

## 实验一：java 类与对象

### 【实验目的】

1. 掌握Java类的定义及使用；
2. 理解构造方法的使用；
3. 掌握对象的声明和使用；
4. 数组的定义与使用。

尽可能用private

### 【实验内容】

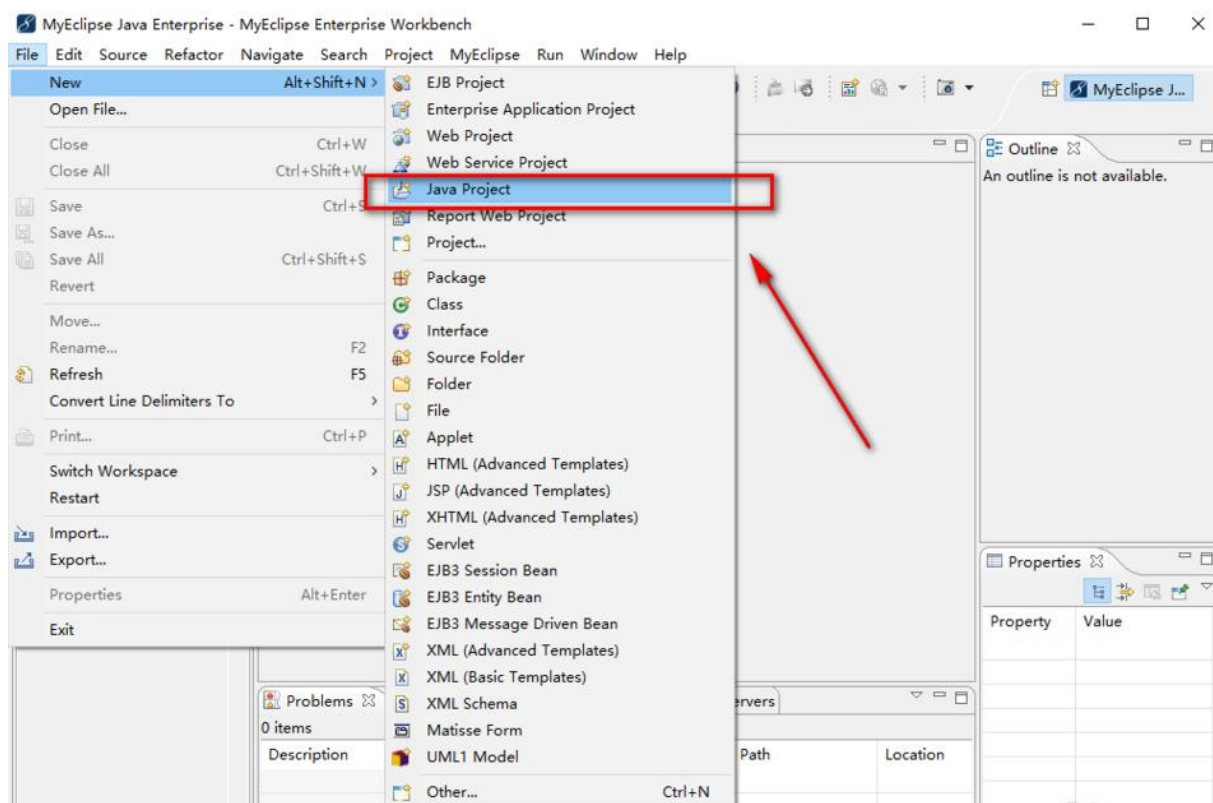
编写一个Java程序片断，定义一个表示雇员的类Employee，

### 【实验要求】

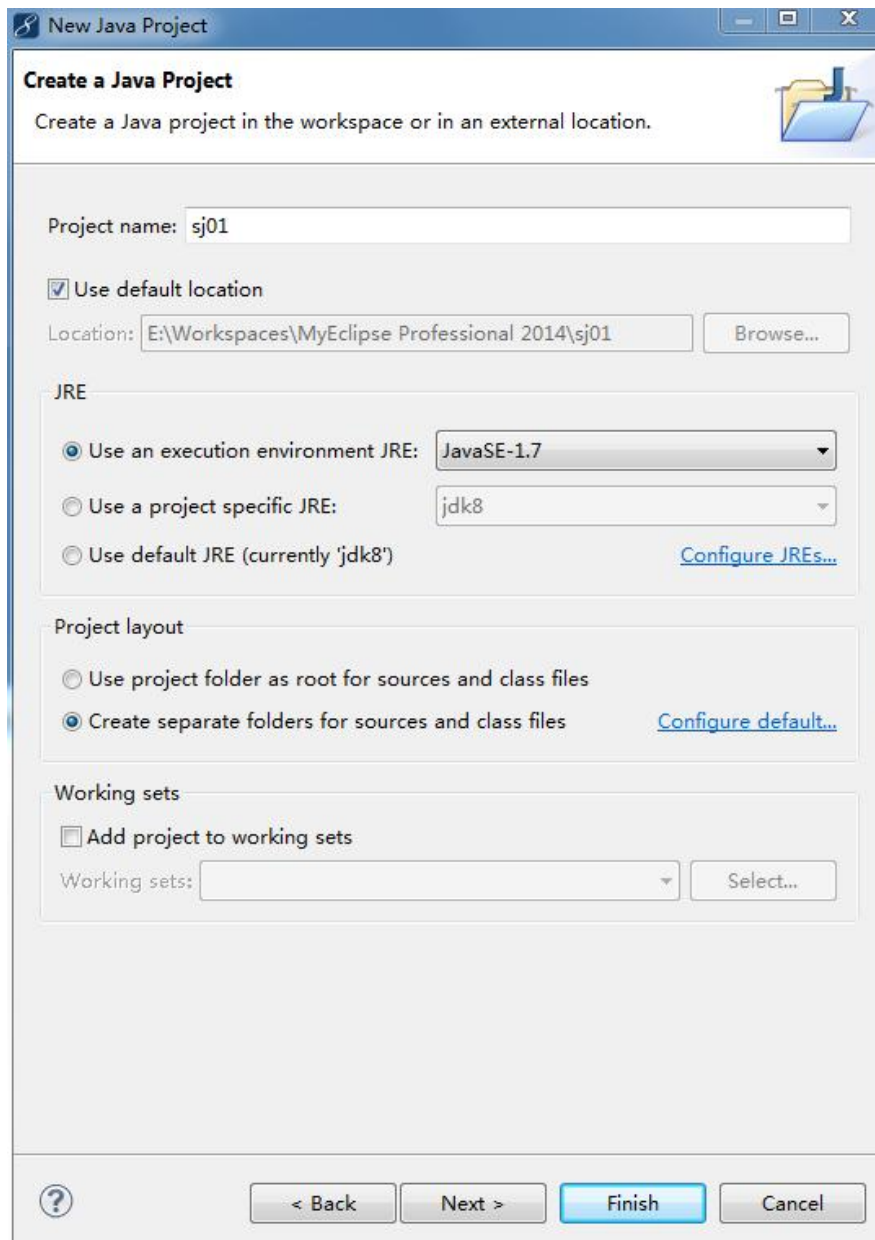
这个类的属性有“sid”、“sname”、“sex”、“age”，方法有“getSid”、“getName”、“getSex”、“getAge”，通过构造方法给类中所有属性赋值。为类Employee增加一个方法public String toString()，该方法把Employee类的对象的所有属性信息组合成一个字符串输出。实例化Employee类的对象，调用对象的toString方法并打印。

