主管领导审核签字

哈尔滨工业大学 2018 学年春季学期

软件构造　　试题

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

片纸鉴心 诚信不败

授课教师_____　密　姓名_____　封　学号_____　线　院系_____

一　单选题（每题 2 分，共 30 分。请将全部答案填写至下表中，否则视为无效）

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	

1. Memory dump 属于软件三维度视图的_____

A　Build-time 和 code-level view　　B　Run-time 和 period view

C　Run-time 和 moment view　　D　Moment 和 code-level view

2. 以下关于软件配置管理 SCM 和 Git 的说法，不正确的是____

A　软件配置项 SCI 是软件演化过程中发生变化和 SCM 管理变化的基本单元，不需再细分

B　Git 中在本地机器上的.git 目录对应于 SCM 中的配置管理数据库 CMDB

C　Git 中的 SCI 是“文件”，它有三种形态：已修改(modified)、已暂存(staged)、已提交(committed)

D　Git 中两次相邻提交 v1 和 v2，若后者提交时间晚于前者，那么 Git 仓库中只记录 v2 中的文件相对于 v1 中的文件发生变化的代码行（增加和删除的代码行）。

3. 以下四个代码段在编译和执行时的结果分别是_____

```
int[] arr = new int[] {1,2};
arr[2] = 3;
-----------
int[] arr = new int[] {1,2};
arr[0] = "3";
-----------
String s = null;
System.out.println(s == null);
```


草　纸

（草纸内不得答题）

```
-----------
String s = null;
System.out.println(s.length);
```

A 编译错误 / 运行时错误 / 无错误 / 无错误
B 运行时错误 / 编译错误 / 无错误 / 运行时错误
C 编译错误 / 运行时错误 / 运行时错误 / 运行时错误
D 运行时错误 / 编译错误 / 编译错误 / 无错误

4. 函数 boolean hasSameCharset (String str1, String str2)用于判定两个字符串 str1 和 str2 是否包含了相同的字符集合。例如，"HIT"和"HITTIH"都且仅包含 H、I、T 三个字符，但将"HIT"和"MIT"作为参数输入该函数则返回 false。以下给出了该函数的 3 个不同 spec，其中 spec 强度最弱的是_____

第 1 个 *requires*: *effects*:	 str1 和 str2 的长度均>0 若 str1 包含的字符集合与 str2 包含的字符集合相同，则返回 true，否则返回 false
第 2 个 *requires*: *effects*:	 str1 和 str2 的长度均>0，且均不包含重复字符 若 str1 包含的字符集合与 str2 包含的字符集合相同，则返回 true，否则返回 false
第 3 个 *requires*: *effects*:	 无 若 str1 或 str2 为空，则抛出 NullParameterException；否则，若 str1 包含的字符集合与 str2 包含的字符集合相同，则返回 true，否则返回 false

A 第 1 个　　B 第 2 个　　C 第 3 个　　D 无法判断

5. 类 WordList 有以下四个方法，仅根据其方法定义来确定其类型，最有可能分别是_____

```
public WordList(List<String> words)
public void unique()
public WordList getCaptitalized()
public Map<String, Integer> getFrequencies()
```

注：Creator (C), Mutator (M), Producer (P), Observer (O)

A C / O / M / O　　B C / M / P / O
C P / O / C / P　　D C / M / O / O

6. 以下关于 ADT 的 RI 和 AF 的说法，不正确的是_____

A ADT 的 Abstract 空间(A)中的某个值，在其 Rep 空间(R)中可能有多个值与其对应
B 如果 ADT 的某个方法返回一个 mutable 的对象，那么表明该 ADT 产生了表示泄露
C 如果 ADT 的任意 constructor 所构造出的 object 都满足 RI、每个 mutator 方法执行结束后都保持 RI 为真，那么该 ADT 的 RI 就永远保持为真
D 一个 immutable 的 ADT，其 rep 可以是 mutable 的

