

第 5 章作业-1

(共 100 分)

提交说明：

1. 上传到[坚果云链接中](#) (上传时请勿挂 vpn 或其他梯子, 否则可能上传失败)
2. 文件名严格按照: **学号-姓名-第几次作业.docx**(或 pdf)的方式命名
3. 每次作业在截止日期之前, 若发现提交内容有误, 可以再提交一次。但每次作业最多只能提交两次。第二次提交时在上述文件名后加上“(2)”的字样。
4. 作业提交形式: 单一的 word 或 pdf 文档, 最好用 word, 方便助教留下扣分记录。**不要上传 Eclipse、VScode、IDEA 等项目文件**, 不要压缩打包, 不要以文件夹形式上传。
5. 截止时间: 10 月 8 日 (周日) 晚 23:59。

一、程序阅读题, 写出程序运行的输出结果, 并**进行解释, 解释写在输出结果后, 不要写在代码中** (每题 15 分, 共 60 分)

第 1 题

```
class worker {
    int num;
    worker() {
        num += 5;
    }

    worker(int n) {
        num = n;
    }

    void workShow() {
        System.out.println("Inside worker method, num=" + num);
    }
}

class programmer extends worker {
    int num = 1;
    programmer(int n) {
        num += n;
    }

    void workShow() {
        System.out.println("Inside programmer method, num=" + num + ",
super.num=" + super.num);
    }
}

public class Dispatch {
    public static void main(String args[]) {
        worker a = new programmer(10);
        a.workShow();
    }
}
```

第2题

```
class Fu {
    boolean show(char a) {
        System.out.println(a);
        return true;
    }
}

class Demo extends Fu {
    public static void main(String[] args) {
        int i = 0;
        Fu f = new Fu();
        Fu f2 = new Demo();
        for (f.show('A'); i < 2 || f2.show('B'); f.show('C')) {
            i++;
            f2.show('D');
        }
    }

    @Override
    boolean show(char a) {
        System.out.println(a);
        return false;
    }
}
```

第3题：读程序题与填空

```
class Animal {
    String name;

    Animal(String name) {
        this.name = name;
    }

    public void enjoy() {
        System.out.println("动物" + name + "的叫声.....");
    }
}

class Cat extends Animal {
    String eyesColor;

    Cat(String n, String c) {
        super(n);
        eyesColor = c;
    }

    @Override
    public void enjoy() {
        System.out.println(eyesColor + "的猫" + name + "喵喵喵喵.....");
    }
}
```

```

    }
}

class Dog extends Animal {
    String furColor;

    Dog(String n, String c) {
        super(n);
        furColor = c;
    }

    @Override
    public void enjoy() {
        System.out.println(furColor + "的狗" + name + "汪汪汪汪.....");
    }
}

class Bird extends Animal {
    Bird() {
        super("Poli");
    }

    @Override
    public void enjoy() {
        System.out.println("鸟儿" + name + "叽叽喳喳.....");
    }
}

class AnimalCalls {
    private Animal pet;

    AnimalCalls(Animal pet) {
        this.pet = pet;
    }

    public void myPetEnjoy() {
        pet.enjoy();
    }
}

public class TestCall {
    public static void main(String args[]) {
        Cat c = new Cat("Tom", "蓝色");
        Dog d = new Dog("Ben", "黑色");
        Bird b = new Bird();
        AnimalCalls c1 = new AnimalCalls(c);
        AnimalCalls c2 = new AnimalCalls(d);
        AnimalCalls c3 = new AnimalCalls(b);
        c1.myPetEnjoy();
        c2.myPetEnjoy();
        c3.myPetEnjoy();
    }
}

```

(1)请给出程序运行结果, 并解释结果 (结果 5 分, 解释 5 分)

(2) 方法

```
AnimalCalls(Animal pet) {  
    this.pet = pet;  
}
```

是_____类的_____方法。 (2 分)

(3)该程序中, _____为超类, _____为子类。 (2 分)

(4) c2.myPetEnjoy()调用的是_____类的 enjoy 方法 (1 分)

4. 读程序题与填空: 先补齐程序中的下划线处, 再写出程序在 console 端的输出结果, 并解释。

```
abstract class Person {  
    public Person(String n) {  
        name = n;  
    }  
  
    public abstract String getDescription();  
  
    public String getName() {  
        return name;  
    }  
  
    private String name;  
}  
  
class Employee extends Person {  
    public Employee(String n, double s) {  
        super(____);  
        salary = ____;  
    }  
  
    public double getSalary() {  
        return salary;  
    }  
  
    @Override  
    public String getDescription() {  
        return String.format("an employee with a salary of $%.2f",  
salary);  
    }  
  
    public void raiseSalary(double byPercent) {  
        double raise = salary * byPercent / 100;  
        salary += raise;  
    }  
  
    private double salary;  
}
```

```

class Student extends Person {
    public Student(String n, String m) {
        super(n);
        major = m;
    }

    @Override
    public String getDescription() {
        return "a student majoring in " + major;
    }

    private String major;
}

public class PersonTest {
    public static void main(String[] args) {
        Person[] people = new Person[2];
        people[0] = new Employee("Harry Hacker", 50000);
        (____ people[0]).raiseSalary(10);
        people[1] = new Student("Maria Morris", "computer science");

        for (Person ____ : ____ ) {
            System.out.println(p.getName() + ", " + p.getDescription());
        }
    }
}

```

此题填空部分每空 1 分，共 5 分

输出结果 (5 分) :

解释 (5 分)

二、编程题 (每题 20 分，共 40 分)

1、先定义一个接口 Perarea，它有两个方法，一个是计算图形的面积，另一个是计算图形周长：

```

double get_area();
double get_perimeter();

```

再实现一个长方形和一个圆形的类，它们都实现了这个接口。这两个类实现带参数的构造方法，再分别构造长为 10，宽为 5 的长方形，以及半径为 5 的圆，然后**通过接口**分别打印出这两个图形的面积和周长。**要写出输出结果。**

2、本程序的数值默认都是 double 型。

(1) 编写一个矩形类 Rect, 包含成员:

- a) 两个属性: 矩形的宽 width; 矩形的高 height。
- b) 两个构造方法: 一个带有两个参数, 用于 width 和 height 属性初始化; 一个不带参数, 将矩形初始化为宽和高都为 10。
- c) 两个实例方法: 一个求矩形面积 area(), 另一个求矩形周长 perimeter()。

(2) 通过继承 Rect 类, 编写一个具有确定位置的矩形类 PlainRect, 其确定位置用矩形的左上角坐标来标识。并添加新成员:

- a) 两个属性 (矩形左上角坐标 startX 和 startY)。
- b) 两个构造方法: 一个带 4 个参数, 用于对 startX、startY、width 和 height 属性初始化; 一个不带参数, 将矩形初始化为左上角标为 (0,0)、长和宽都为 10 的矩形。
- c) 一个实例方法: 判断某个点是否在矩形内部 isInside(double x,double y)。
如在矩形内, 返回 true, 否则, 返回 false。 (提示: 点在矩形内需满足条件:
`x >= startX && x <= (startX+width) && y < startY && y >= (startY-height)`)

(3) 编写 PlainRect 类的测试程序

- a) 创建一个左上角坐标为 (10, 10), 长为 20, 宽为 10 的 PlainRect 对象; 计算并打印输出矩形的面积和周长;
- b) 判断点 (25.5, 13) 是否在矩形内, 并打印输出相关信息。