### 【上机指导】

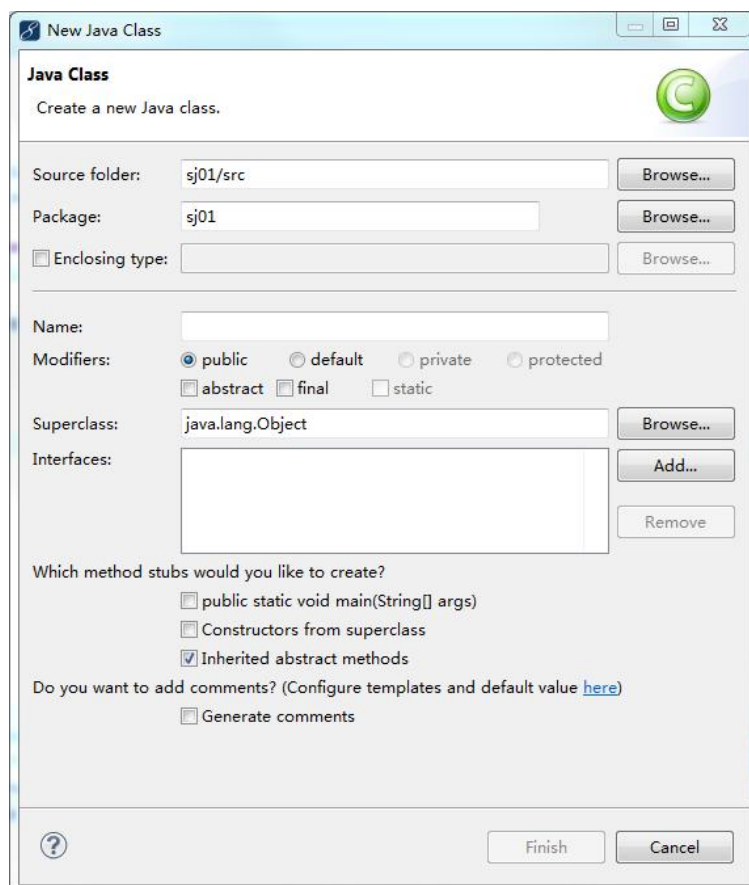
- (1) 使用单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File”->“New”->“Project”，如图：



- (3) 为创建的工程起名为sj01, 点“完成”，如图：



(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Employee”，点“完成”；

(5)、在Employee类文件中输入如下代码：

```
public class Employee{
    private String sname,sex;
    int age,sid;
    Employee (int id,String name ,String Sex,int Age){
        sid=id;
        sname=name;
        sex=Sex;
        age=Age;
    }
    int getSid(){
        return sid;
    }
    String getSname(){
        return sname;
    }
    String getSex(){
        return sex;
    }
    int getAge(){
        return age;
    }
}
```

权限控制

用this指针，  
指向当前变量

因为和静态变  
量做区分

代码的可读性  
与规范

```
}  
public String toString( ){  
    String a;  
    a="sid:"+sid+"",sname:"+sname+"  
+","sex:"+sex+"",age:"+age;  
    return a;  
}  
}
```

(6) 创建TestOne类，并输入如下代码：

```
public class Test01{  
    public static void main(String[] args) {  
        String name="zhangsan",sex="nan";  
        int age=10,id=1;  
        Employee employee_demo=new Employee(id,name,sex,age);  
        String a;  
        a= employee_demo.toString();  
        System.out.println(a);  
    }  
}
```

(7) 在” Test01”上右键，选择 “Run AS” -> “Java Application”，结果如图：



## 实验二：继承、接口与多态

### 【实验目的】

1. 掌握类的继承机制；
2. 熟悉类中成员变量和方法的访问控制；
3. 熟悉接口的定义方法；
4. 熟悉方法或构造方法多态性。

### 【实验内容 1】

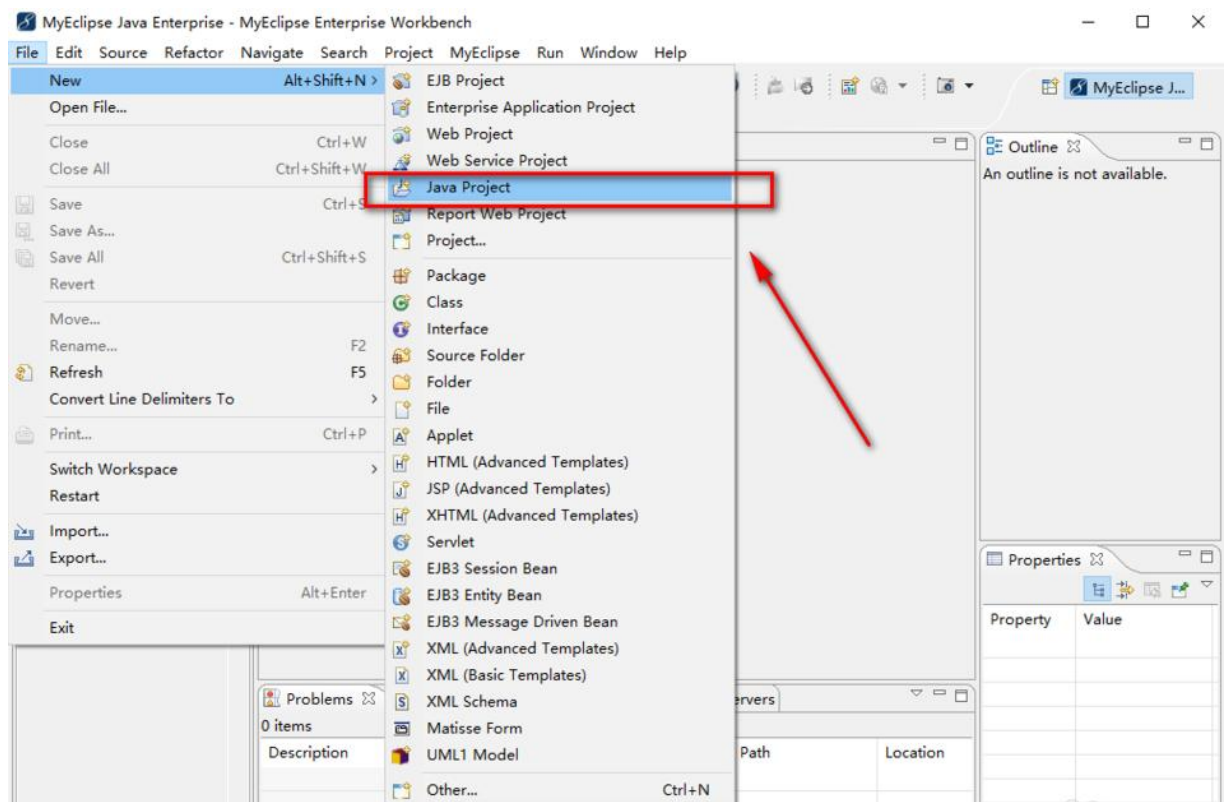
创建父类、子类实现继承

### 【实验要求】

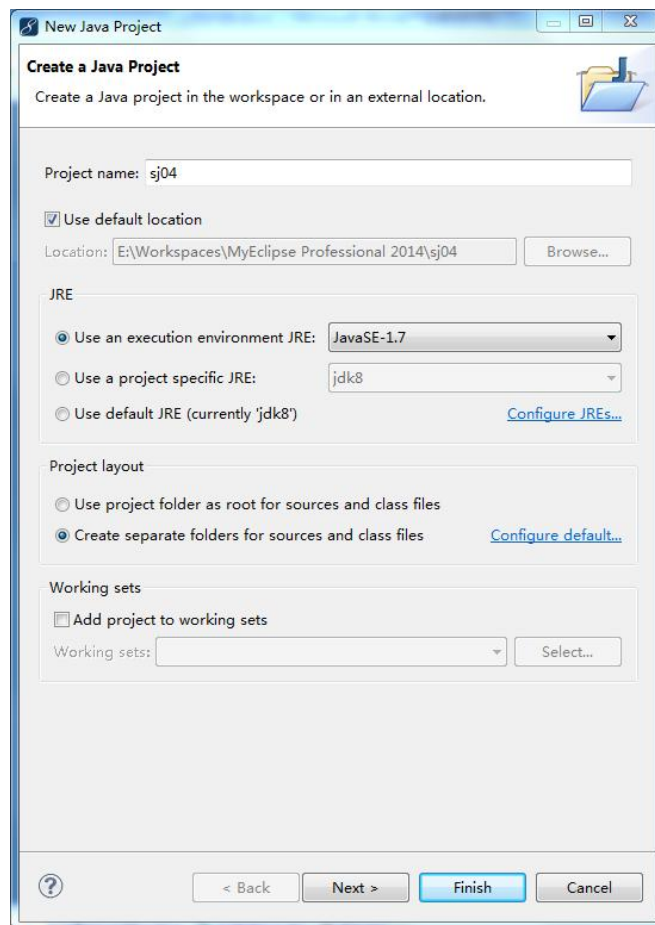
编程实现类的继承。编写父类People，子类Employee继承自人类。人类具有姓名，性别，年龄等性质，还具有吃和说的行为。雇员类继承父类，还拥有工号性质和工作行为。构造人类和学生类的对象，调用吃、说、工作的方法输出有关信息。

