实验8 多线程

班　级:　　软工1805
学　号:　　201806061219
姓　名:　　王程飞
完成日期:　2019.12.23

1.

> 用继承Thread类和实现Runnable接口等两种方式分别编写一个线程程序PrintLetter.java，输出从a到z的26个字母，要求每隔1～2秒钟输出一个字母

代码

```
class PrintLetter extends Thread {
    @Override
    public void run() {
        char a = 'a';
        do {
            try {
                System.out.println(a);
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        } while (a++ < 'z');
    }
}
new PrintLetter().start(); // 继承于Thread
new Thread(() -> {
    char a = 'a';
    do {
        try {
            System.out.println(a);
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    } while (a++ < 'z');
}).start(); // 实现Runnable接口
```

2.

> 编写一个Java程序（包括一个测试线程程序类TestThread，一个Thread类的子类PrintThread）。在测试程序中用子类PrintThread创建2个线程，使得其中一个线程运行时打印5次“线程1正在运行”，另一个线程运行时打印5次“线程2正在运行”

代码

```
class PrintThread extends Thread {
    String name;
    PrintThread(String name) {
        this.name = name;
        this.start();
    }
    @Override
    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println(name + "正在运行");
        }
    }
}
PrintThread t1 = new PrintThread("线程1");
PrintThread t2 = new PrintThread("线程2");
```

3.

> 假设一个银行的ATM机，它可以允许用户存款也可以取款。现在一个账户上有存款200元，用户A和用户B都拥有在这个账户上存款和取款的权利。用户A将存入100元，而用户B将取出50元，那么最后账户的存款应是250元。

代码

```
ATMHelper u1 = new ATMHelper("甲", 100);
ATMHelper u2 = new ATMHelper("乙", -50);
try {
    u1.start();
    u2.start();
} catch (Exception e) {
    e.printStackTrace();
}


class ATMHelper extends Thread {
    static int m = 200;

    int change;
    String name;

    ATMHelper(String name, int change) {
        this.name = name;
        this.change = change;
    }

    synchronized void change() throws Exception {
        System.out.println(name + ": " + change);
        if (change + m < 0) {
            throw new Exception("余额不足");
        } else {
            m = change + m;
        }
    }

    @Override
    public void run() {
        try {
            change();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```