姓名_____ 学号_____ 院系_____ 授课教师_____ 密 封 线

草　纸

（草纸内不得答题）

密

封

线

授课教师_____

姓名_____

学号_____

院系_____

7. 某个 ADT Person 中 override 了 equals()方法，该方法违反了____特性（注：name 是 String 类型，id 为 int 类型，二者均为 Person 的成员变量；Math.abs()为计算绝对值的方法，String.toLowerCase()将字符串中的所有字符变为小写）

```
public boolean equals(Object o) {
    if(o == null) return false;
    if(! (o instanceof Person) ) return false;
       Personp = (Person) o;
       if( p.name.toLowerCase().equals(this.name) && Math.abs(p.id - this.id)<=10)
          return true;
    return false;
}
```

A 传递性　　B 自反性、传递性　　C 自反性、传递性、对称性　　D 都未违反

8. 以下关于 Liskov Substitution Principle 的说法，不正确的是_____

A 子类型中 override 父类型中的方法，子类型的该方法需具有等价或更强的 post-condition、等价或更弱的 pre-condition

B 子类型的表示不变量 RI 应强于或等价于父类型的 RI

C 某个类是 immutable 的，它可以派生出一个 mutable 的子类

D 如果父类型中的某个方法声明中未 throws 任何 checked 异常，那么子类型中 override 该方法时，在 override 方法内部抛出的任何异常必须被捕获并处理

9. 某个类 Y 中有一个方法

```
public Number getAvgScore(ArrayList<?> scores) throws IOException
```

Y 有一个子类 B，B 中 override 了上述方法。以下是四种可能的 override 方案，其中___是符合 LSP 和 Java 编程语言要求的合法方案

A `private Number getAvgScore(ArrayList<?> scores) throws IOException`

B `public Double getAvgScore(ArrayList<?> scores) throws IOException`

C `public Number getAvgScore(ArrayList<?> scores) throws Exception`

D `public Number getAvgScore(List<?> scores)`

10. 类 A 有一个静态成员变量 int a 和一个非静态成员变量 Date b，它的某个方法内使用了一个局部变量 Calendar c。Java 虚拟机在为 a、b、c 三个变量分配内存时，分别在_____中分配

A stack / stack / heap　　B stack / heap / stack

C heap / stack / stack　　D heap / heap / heap

11. Java 中针对 heap 的四种 GC 策略，以下说法不正确的是_____

A Mark-sweep 检查内存中各对象是 live 还是 dead，对 dead 对象做出标记，进而将其清理

B 每次进行 Mark-compact 策略的 GC 之后，heap 中可被再分配的区域是连续的

C Copy 是四种策略中唯一不需要检查对象 live/dead 的 GC 策略

D Reference counting 可以使内存中 dead 对象的 zombie time 最短

草　纸

（草纸内不得答题）

密

封

线

姓名_____ 授课教师_____

学号_____

院系_____

12.针对以下代码，如果执行时抛出了 IOException，那么其执行路径可能是_____

```
1    FileReader reader = null;
2    try {
         // constructor may throw FileNotFoundException
3        reader = new FileReader("someFile");
4        int i=0;
5        while(i != -1){
             //reader.read() may throw IOException
6            i = reader.read();
7            System.out.println((char) i);
         }
8     } catch (FileNotFoundException e) {
9        e.printStackTrace();
10   } catch (IOException e) {
11       e.printStackTrace();
12   } finally {
13       if(reader != null){
14           try {
15              reader.close();
16           } catch (IOException e) {
17               e.printStackTrace();
             }
         }
     }
18   System.out.println(“End”);
```

A 3-10-11-12-13-14-15-18

B 3-4-5-6-10-11-12-13-14-15-18

C 3-4-5-6-7-12-18

D 3-4-5-6-7-10-11-12-13-14-15-16-17-18

13.某个 Java 程序正在运行，其进程号为 4392，在命令行使用 jmap –heap 4392，得到的输出中包含以下部分内容。据此可以计算：Java 为该程序分配的 old generation space 的最大尺寸为 hs，当 old generation space 被占用的内存尺寸达到 ss 时 JVM 即对其进行扩展。那么，hs 和 ss 的取值分别为_____

```
Heap Configuration:
   MinHeapFreeRatio = 20
   MaxHeapFreeRatio = 80
   MaxHeapSize      = 256MB
   NewSize          = 16MB
   MaxNewSize       = 64MB
   OldSize          = 48MB
   NewRatio         = 3
   SurvivorRatio    = 8
   ...
```