### 【上机指导】

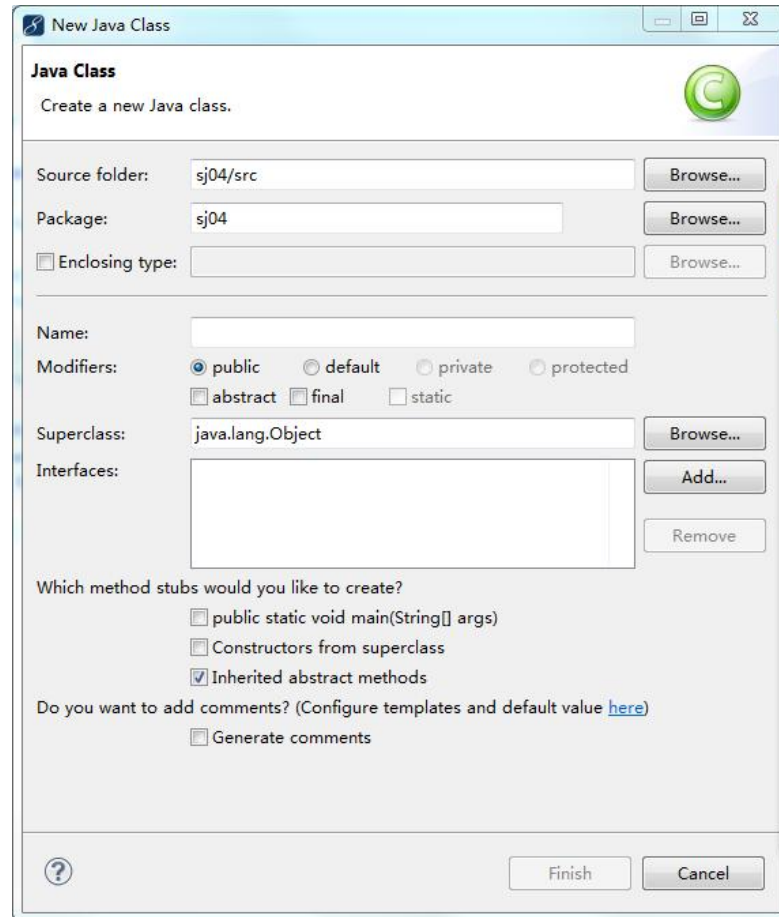
- (1) 单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File”->“New”->“Project”，如图：



- (3) 为创建的工程起名为sj04, 点“完成”，如图：



(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“People”，点“完成”；

(5) 在People类文件中输入如下代码：

```
public class People {  
  
    protected String name; //姓名  
    protected String gender; //性别  
    protected int age; //年龄  
  
    public void eat() { //吃  
        System.out.println("我是人，我爱吃饭！");  
    }  
    public void speak() { //说话  
        System.out.println("我是人， 我爱说话！");  
    }  
    public String getName(){  
        return this.name;  
    }  
    public String getGender(){  
        return this.gender;  
    }  
    public int getAge(){
```

```
        return this.age;
    }

    public People(String name, String gender, int age) { //初始化共有属性
        this.name = name;
        this.gender = gender;
        this.age = age;
    }
}
```

(6) 创建Employee类，并输入如下代码：

```
public class Employee extends People{
    protected long EmpNo; //工号
    public void eat() {
        System.out.println("我是员工，我爱吃饭");
    }
    public void speak() {
        System.out.println("我是员工，我爱说话");
    }
    public void work() { //
        System.out.println("我是员工，我的工作内容很简单！");
    }
    public long getEmpNo(){
        return this.EmpNo;
    }

    public Employee(String name, String gender, int age, long EmpNo) { //
        //构造方法
        super(name, gender, age); //重定义父类成员变量
        this.EmpNo = EmpNo;
    }
}
```

(7) 创建Test04类，并输入如下代码：

```
public class Test04 {
    public static void main(String[] args) {
        Employee employee = new Employee("张三","男",18,13000000);

        System.out.println("这是一名员工：");
        System.out.println("姓名： "+employee.getName());
        System.out.println("性别： "+employee.getGender());
        System.out.println("年龄： "+employee.getAge());
        System.out.println("工号： "+employee.getEmpNo());
        employee.eat();
        employee.speak();
    }
}
```



```
        employee.work();  
        System.out.println();  
        People people = new People("丽丝","女",16);  
        System.out.println("这是一个普通的人：");  
        System.out.println("姓名：" + people.getName());  
        System.out.println("性别：" + people.getGender());  
        System.out.println("年龄：" + people.getAge());  
        people.eat();  
        people.speak();  
    }  
}
```

