

主管  
领导  
审核  
签字

题号	一	二	三	四	五	六	七	八	九	十	总分
得分											
阅卷人											

片纸鉴心 诚信不败

授课教师

姓名

学号

院系

密

封

线

请大家阅读下面的代码，之后回答问题。（80 分）

/\*需求简要描述：\*/  
/\*疫情期间需要记录核酸检验的结果和日期，为此设计了护士类（Nurse）和检验记录类（Record）。\*/  
/\*每次检验会生成检验记录对象（Record），每名护士（Nurse）维护被其检验的人员集合（Set<String>），检验记录集合（List<Record>），所有的检验记录需按日期先后顺序存储。\*/  
/\*每名护士具有姓名（String）属性，且不同护士之间的姓名不能相同\*/  
/\*检验记录（Record）中记录了被检验的人员姓名（String）、检验时间（Date）和检验结果（boolean）。\*/  
/\*护士之间可以相互代班，每名护士拥有一个列表（Set<Nurse>），该列表中存储了能够代替当前护士上班的其他护士，护士之间的代班是相互的，即如果 A 护士能够代班 B 护士，那么 B 护士即能代班 A 护士。\*/  
/\*护士姓名不能为空，且为英文字符，姓在前名在后，姓和名的首字母大写，如：LiXiaoming 或 LiMing，被检验人姓名的规则同护士姓名。\*/

```
1 public class Nurse {  
  //rep  
2   private final String name;  
3   private final Set<Nurse> switches;  
4   private final Set<String> persons;  
5   private final List<Record> records;  
  
  // methods  
  //构造函数  
6   public Nurse (string name) {  
7     this.name=name;  
8     this.switches=new HashSet<>();  
9     this.persons=new HashSet<>();
```

---

```
10  this.records=new ArrayList<>();
11  }

//增加一条检验记录，同时将被检验人增加到 persons 集合中
12  public void insertRecord (Record one) {
13      this.records.add (one);
14      this.persons.add (one.getName());
15      //利用 Collections 类提供的 sort 方法对 records 进行排序
16      .....
17  }

//根据被检验人姓名和检验时间删除相应的检验记录
18  public void removeRecord (String name, Date time) {
19      Iterator iter=this.records.iterator ();
20      while (iter.hasNext ()) {
21          Record r=iter.next ();
22          int compareTo = r.getDate.compareTo(time)
23          if (r.getName().equals (name) && compareTo){
24              iter.remove ();
25              break;
26          }
27      }

//按输入的被检验人姓名和日期获取检验记录
28  public Record getOneRecord (String name, Date time) {
29      Iterator iter=this.records.iterator ();
30      while (iter.hasNext ()) {
31          Record r=iter.next ();
32          int compareTo = r.getDate.compareTo(time)
33          if (r.getName().equals (name) && compareTo)
34              return r;
35      }
36  }

//按输入的条形码获取检验记录
37  public Record getOneRecordbyLabel (String label) {
38      Iterator iter=this.records.iterator ();
39      while (iter.hasNext ()) {
40          Record r=iter.next ();
41          if (r.getLabel.equals (label)
42              return r;
43          }
44  }
```

授课教师

姓名

学号

院系

```

..... //将一名护士加入到当前护士的代班列表中
45  public void addSet (Nurse other){}
46      this.switches.add(other);
47      other.addSet(this);
48  }

..... //将一名护士从当前护士的代班列表中删除
49  public void deleteSet (Nurse other){
50      this.switches.remove(other);
51      other.deleteSet(this);
52  }

..... //通过姓名判断两名护士是否是同一人
53  public boolean equals (Object other){
54      Nurse a=(Nurse) other;
55      if (this.nurse.name.toLowerCase( ).equals (a.name))
56          return true;
57      else
58          return false;
59  }

..... //haseCode 方法
60  public int hashCode( ){
61      return this.name.length( );
62  }

..... // 利用正则表达式判断字符串是否符合一条标准的检验记录。
63  public boolean read (String content){
64      String regEx = "*****";
65      Pattern pattern= Pattern.compile(regEx);
66      Matcher m = pattern.matcher(content);
67      if (m.matches( ))
68          return true;
69      else
70          return false;
71  }
72 }

```

---

//检验记录类，记录了被检验人，检验日期，检验结果和检验记录条形码，其中检验  
//结果为布尔值，“Y”代表结果阳性，“N”代表结果阴性。

```
73 public class Record{
    //rep
74     public final String person;
75     public final Date time;
76     public final boolean result;
77     public final String label;

    //methods
78     public Record(String name, Date time, boolean result){
79         this.person=name;
80         this.time=time;
81         this.result=result
82         this.label=generateLabel();
83     }
84     public String getName (){
85         return this.person;
86     }
87     public Date getDate (){
88         return this.time;
89     }
90     public String generateLabel (){
        //省略了生成标签的方法
91         .....
92     }
93 }
```

授课教师

姓名

学号

院系

密

封

线

1、请根据需求以注释的形式给出 Nurse 类的 RI（表示不变性）。（8 分）

2、请分析上述 Record 类中的代码中是否存在表示暴露（Rep Exposure）缺陷，如果存在，请指出具体位置（代码行数），并予以修改。（8 分）

3、客户端按照下面的代码运行后，请给出运行后的 snapshot diagram 图示。（10 分）

```
nurse1=new Nurse ("LiMing");
nurse2=new Nurse ("WangQiang");
record1=new Record ("ZhangLing", new Date("2020-06-19"), false);
record2=new Record ("LiuShui", new Date("2020-06-22"), true);
nurse1.insertRecord (record1);
nurse2.insertRecord (record2);
nurse1.addSet(nurse2);
```