草　纸

（草纸内不得答题）

A 48MB，12.8MB　　B 64MB，51.2MB　　C 192MB，38.4MB　　D 256MB，67.2MB

14.两个线程的 run()的代码如下所示，可能存在死锁的是___

Thread 1	Thread 2

synchronized (a) {
  synchronized (c) {
  // ...
  }
  synchronized (b) {
  // ...
  }
}
	synchronized (b) {
	  synchronized (c) {
	  // ...
	  }
	}
	synchronized (c) {
	  synchronized (a) {
	  // ...
	  }
	}

A a 和 b　　B a 和 c　　C b 和 c　　D 不存在死锁

15.当执行以下代码的 main()之后，控制台的输出内容不可能是_____

```
public class B extends Thread {
    public void run() {
       try {
          Thread.sleep(200);
          System.out.print("a");
       } catch (InterruptedException e) {
          System.out.print("b");
       }
       System.out.print("c");
    }
   public static void main(String args[]) throws InterruptedException {
       B b = new B();
       b.start();
       Thread.sleep(200);
       b.interrupt();
    }
}
```

A ac　　B bc　　C ab　　D abc

姓名_____ 授课教师_____

学号_____

院系_____

密 封 线

草　纸

（草纸内不得答题）

密 封 线

授课教师_____ 姓名_____ 学号_____ 院系_____

二 简答题（共 20 分）

1. ADT 泛型（10 分）

```
1  public class Bag<E> {
2     private Map<E, Integer> elements;

      /** 创建一个空的 bag */
3     public Bag() {  elements = new HashMap<>(); }

      /** 修改 elements，在其中增加 1 个元素 e */
4     public Bag<E> add(E e) {
5        if (elements.get(e) == null)
6           elements.put(e, Integer.valueOf(1));
7        else {
8           int count = elements.get(e).intValue();
9           elements.put(e, Integer.valueOf(count + 1));
10       }
11       return this;
12    }

      /**  修改 elements，将 e 的出现次数减 1，若次数变为 0 则从中删除 e */
13    public Bag<E> remove(E e) {
14       if (elements.get(e) == null)
15          return this;
16       else {
17          int count = elements.get(e).intValue();
18          if (count == 1)  elements.remove(e);
19          else elements.put(e, Integer.valueOf(count - 1));
20       }
21       return this;
22    }

      /** 返回 e 在该 bag 中出现的次数*/
23    public int count(E e) {
24       if (elements.get(e) == null) return 0;
25       return elements.get(e).intValue();
26    }
27 }

28    public static void main(String[] args) {
29       Bag<Number> b1 = new Bag<Number>().add(1).add(3.14).add(4L);
30       Bag<Object> b2 = new Bag<Object>().add(1).add("a");
31       Bag<Integer> b3 = new Bag<Integer>().add(1).add(3);
32       Bag<Number> b4 = b3.remove(2);
33       b2.add(new Bag<String>());
34       b3.add(3.14).remove(1);
```


草　纸

（草纸内不得答题）

```
35        System.out.println( sum(b1) );
36        System.out.println( sum(b2) );
37        System.out.println( sum(b3) );
38    }
```

观察代码 29-37 行，各行语句是否可以正确通过编译并正确执行？请指出有错误的行并解释原因。

2．线程安全（10 分）

以下给出了某 ADT 的一部分代码，请判断其是否 threadsafe。若非，请将其修改为 threadsafe 的代码。答题时，请先指出有潜在 threadsafe 风险的行号或行号范围，然后阐述如何修改以消除所有潜在风险，并尽可能保证高并发性能。最后以注释的形式撰写该 ADT 的线程安全策略。

```
1  public class Poem {

2      public String title = new String();
3      private List<String> originPoem = new ArrayList();
4      private final List<String> poem
                                = Collections.synchronizedList(originPoem);

5      public void removeSomeLines(String filter) {
6         Iterator<String> iter = poem.iterator();
7         while (iter.hasNext()) {
8            String line = iter.next();
9            if (line.contains(filter))
10               iter.remove();
11        }
12        title = title.toUpperCase();
13    }
```