(8) 在“test04”上右键，选择“Run AS” -> “Java Application”，结果如图：



## 【实验内容 2】

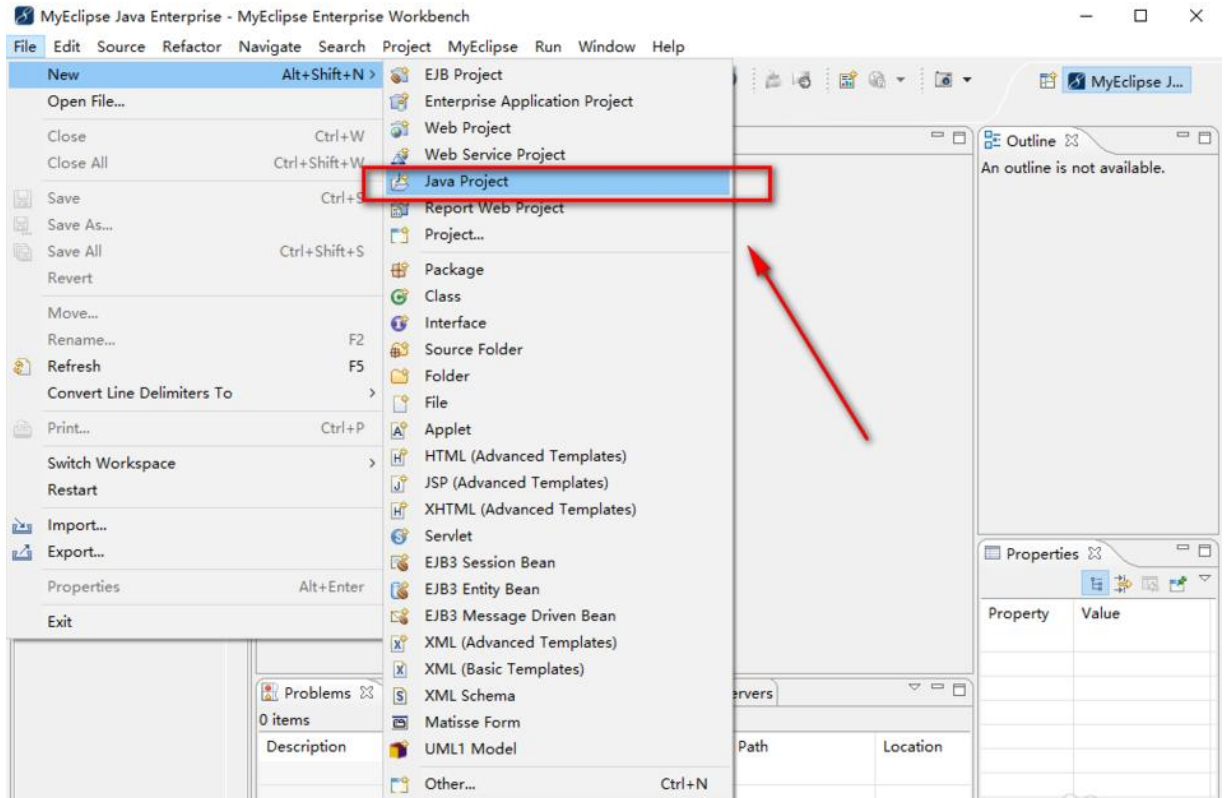
编程实现动物世界的继承关系。

### 【实验要求】

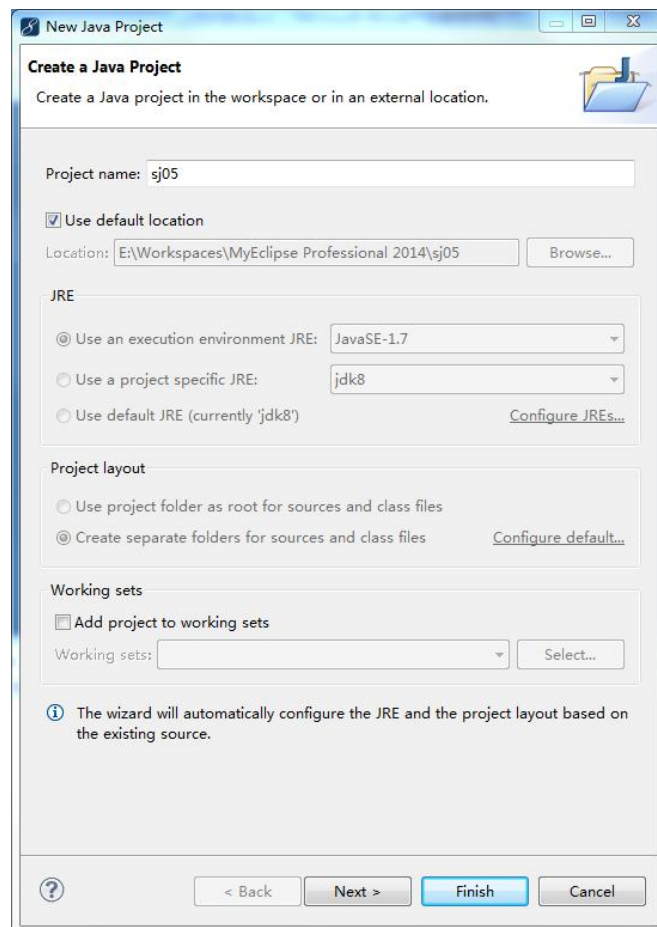
动物(Animal)具有行为：吃(eat)、睡觉(sleep)；动物包括：兔子(rabbit)、老虎(tiger)；这些动物吃的行为各不相同(兔子吃草、老虎吃肉)，但睡觉的行为是一致的。请通过继承实现以上需求，并编写测试类AnimalTest进行测试。

### 【上机指导】

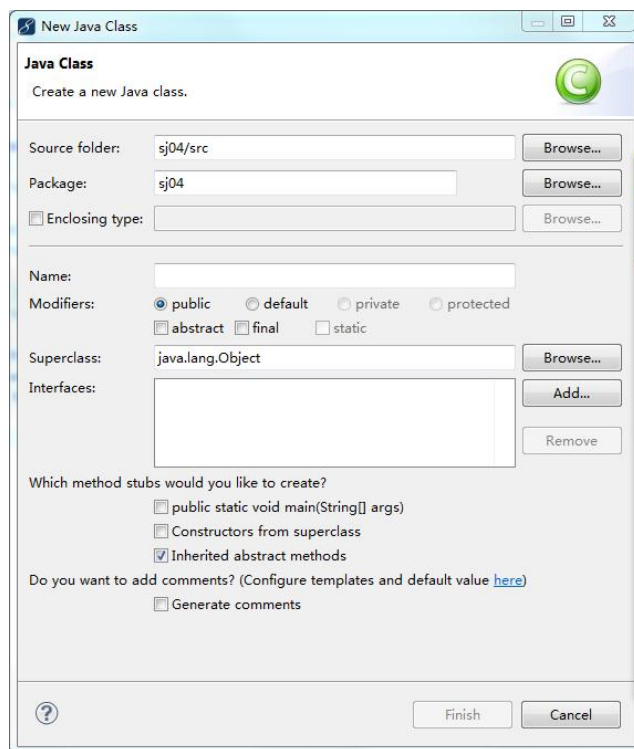
- (1) 单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File” -> “New” -> “Project”，如图：



(3) 为创建的工程起名为sj05, 点“完成”，如图：



(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Animal”，点“完成”；

(5) 在Animal类文件中输入如下代码：

```
public class Animal {  
    void eat()  
    {  
        System.out.println("food! ");  
    }  
    void sleep(){  
        System.out.println("sleep");  
    }  
}
```

(6) 创建rabbit类，并输入如下代码：

```
public class Rabbit extends Animal {  
    void eat()  
    {  
        System.out.println("我是兔子，我吃草！");  
    }  
}
```

(7) 创建tiger类，并输入如下代码：

```
public class Tiger extends Animal {  
    void eat(){  
        System.out.println("我是老虎，我吃肉！");  
    }  
}
```

```
}
```

(8) 创建Test05类，并输入如下代码：

```
public class Test05 {  
    public static void main(String[] args) {  
        Animal a = new Animal();  
        Rabbit r = new Rabbit();  
        Tiger t = new Tiger();  
        a.eat();  
        a.sleep();  
        r.eat();  
        r.sleep();  
        t.eat();  
        t.sleep();  
    }  
}
```