4、请判断代码中实现的判相等方法（53-59）是否合理，并给出理由，如不合理，请给出你认为合理的判相等方法（只需给出思路即可，无需给出具体代码）。（8 分）

5、为防止多线程使用 Nurse 类时出现错误，可以使用 monitor 设计模式为类 Nurse 加锁，即在该类的每个方法描述前加入 synchronized 关键字。在使用 monitor 设计模式后，假设两个线程类，TreadA 和 TreadB，分别调用两个 Nurse 对象（nurse1 和 nurse2）的 addSet（45-48 行）和 deleteSet（49-52 行）操作，请问在执行时是否会出现问题，如出现请给出至少两种解决策略。（8 分）

```
nurse1=new Nurse ("LiMing");
nurse2=new Nurse ("WangQiang");
Tread treadA= new TreadA();
Tread treadB= new TreadB();
//Thread A 执行下列代码
nurse1.addSet(nurse2);
nurse1.deleteSet(nurse2);
```

```
//Tread B 执行下列代码
nurse2.addSet(nurse1);
nurse2.deleteSet(nurse2);
```

---

6、为对检验记录（Record）进行更好的保管，系统会自动为每个检测记录生成条形码（代码行 90-92），条形码的组成规则如下：

- 1) 条形码由两部分组成，分别对应护士姓名和随机生成的编码，中间用“\_”分割；
  - 2) 护士姓名为英文字符，姓在前名在后，姓和名的首字母大写，如：“LiXiaoming”或“LiMing”，且不为空；
  - 3) 随机生成的编码由英文小写字符和数字混合而成，共 8 位，首位必须为英文字符，其它位为英文或数字；
- 请根据上述规则为 Nurse 类中的方法 `getOneRecordbyLabel` 撰写测试策略 `testing strategy`，并给出测试用例。（12 分）

7、Nurse 类中的 `read` 方法（63-71 行）可利用正则表达式判断输入的字符串（`content`）是否是一条标准的检验记录。检验记录的形式如题目 6 所示。请在下面给出正确的正则表达式 `regEx`。（8 分）

8、检验记录集合（`List<Record>`）中所有的检验记录需按日期先后顺序存储，故此需要实现检验记录排序方法。考虑到职责分配原则，Nurse 类中不希望实现对 Record 对象判断先后次序的方法，而是直接调用 Collections 类提供的 `sort` 函数对 `List<Record>` 进行排序（15 行）。故此，请给出使 Record 对象能够比较大小的设计方案。（8 分）

9、代码中的 `read` 方法（63-71 行）利用正则表达式判断输入的字符串（`content`）是否是一条标准的检验记录。此种方式读取效率有限，故而可实现 `read` 方法的一个重载方法（`overload`），用于从文件中读取连续字符串。考虑从文件中读取字符串时会出现某些异常情况，请为可能的异常编写异常类，用于提示用户异常发生的原因，例如护士姓名首字母非大写，请至少给出两种异常情况下的异常类。

请根据上述需求：

- 1) 首先设计重载方法（只需给出方法的声明部分）；（4 分）
- 2) 给出可能发生的异常类的定义和实现。（6 分）

授课教师

姓名

学号

院系

10、简答题。(20 分)

1) 请对比 SVN 和 GIT 在存储文件方面的区别。(3 分)

2) 请分别对比 exception 和 error 的区别, 以及 checked exception 和 unchecked exception 的区别。(3 分)

3) 判断以下代码是否符合 LSP 原理, 如不符合请给出原因。(4 分)

```
class Rectangle {  
    // invariant h>0 && w>0;  
    int h, w;  
    Rectangle(int h, int w) {  
        this.h=h;  
        this.w=w;  
    }  
    // requires factor > 0;  
    void scale(int factor) {  
        w=w*factor;  
        h=h*factor;  
    }  
}  
class Square extends Rectangle {  
    // invariant h>0 && w>0;  
    //invariant h==w;  
    Square(int w) {  
        super(w, w);  
    }  
}
```

---

4) 请指出下面给出的类是否是 `immutable` 的, 如果不是请给出具体原因和修改策略。(5 分)

```
public myString{
    public final char a[ ];
    myString(char[ ] input){
        a=input;
    }
    public void length ( ){
        return a.length;
    }
    public myString void getstring ( ) {
        return a;
    }
    public char void getchar (int i)
    {
        if (i<this.length)
            return a[i];
    }
}
```

哈工大网盘计划  
密码1920



5) 观察 `Animal`、`Duck` 和 `Penguin` 类, 之后请分别给出后续代码的运行结果。(5 分)

```
public Animal{
    public void fly( ){
        System.out.println("fly in 100 meters");
    }
    public void run (int p){
        System.out.println("run in 100 meters");
    }
}
public Duck extends Animal{
    public void fly( ){
        System.out.println("fly in 200 meters");
    }
    public void run (string p){
        System.out.println("run in 200 meters");
    }
}
public Penguin extends Duck{
    public void run (int p){
        System.out.println("run in 300 meters");
    }
}
```



授课教师

姓名

学号

院系

密

封

线

```
: Animal a = new Duck( );  
: Duck b=new Duck( );  
: Animal c=new Penguin( );  
: a.run();  
: b.fly( );  
: c.fly( );  
: c.run(100);  
: c.run("100");
```