姓名_____ 授课教师_____

学号_____

院系_____

密 封 线

草　纸

（草纸内不得答题）

```
14     public String toString() {
15        StringBuilder sb = new StringBuilder(title + "\n");
16        for (String s : poem)
17           sb.append(s + "\n");
18        return sb.toString();
19     }
        ...
20  }
```

草　纸

（草纸内不得答题）

密

封

线

授课教师_____

姓名_____

学号_____

院系_____

第三至七题为综合设计题（共 50 分）

背景描述：

- 教师给学生发布了一组论文，要求每个学生选择不少于 1 篇、不多于 4 篇论文加以阅读并做报告。
- 由于每篇论文只能被 1 个学生选择，故多个学生选择论文时可能产生冲突。如果某个论文已被选择，则其他学生不能再选它。如果某个学生已经选择了至少 1 篇论文，那么他不能再次选其他论文。
- 论文分为学位论文 Thesis、期刊论文 JournalPaper 和会议论文 ConferencePaper 三类。
- 学生可设定自己对论文类型的偏好和待选论文数量的偏好，例如：希望选择 JournalPaper 1 篇。

设计了一组 ADT：论文 Paper（及其三个子类型 Thesis、JournalPaper、ConferencePaper）、教师发布的论文清单 PaperList、学生论文选择结果 PaperSelection、学生 Student。Client.java 是客户端程序。它们的代码见试卷最后部分。

三 Code Snapshot (10 分)

假如 Client.java 某次运行之后控制台输出如下所示：

```
Art (JournalPaper)          selected by Windy
Science (ConferencePaper)   selected by Stormy
Reading (Thesis)            selected by Cloudy
Tech (JournalPaper)         selected by Stormy
```

请基于上述输出结果，绘制 Client.java 本次运行到第 18 行代码结束时的完整 snapshot diagram，表示出该时刻内存中的对象及其之间的关系。如果某些对象的属性值无法从上述输出中获知，可以标识为“未知”。假定在 main()运行启动之后 JVM 从未进行 GC，你的图中也需要包含所有即等待 GC 的对象。

草　纸

（草纸内不得答题）

密

封

线

授课教师_____

姓名_____

学号_____

院系_____

本题有两个子问题，请二选一作答，全答则按前第 1 个子问题评分

四 安全性和健壮性(10 分)

子问题 1:表示泄漏 PaperList 类的 selectPaperByRandom()方法(见 PaperList 类代码的第 15-22 行)，将某个 Student 对象作为参数传递进去，根据该学生希望选择的论文数量和偏好的论文类型，从 PaperList 中随机选择符合要求的论文并返回结果 List<Paper>。

(1) 第 14 行是否存在表示泄露的风险？第 22 行呢？请阐述理由。若存在风险，需说明如何修改（文字陈述即可，无需给出代码）

(2) 除了这两处有嫌疑之外，还有其他地方（不限于 PaperList 类）有表示泄露的嫌疑吗？请指出（类名+行号），分别简要阐述理由和建议的修改策略。

子问题 2：异常处理 Paperlist 类的第 17 和 18 行，用于判断目前可选的论文数目与学生希望选择的论文数量之间的关系。目前的代码逻辑是：如果可选论文数目少于学生希望选择的数量，则不再为学生选择论文，而是直接返回 null。但是，返回 null 并不是一种好的编程风格。为此，增加一个自定义 checked 异常 NoEnoughPapersException，当发生该异常时，程序应提示输出“学生 XX 希望选论文 X 篇，但目前只有 Y 篇可选”，不再为该学生选择论文。请写出该异常类的定义代码，对 Paperlist 类的相关代码进行修改，使之可抛出异常，并修改 Client.java 代码，使之捕获该异常。

草　纸

（草纸内不得答题）

本题有两个子问题，请二选一作答，全答则按前第 1 个子问题评分

五 面向可维护性的设计模式(10 分)