(9) 在” test05”上右键，选择 “Run AS” -> “Java Application”，结果如图：



## 实验三：异常处理

### 【实验目的】

1. 掌握基本异常的处理机制；
2. 熟悉“try”语句与“catch”语句的搭配使用；
3. 掌握声明异常和抛出异常。

### 【实验内容 1】

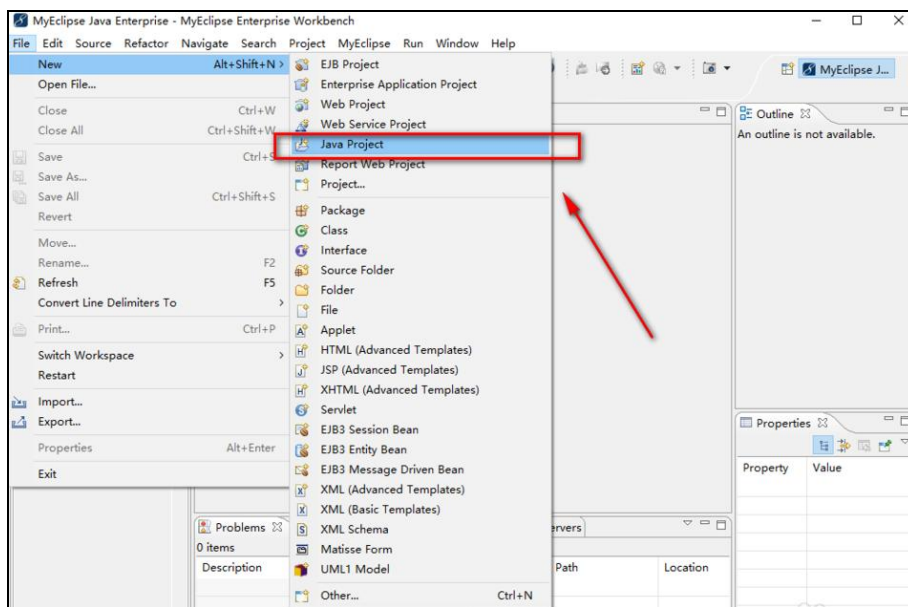
Try catch异常处理。

### 【实验要求】

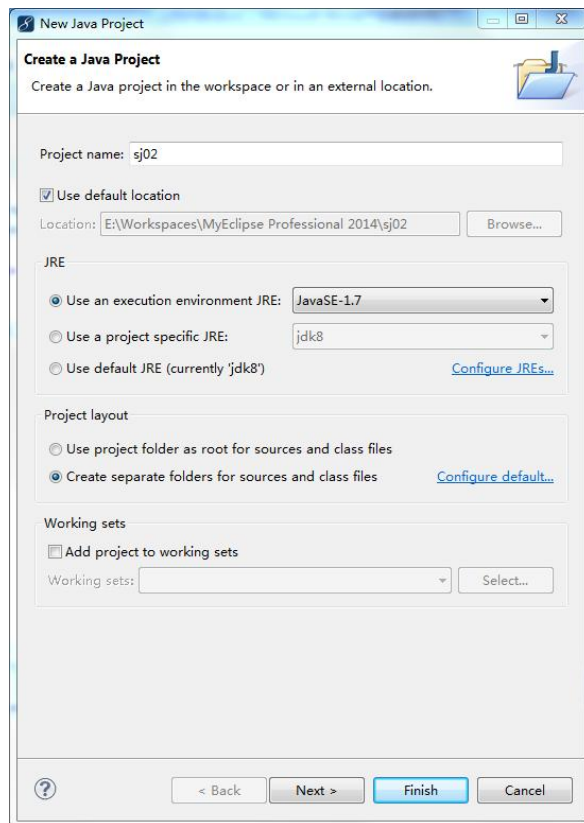
用户输入数字代表天，如果数字值在1-7之间，提示今天是数字对应的星期几，如果输入数字不是1-7之间，则抛出自定义的异常；

### 【上机指导】

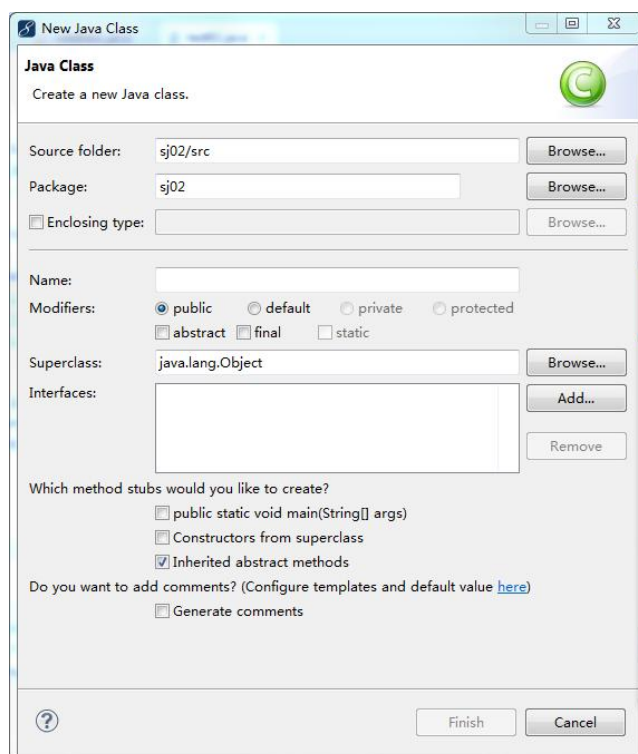
- (1) 单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File” -> “New” -> “Project”，如图：



- (3) 为创建的工程起名为sj02, 点“完成”，如图：



(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Week”，点“完成”；

(5) 在Week类文件中输入如下代码：

```
public class Week {
```

```
String[] days ={"星期一","星期二","星期三","星期四","星期五",
"星期六","星期日"};
String number;
}
```

(6) 创建WeekException类并输入如下代码:

```
public class WeekExcotion extends Exception {
    public WeekExceptio(){
        System.out.println("不符合输入要求! ");
    }
}
```

(7) 创建Test02类并输入如下代码:

```
import java.io.*;
public class Test02 {

    public static void main(String[] args) throws WeekException,
IOException {
        Week w = new Week();//上面定义
        BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
        try{
            System.out.println("请输入要查询的编号 (1~7) : ");
            w.number= in.readLine();
            if(Integer.parseInt(w.number)<1||Integer.parseInt(w.number)>7)
                throw new WeekException();
        } catch (WeekException e) {//上面定义
            e.printStackTrace();
        }

        for (int i = 0; i < 7; i++) {
            if (i == Integer.parseInt(w.number)-1) {
                System.out.println(w.days[i]);
                break;
            }
        }
    }
}
```

(8) 在” test02”上右键, 选择 “Run AS” -> “Java Application”, 结果如图:





(9) 输入数字1, 如图:

请输入要查询的编号 (1~7):

1

星期一

### 【实验内容 2】

Try catch finally异常处理。

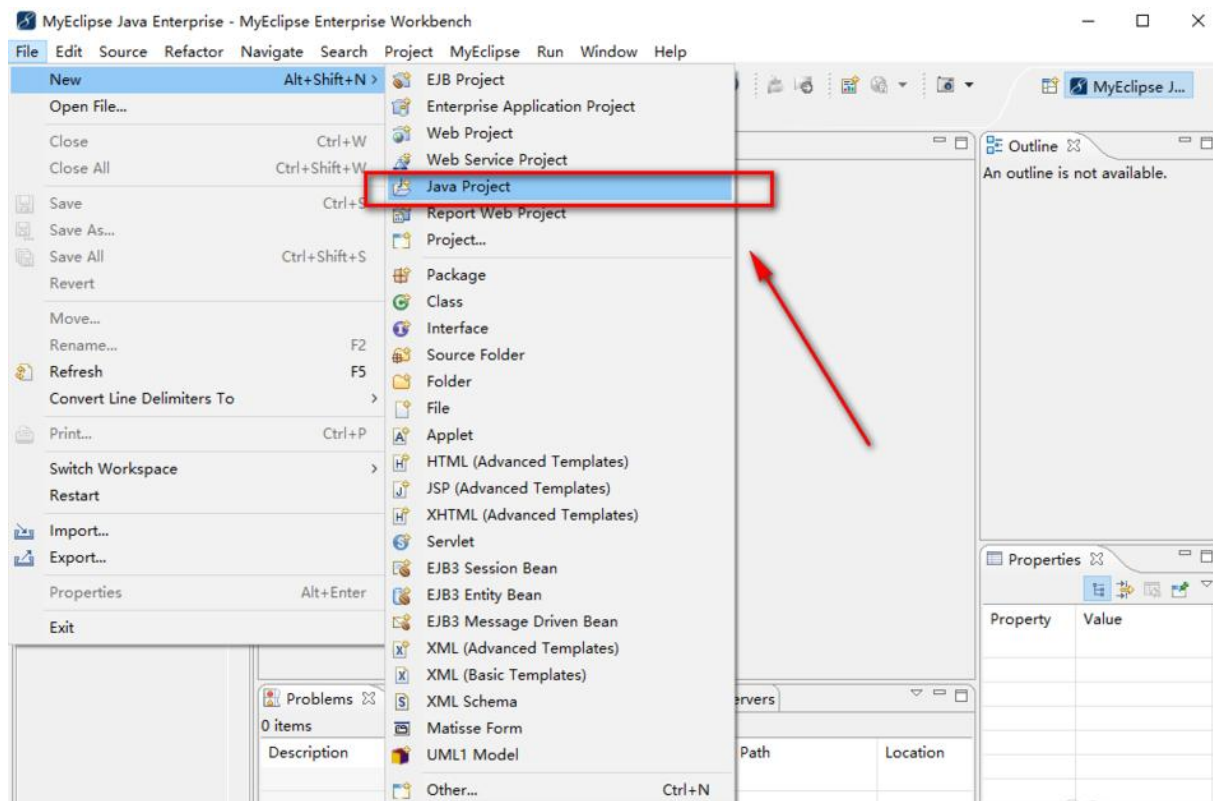
### 【实验要求】

创建一个异常处理的类, 程序中首先输出“这是一个异常处理的例子”, 然后在程序中主动产生一个“ArithmeticException”类型被“0”除的异常, 并用“catch”语句捕获这个异常。最后通过“ArithmeticException”类的对象e的方法“getMessage”给出异常的具体类型并显示出来。finally中给出最后的输出语句。

