 数据成员的访问控制能力。在报告中附上程序截图、运行结果截图和详细的文字说明。（10 分）
包及类的位置如下：



```
src
├── cn.edu.szu.anothercollege
│   └── TestThere.java
└── cn.edu.szu.csse
    ├── PublicClass.java
    └── Test.java
```

PublicClass 类如下：

```
package cn.edu.szu.csse;

public class PublicClass{
    protected int a = 1;
    public int b = 2;
    protected void forTest_pro(){
        System.out.println("This is a protected method.");
    }
    public void forTest_pub() {
        System.out.println("This is a public method.");
    }
}
```


在同一个包中的测试类(Test):

```
package cn.edu.szu.csse;

import cn.edu.szu.csse.PublicClass;

public class Test{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        PublicClass test = new PublicClass();

        System.out.println(test.a);
        System.out.println(test.b);
        test.forTest_pro();
        test.forTest_pub();
    }

}
```

编译器没有报错，运行结果如下：

```
1
2
This is a protected method.
This is a public method.
```

[可见，在同一个包中，测试类对 PublicClass 的 protected 和 public 均可访问](#)

在另一个包中的测试类 (TestThere):

```
package cn.edu.szu.anothercollege;

import cn.edu.szu.csse.PublicClass;

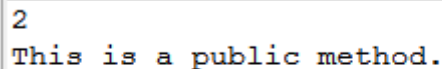
public class TestThere {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        PublicClass test = new PublicClass();

        System.out.println(test.a);
        System.out.println(test.b);
        test.forTest_pro();
        test.forTest_pub();
    }

}
```

编译器在访问 `protected` 成员的时候报错
注释掉报错的两行，运行结果如下



```
2  
This is a public method.
```

可见，在不同的包中，测试类对 `PublicClass` 类的 `public` 可访问，`protected` 不可访问

6. 编写一个抽象类 `Animal`，它具有两个抽象方法 `run` 和 `eat`。分别实现这个抽象类的三个子类 `Tiger`、`Cat` 和 `Dog`。实现一个测试类，在测试类的 `main` 方法中分别使用这三个子类创建对象实体，然后通过它们的上转型对象变量调用 `run` 方法和 `eat` 方法。在报告中附上程序截图、运行结果截图和详细的文字说明。（15 分）

程序代码如下：

`Animal`，`Tiger`，`Cat` 及 `Dog` 类：

```
abstract class Animal{           //abstract class Animal  
    abstract void run();  
    abstract void eat();  
}  
  
class Tiger extends Animal{  
    void run() {  
        System.out.println("Tiger typically has a speed of 30-40 mph.");  
    }  
    void eat() {  
        System.out.println("Tiger eat Moose, Deer, cattle, Horse, "  
            + "Goat etc.");  
    }  
}  
  
class Cat extends Animal{  
    void run() {  
        System.out.println("Cat can run up to 30 mph.");  
    }  
    void eat() {  
        System.out.println("Cat eat cooked beef, chicken, turkey, "  
            + "and small amounts of lean deli meats.");  
    }  
}  
  
class Dog extends Animal{  
    void run() {  
        System.out.println("Dog can run about 15-20 mph.");  
    }  
    void eat() {  
        System.out.println("Dog are largely carnivores but also"  
            + " eat plant-based foods.");  
    }  
}
```

测试类:

```
public class Test {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Animal tiger = new Tiger();    //Tiger上转型变量  
        Animal cat = new Cat();        //Cat上转型变量  
        Animal dog = new Dog();        //Dog上转型变量  
        tiger.run();  
        tiger.eat();  
        cat.run();  
        cat.eat();  
        dog.run();  
        dog.eat();  
    }  
}
```

运行结果如下:

```
Tiger typically has a speed of 30-40 mph.  
Tiger eat Moose, Deer, cattle, Horse, Goat etc.  
Cat can run up to 30 mph.  
Cat eat cooked beef, chicken, turkey, and small amounts of lean deli meats.  
Dog can run about 15-20 mph.  
Dog are largely carnivores but also eat plant-based foods.
```

7.一个三维向量由三个分量组成。三维向量的相加和相减等价于对应三个分量的相加和相减。比如两个三维向量 (5,2,1) 和 (3,-1,0)，它们的和为 (8,1,1)，它们的差为 (2,3,1)。编写一个接口 **Computable**，它具有两个抽象方法 **add** 和 **minus**。编写一个 **Vector** 类，通过 **Computable** 接口实现三维向量的相加和相减。在报告中附上程序截图、运行结果截图和详细的文字说明。(15 分)

程序如下:

接口 **Computable**:

```
interface Computable{  
    public abstract void add(Vector v);    //抽象方法add  
    public abstract void minus(Vector v);  //抽象方法minus  
}
```

Vector 类:

```

class Vector implements Computable{
    private double x = 0.0;
    private double y = 0.0;
    private double z = 0.0;
    public Vector(double x, double y, double z){
        this.x = x;
        this.y = y;
        this.z = z;
    }
    public void add(Vector v) {
        this.x += v.x;
        this.y += v.y;
        this.z += v.z;
    }
    public void minus(Vector v) {
        this.x -= v.x;
        this.y -= v.y;
        this.z -= v.z;
    }
    public void display() {
        System.out.printf("(%.2f, %.2f, %.2f)\n", x, y, z);
    }
}

```

测试类:

```

public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Vector a = new Vector(1, 2, 3);
        Vector b = new Vector(4, 5, 6);
        a.add(b);          //a = a + b
        a.display();
        a.minus(b);        //a = a - b
        a.display();
    }

}

```

运行结果:

```

terminated: Test (7) Java Application [C:\Program Files\
(5.00, 7.00, 9.00)
(1.00, 2.00, 3.00)

```

四、实验总结与体会

(写写感想、建议等)

1. 在 import 另一个包中的类时, 只能引入 public 类, 不能引入非 public 类; 而且只对 public 类的 public 数据成员可访问。
2. 在 import 同一个包中的类时, 可以引入非 public 类, 并且对 protected 和 public 数据成员均可访问

五、成绩评定及评语

1.指导老师批阅意见：

2.成绩评定：

指导教师签字：毛斐巧
2019 年 月 日

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。