子问题 1 Paperlist 类的 selectPaperByRandom()函数目前采用随机选择论文的方案（见第 19 行调用的 randomSelectPapers()方法），它有两个参数：可选论文清单 availablePapers、需要选择的论文数量），返回一组选定的论文。假如用户希望将来在“随机选择”之外还有其他更灵活的论文选择策略（例如按论文加入列表的次序从早到晚来选、根据论文 title 的关键词来选择、等等），并可在 client 端灵活使用某种特定的论文选择策略。请基于 Stategy 设计模式对当前设计进行修改，以最小的修改代价让程序灵活适应各种论文选择策略。

子问题 2 考虑到将来对 PaperSelection 类的扩展，为了遵循 OCP 原则，使用 Visitor 设计模式对该类进行改造，程序员将来可开发多个新的 visitor 子类，实现对 PaperSelection 内部存储数据的各种灵活查询（例如：查询有多少学生已经选择了至少 1 篇论文）。请对 PaperSelection 类进行改造，并试着撰写一个具体的 visitor 类，以查询有多少学生已选择了至少 1 篇论文。

答题时可以绘制类图来描述你的设计，也可以用文字阐述。

密 封 线

姓名_____ 授课教师_____

学号_____

院系_____

草 纸

（草纸内不得答题）

授课教师____ 姓名____ 学号____ 院系____

密 封 线

六 (10分) Spec 和测试

(1) 下面给出了 PaperList 类的 selectPaperByRandom()方法的 spec，但不完整。请在空白处补充相应的内容，使用中文和代码中出现的类名、方法名、变量名等。不要自己想象，要基于问题描述中和代码中所给出的信息进行 spec 设计。在做此题时，你务必已经完成了第五题。另：若你认为该函数需抛出 NoEnoughPapers Exception 之外的其他异常以应对 pre-condition，请自行在下方空白处补充。

```
/**
 * For a student who has not yet selected any papers, this method randomly selects a set
 * of papers (which has not yet been unselected by any other students) from paper list,
 * and the result should strictly satisfy preferences of the student (preferred quantity
 * and paper type).
 *
 * @param student



 * @return



 * @throws NoEnoughPapersException



 *




*/
```

(2)为 selectPaperByRandom()设计黑盒测试用例。考虑到该方法的复杂性，先从最基本场景开始测试：针对一个固定的 PaperList 对象，仅考虑一个 Student 对象。假如你的 JUnit 测试代码如下所示，2-7 行构建了一个 PaperList 对象，第 8 行构建了一个 Student 对象，第 9-13 行进行具体测试。请使用等价类划分和边界值分析策略，设计若干 Student 对象，逐个替换到第 8 行的位置，使测试尽可能充分。将测试用例填入下面表格里即可，并简要阐述你的 testing strategy。注：设计测试用例时，需要基于你在(1)中为该函数写出的 spec。

草　　纸

（草纸内不得答题）

```
   @Test
1  void testSelectPaperByRandom() {
1     final PaperList listOfRainy = new PaperList("Rainy",
2        Arrays.asList(
3           new Thesis         ("Reading",     Arrays.asList("a", "b")),
4           new ConferencePaper("Science",     Arrays.asList("b")),
5           new JournalPaper   ("Technology",  Arrays.asList("c", "d")),
6           new JournalPaper   ("Art",         Arrays.asList("e"))
         )
      );
8     final Student student = new Student("Cloudy", 1, PaperType.Thesis);
9     try {
10      List<Paper> papers = listOfRainy.selectPaperByRandom(student);
11      assertTrue(...);   //此处开展具体测试，判断后置条件是否满足，可能是多行
      }
12    catch (NotEnoughPapersException e) {
13      assertTrue(...);    //此处开展具体测试，判断后置条件中的异常是否正确捕捉
      }
   }
```

测试策略(testing strategy)：

```
/**
  *
  *
  *
  *
  *
  *
  *
*/
```

测试用例（行数不足的话，请自行在下方空白处补充新行）：

Name	preferredQuantity	preferredType	设计依据
"Cloudy"			
"Cloudy"			
"Cloudy"			
"Cloudy"			
"Cloudy"			
"Cloudy"			