### 【实验指导】

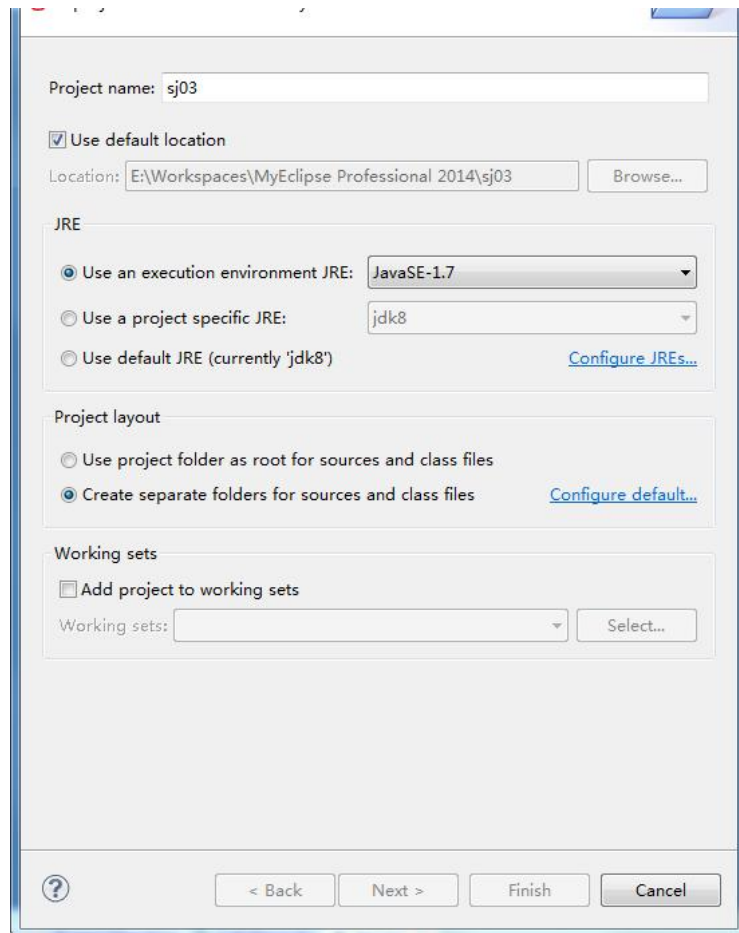
(1) 单击“开始”菜单, 选择“程序”, 选择“MyEclipse”, 单击“MyEclipse”项。

(2) 在MyEclipse中单击“File”->“New”->“Project”, 如图:

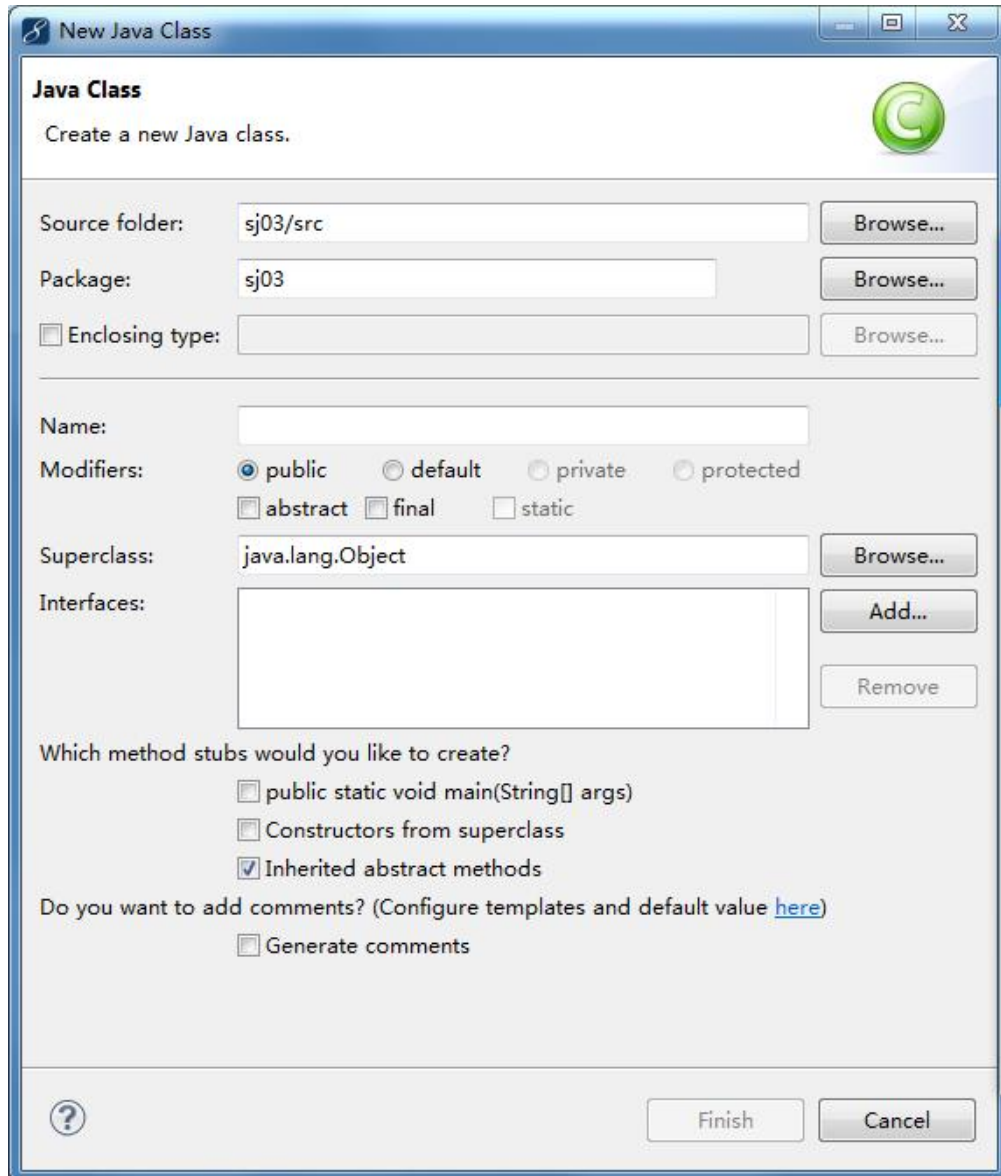


(3) 为创建的工程起名为sj03, 点“完成”, 如图:





(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Test03”，点“完成”；

(5) 在Test03类文件中输入如下代码：

```
import java.io.IOException;
```

```
public class Test03 {
```

```
    public static void main(String[] args) throws IOException {  
        int i=1,j;  
        try{  
            System.out.println("Try:这是一个异常处理的例子: ");  
            j=i/0;  
            throw new ArithmeticException();  
        }catch(ArithmeticException e){
```

```
            System.out.println("Catch:"+e+";"+"\\n"+"reason:"+e.getMessage());
```

```
    }finally{  
        System.out.println("Finally:must go inside finally");  
    }  
}  
  
}
```

