

**2021年春季学期  
计算学部《软件构造》课程**

**课程报告**

|  |  |
| --- | --- |
| 姓名 |  |
| 学号 |  |
| 班号 |  |
| 电子邮件 |  |
| 手机号码 |  |

已知如下业务背景：

1. 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）。
2. 每次检验会生成检验记录（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。
3. 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。
4. 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）。
5. 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。
6. 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。

针对此业务背景的部分代码如下：

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( ));

//利用Collections类提供的sort方法对records进行排序

15 ……….

16 }

//根据被检验人姓名和检验时间删除相应的检验记录

17 public void removeRecord (String name, Date time) {

18 Iterator iter=this.records.iterator ( );

19 while (iter.hasNext ( )){

20 Record r=iter.next ( );

21 int compareTo = r.getDate.compareTo(time)

22 if (r.getName( ).equals (name) && compareTo){

23 iter.remove ( );

24 break;

25 }

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 }

//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 }

1. 请根据需求以注释的形式给出Nurse类的RI（表示不变性）
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。
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。

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);

1. 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（给出理由和修改后的代码）
2. 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：

* 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；
* 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”；
* 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；

请跟据上述规则为Nurse类中的方法getOneRecordbyLabel撰写测试策略 testing strategy ，并给出测试用例。

1. Nurse类中的 read方法（63-72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5所示。请在下面给出正确的正则表达式 regEx。
2. 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。
3. 代码中的read方法（63-72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，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；

2）给出可能发生的异常类的定义和实现。

简答题：

1. 请对比SVN和GIT在存储文件方面的区别。
2. 请分别对比exception和error的区别，以及checked exception和unchecked exception的区别。
3. 判断以下代码是否符合LSP原理，如不符合请给出原因。

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);

}

}

1. 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。

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];

}

}

1. 观察Animal、Duck和Penguin类，之后请分别给出后续代码的运行结果。

public Animal{

void fly( ){

System.out.println(“fly in 100 meters”);

}

void run (int p){

System.out.println(“run in 100 meters”);

}

}

public Duck extends Animal{

void fly( ){

System.out.println(“fly in 200 meters”);

}

void run (string p){

System.out.println(“run in 200 meters”);

}

}

public Penguin extends Duck{

void run (int p){

System.out.println(“run in 300 meters”);

}

}

Animal a = new Duck( );

Duck b=new Duck( );

Animal c=new Penguin( );

a.run( );

b.fly( );

c.fly( );

c.run(100);

c.run(“100”);

注：请自行根据所回答内容的多少调整题与题之间的间距。

报告请于18周周日之前通过邮件发送给授课教师，报告命名方式为“学号-姓名-软件构造2021春课程报告”，各授课教师邮件地址如下：

徐汉川 [xhc@hit.edu.cn](mailto:xhc@hit.edu.cn)

陈惠鹏 [chp@ir.hit.edu.cn](mailto:chp@ir.hit.edu.cn)

刘铭 liuming1981@hit.edu.cn