授课教师_____ 姓名_____ 学号_____ 院系_____

密 封 线

草　纸

（草纸内不得答题）

密

封

线

授课教师_____

姓名_____

学号_____

院系_____

七 语法和正则表达式(10 分)

目前学生只能选择三种类型论文之一（Thesis、ConferencePaper、JournalPaper）或者不限定类型（Any）。若对该需求做变更，允许学生使用复杂的条件来限定自己倾向的论文类型和数量。学生可构造一个字符串，用“与”(&&)、“或”(||)、“非”(!)、圆括号(“()”)来组合四种类型与数量。为简化起见，三种类型分别用 T、C、J 表示，不限定类型用 A 表示。^仅能出现在 T、C、J 之后，^之后为数字（不多于 2 位的正整数）。

下面给出了一些合法的字符串实例：

- J^1　　选择 1 篇期刊论文
- A^4　　选择 4 篇任意类型的论文
- (!J)^3　　选择 3 篇非期刊论文
- T^1 || T^2　　选择 1 篇 或 2 篇学位论文
- J^1 || C^2 || T^1　　选择 1 篇期刊论文 或 2 篇会议论文 或 1 篇学位论文
- C^2 && T^1　　选择 2 篇会议论文 + 1 篇学位论文
- J^2 || (!J)^1　　选择 2 篇期刊论文 或 1 篇其他论文
- T^1 && (J^2 || C^1)　　选择 1 篇学位论文 + 2 篇期刊或 1 篇会议论文
- (T^1 && J^2) || (T^2 && C^1)　　要么选择 1 篇学位论文和 2 篇期刊论文，要么选择 2 篇学位论文和 1 篇会议论文

请为该字符串写出 BNF 格式的语法表示，并给出其正则表达式。由于这里并未给出“非法”的字符串限定规则，你的语法和正则表达式只需要能够完全覆盖上述合法实例即可（意即:哪怕它能覆盖其他非法字符串，也认为是正确的）。

草　纸

（草纸内不得答题）

为答题方便，本页及后续各页，答题时可以从试卷上拆下来。交卷时单独提交，无需与前面各页再订到一起。

PaperType.java　论文类型，其中 Any 代表“无偏好，任何类型的论文均可”

```
public enum PaperType {
   Thesis,
   ConferencePaper,
   JournalPaper,
   Any
}
```

Student.java

```
  // Immutable class  学生
  public class Student {
1  private final String name;             //学生姓名
2  private final int preferredQuantity;   //希望选择的论文数量
3  private final PaperType preferredType; //偏好的论文类型

4  public Student(String name, int quantity, PaperType type) {
5     this.name = name;
6     this.preferredQuantity = quantity;
7     this.preferredType = type;
   }

8  public int getPreferredQuantity() {
9     return this.preferredQuantity;
   }

10 public PaperType getPreferredPaperType() {
11    return this.preferredType;
   }

12 public String getName() {
13    return name;
   }
  }
```

Paper.java

```
  // Immutable class  论文
  public class Paper {
1  private final String title;         //论文题目，长度至少为 3 个字符
2  private final List<String> authors; //作者列表，最少 1 个，最多不限

3  public Paper(String title, List<String> authors) {
4     this.title = title;
```

密 封 线

授课教师＿＿＿＿＿＿ 姓名＿＿＿＿＿＿ 学号＿＿＿＿＿＿ 院系＿＿＿＿＿＿

草　纸

（草纸内不得答题）

```
5      this.authors = authors;
   }

6  public String getTitle() {
7      return this.title;
   }

8  public List<String> getAuthors() {
9      return Collections.unmodifiableList(authors);
   }

10 public String toString() {
11     return this.title +" (" + this.getClass().getName() +")";
   }
  }

   //三种类型的论文
12 class Thesis           extends Paper {...}
13 class JournalPaper     extends Paper {...}
14 class ConferencePaper  extends Paper {...}
```