(6) 在” Test03”上右键，选择 “Run AS” -> “Java Application”，结果如图：

Try:这是一个异常处理的例子：

Catch:java.lang.ArithmeticException: / by zero;

reason:/ by zero

Finally:must go inside finally



## 实验四：输入输出

### 【实验目的】

1. 熟悉Java的文件读写机制，练习输入输出流的使用。

### 【实验内容 1】

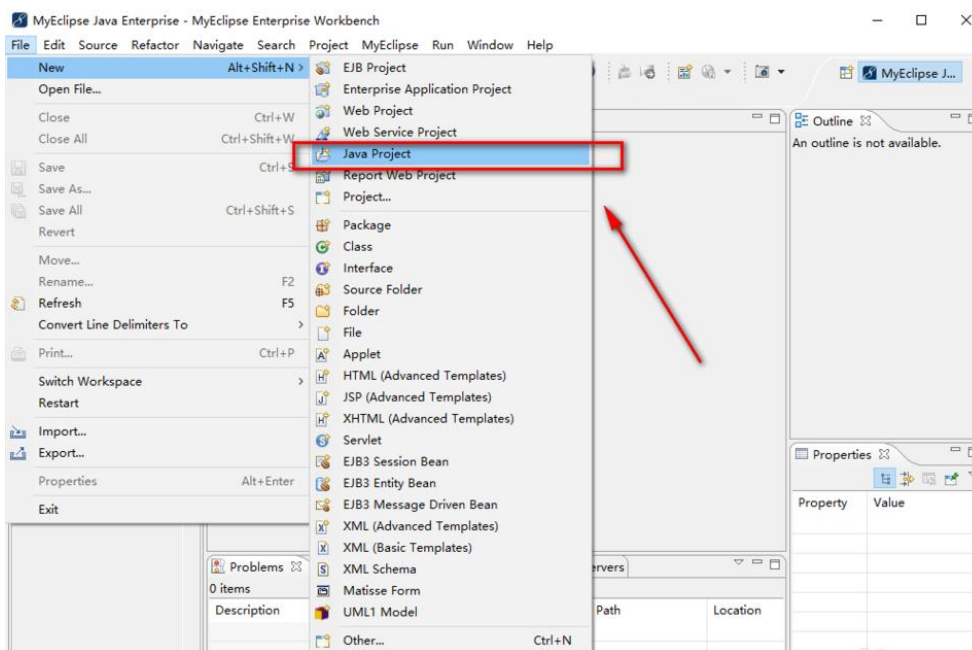
将字符写入文件；

### 【实验要求】

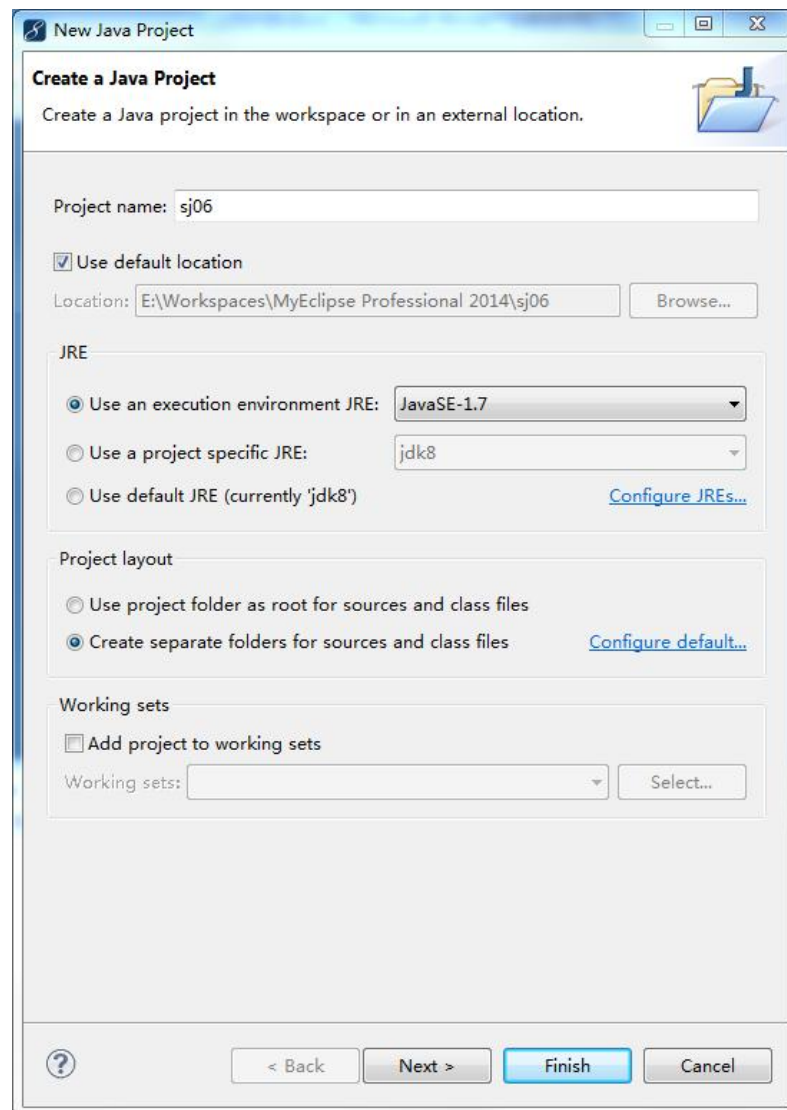
在D盘创建文件test.txt，将字符串“this is test”和“这是一个写入测试”写入到文件中。

### 【上机指导】

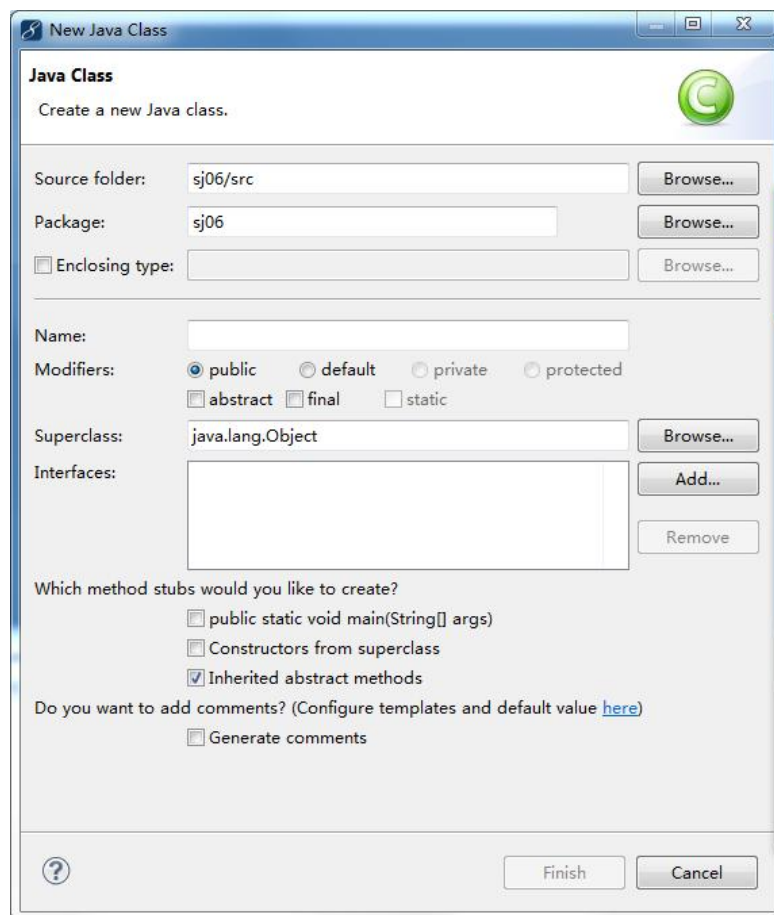
- (1) 单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File” -> “New” -> “Project”，如图：