PaperList.java

```
  // Mutable class 教师发布的一组论文，以及学生选择的结果
  public class PaperList {
1  private final String teacher;      //教师姓名
2  private final List<Paper> papers;  //该教师发布的论文清单
3  private final PaperSelection selections = new PaperSelection();//学生选择论文结果

4  public PaperList(String teacher, List<Paper> papers) {
5      this.teacher = teacher;
6      this.papers = papers;
   }
   //根据论文类型，到 papers 中找出尚未被选择的论文，返回清单。type=Any 意味着任何未被他人
   //选择的论文都会包含在返回值中
7  private List<Paper> getUnselectedPapersByType(PaperType type) {
8      List<Paper> results = new ArrayList<>();
9      for(Paper p : this.papers) {
10         if(! this.selections.isSelected(p)) {
11             if(type == PaperType.Any
12                    || type.toString().equals(p.getClass().getName()))
                                              //注：此处代码试图得到 p 的具体类名
13                 results.add(p);
           }
       }
14     return Collections.unmodifiableList(results);
   }
```

草　纸

（草纸内不得答题）

```
    //根据学生偏好，为其随机选择一组论文
15  public List<Paper> selectPaperByRandom(Student student) {

16      List<Paper> availablePapers =
            getUnselectedPapersByType(student.getPreferredPaperType());

        //若可选论文的数量少于学生希望选择的数量，不再继续执行，直接返回 null
17      if(availablePapers.size() < student.getPreferredQuantity())
18         return null;

19      List<Paper> papers =  randomSelectPapers(availablePapers,
                                                 student.getPreferredQuantity());
20      for(Paper p : papers)
21         selections.addSelection(student, p);

22      return papers;
    }

23  public PaperSelection getSelectionResult() {
24      return this.selections;
25  }

     //从 availablePapers 论文清单中随机选择 q 篇论文，返回选择结果
26  private List<Paper> randomSelectPapers(List<Paper> availablePapers, int q) {
        //code ignored here
    }
  }
```

PaperSelection.java

```
  // Mutable class  维护已经被学生选择过的论文；若某论文未被选，则不包含在其中
  public class PaperSelection {
   //该 Map 中，key 为论文，value 为学生，表明某论文被某学生选择了
1   private final Map<Paper, Student> selections;

2    public PaperSelection() {
3       selections = new HashMap<Paper, Student>();
    }

    //学生 student 选择了论文 paper，将其加入到 selections 中
4   public boolean addSelection(Student student, Paper paper) {
5       selections.put(paper, student);
6       return true;
    }

    //检查论文 paper 是否已被学生选择过，若是，返回 true，否则 false
```

授课教师_____ 姓名_____ 学号_____ 院系_____

密 封 线

草　纸

（草纸内不得答题）

```
7  public boolean isSelected(Paper paper) {
       //code ignored here
8  }

9  public Map<Paper, Student> getSelections() {
10     return this.selections;
   }

11 public String toString() {
12     String str = "";
13     for(Paper p : selections.keySet()) {
14        str += p.toString() + "\t selected by " + selections.get(p) + "\n";
       }
15     return str;
   }
  }
```

Client.java

这里仅给出 main()中的代码，其他部分忽略

```
1     final PaperList listOfRainy = new PaperList("Rainy",
2        Arrays.asList(
3           new Thesis         ("Reading",     Arrays.asList("a", "b")),
4           new ConferencePaper("Science",     Arrays.asList("b")),
5           new JournalPaper   ("Technology",  Arrays.asList("c", "d")),
6           new JournalPaper   ("Art",         Arrays.asList("e"))
       )
      );

7     final List<Student> students = Arrays.asList(
8           new Student("Cloudy", 1, PaperType.Thesis),
9           new Student("Sunny",  1, PaperType.JournalPaper),
10          new Student("Windy",  1, PaperType.Any),
11          new Student("Stormy", 2, PaperType.Any));

12    for (Student student : students) {
13       new Thread(new Runnable() {
14          public void run() {
15             listOfRainy.selectPaperByRandom(student);
            }
16       }).start();
      }

17    Thread.sleep(1000);
18    System.out.println(listOfRainy.getSelectionResult().toString());
```

密 封 线

姓名 学号 院系 授课教师

草　纸

（草纸内不得答题）