- (3) 为创建的工程起名为sj06, 点“完成”，如图：



(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Test06”，点“完成”；

(5) 在Test06类文件中输入如下代码：

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
public class Test06 {
    public static void main(String[] args) throws IOException {
        File f =new File("d:\\test.txt");
        BufferedWriter out = new BufferedWriter(new FileWriter(f));

        out.write("This is test");
        out.newLine();
        out.write("这是一个写入测试!");
        out.close();
    }
}
```

(6)在” Test06”上右键，选择“Run AS” -> “Java Application”，结果如图：



### 【实验内容 2】

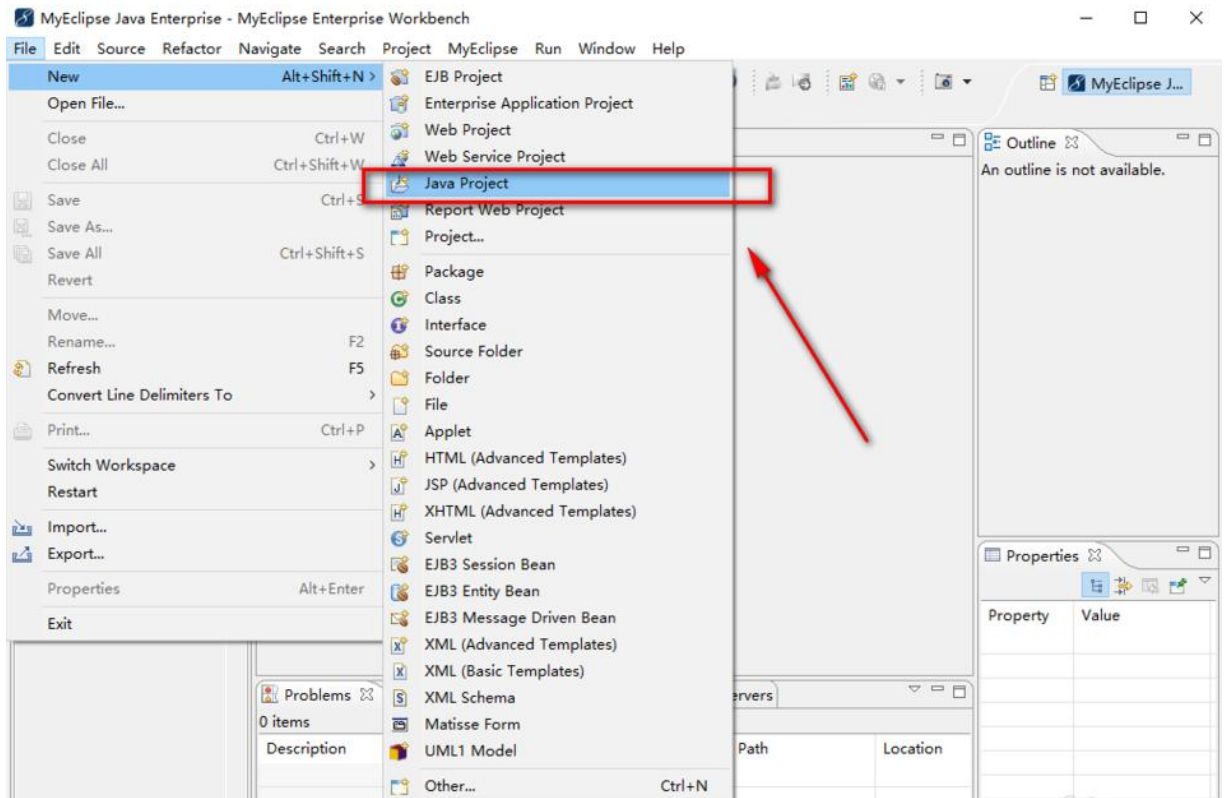
从文件中读取数据

### 【上机要求】

将一个文件test2.txt的内容按行读出，然后写入到文件test3.txt中。

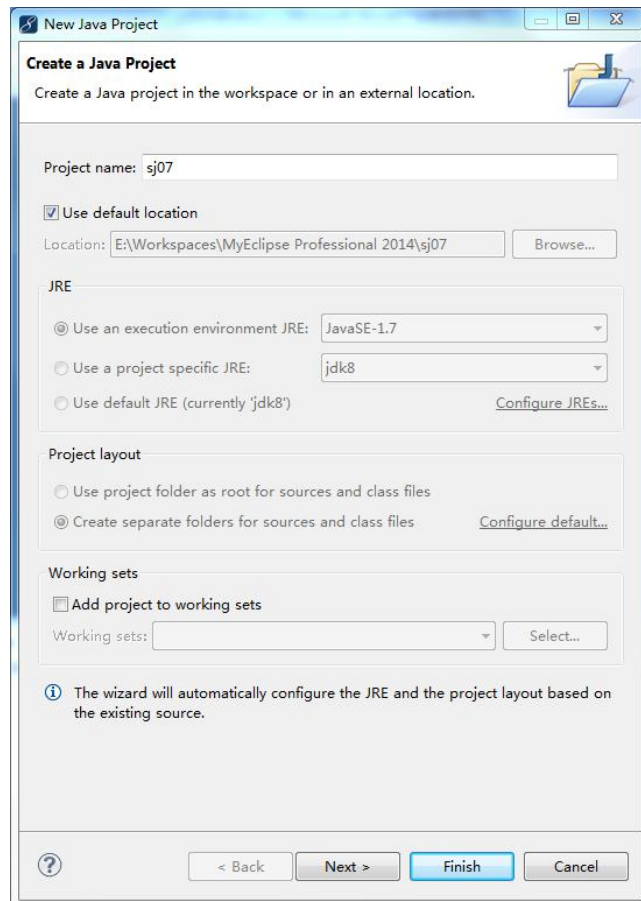
### 【上机指导】

- (1) 单击“开始”菜单，选择“程序”，选择“MyEclipse”，单击“MyEclipse”项。
- (2) 在MyEclipse中单击“File”→“New”→“Project”，如图：

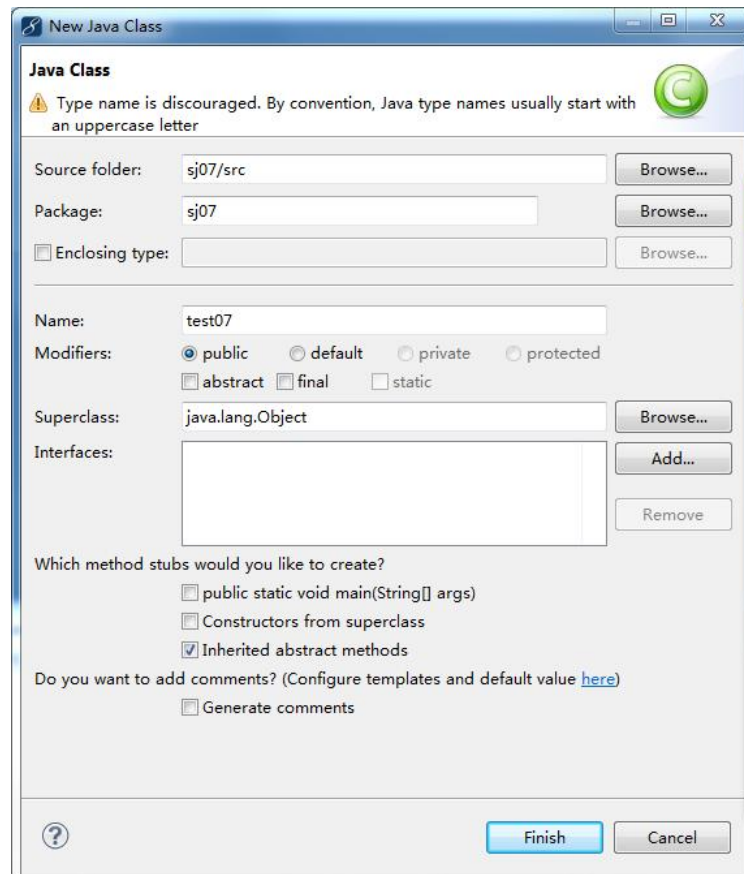


- (3) 为创建的工程起名为sj07,点“完成”，如图：





(4) 展开“Src”，右键点击“Src”，选择“New”->“Class”，后如图：



在“name”中输入“Test07”，点“完成”；

(5) 在Test07类文件中输入如下代码：

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.BufferedWriter;
import java.io.BufferedReader;

public class test07 {
    public static void main(String[] args) throws IOException {

        File f = new File("d:\\test2.txt");
        File F = new File("d:\\test3.txt");
        BufferedWriter out = new BufferedWriter(new FileWriter(f));
        out.write("Hello!");
        out.newLine();
        out.write("Hello, nice to meet you!");
        out.newLine();
        out.close();//先关闭f,再读写进f的内容

        FileInputStream fis = new FileInputStream(f);
        InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader in = new BufferedReader( isr);
        BufferedWriter out2 = new BufferedWriter(new FileWriter(F));
        out2.write(in.readLine());
        out2.newLine();
        out2.write(in.readLine());
        out2.newLine();
        out2.close();
    }
}
```

(6) 在” Test07”上右键，选择“Run AS” -> “Java Application”，结果如图：

