Java程序设计编程作业-7

作业说明

请按照本文档的题目要求和代码框架完成编程作业。代码框架的示例结构如下所示，在`src`文件夹目录下，每一个题目对应一个文件夹(package)。在package内部，通常题目要求实现若干个类(class)，分别对应一个java源代码文件（未完整实现）。此外，每个题目有一个包含main函数的公有类`Test.java`，功能是对题目中待实现的类进行测试。最后，在`src`文件夹目录下有一个`testutil`包，是测试时所用到的工具，同学无需阅读或更改。

```
└─hw1
    ├─README.pdf
    │
    └─src
        ├─problem1
        │    ├─Class1.java (待实现)
        │    ├─Class2.java (待实现)
        │    └─Test.java
        ├─problem2
        │    ├─Class1.java (待实现)
        │    └─Test.java
        ...
        │
        └─testutil
```

同学们完成作业的建议步骤如下：

1. 按照文档，在代码框架中完成题目所要求的功能类的完整实现。
2. 运行每个题目的`Test`类，根据测试结果进行调试。
3. **严格**按照以下文件**结构**和**命名**提交源代码文件，并打包成 zip 格式上传，文件命名为`学号_姓名.zip`。

```
└─2020123456_张三 (学号_姓名)
    ├─report.pdf/txt/md (实验报告，可选)
    │
    └─src
        ├─problem1 (抽象名，根据实际题名决定，小写)
        │    ├─Class1.java （抽象名，根据实际类名决定，首字母大写）
        │    ├─Class2.java
        │    └─... (其他java源代码)
        │
        ├─problem2
        │    ├─Class1.java
        │    └─... (其他java源代码)
        ...
```

请注意：

- 编程作业为个人作业，请独立完成。如果提交的代码参考了资料或个人，请在report中标注来源，否则视为抄

袭。

- 编程作业采用自动化评判，因此不按照规定格式提交作业可能导致错误。
- 请严格按照代码框架和文档要求对成员变量、类、函数、文件命名，函数参数类型、名称、顺序和函数返回值类型请严格按照文档实现，此外类、成员变量和函数访问权限请按照文档要求指定。**Java严格区分大小写，不符合要求的命名会导致错误。**
- 请不要更改代码框架的文件结构和命名，请不要删除代码框架中java源文件开头的包声明`package ...;`。
- 除了题目要求外，可以根据需要实现其他类、函数和成员变量。可以提交其他新增的源代码文件，在新增的**源代码文件开头必须进行包声明**`package ...;`。
- **`Test`类对应的源代码文件以及`testutil`包可以不提交**，同时同学们也可以不改写。无论它们是否被提交或改写，评判过程中会对这两类文件重新覆盖。
- `Test`类包含了测试程序的一些基本测例。无特殊说明的情况下，**作业框架中提供的测例是评判时测例的子集**。最终的评判分数为同学所提交代码在全体测例中通过的比例。同学们可以在`Test`类中根据需要补充测例，进行测试。**本次作业提供的测例是评判时的100%**。
- 实验报告是可选的，内容为除了源代码外额外需要说明的内容，例如参考的资料，复杂题目的实现思路等。**实验报告不影响本次评判。**
- 文档和实验框架的错误、歧义和bug可联系助教wujy22@mails.tsinghua.edu.cn。
- Java API官方在线文档：https://docs.oracle.com/en/java/javase/19/docs/api/
- 本次作业要求实现的所有类和方法均为公有的。

Problem 1: Ticket（50’）

在包`ticket`中实现公有的功能类`Ticket`，用以模拟多线程抢票，Ticket类必须继承自Thread。

本题目提供的辅助类TicketPool模拟一个票池，对外提供两个接口:

```
int getRest() //查询当前票池中，剩余的票数
void reduceRest(int delta) //将当前票池中，剩余的票数减少delta，即票被抢走了delta
```

Ticket类用以模拟单个客户的抢票，在此过程中需要调用TicketPool.getRest来查询剩余的票数，并调用TicketPool.reduceRest进行抢票操作。请注意TicketPool对象只有一个，而Ticket对象有多个，抢票的多个线程是并发的，程序需要处理冲突。

你需要在功能类 Ticket 中实现以下成员:

```
/* 构造函数，两个参数，tpl代表目标票池（辅助类的实例），treq代表需要抢的票数 */
public Ticket(TicketPool tpl, int treq)
/*
  抢票过程:查询目标票池剩余的票数
  如果剩余票数大于等于需要的票数，则在目标票池中抢走需要的票数
  如果剩余票数小于需要的票数，则在目标票池中抢走剩余票数
*/
public void run()
/*
查询抢票结果:返回实际抢到的票的数量
*/
public int getObtained()
```

注意:

- 本题提供测试类Test和辅助类TicketPool，注意这两个类均不需要修改。

Problem 2: Game1（50’）

在包`game1`中创建功能类`Game1Client`，完成一个猜数游戏。游戏分为服务端(Game1Server)和客户端(Game1Client)，服务端会生成一个数字 a，每次客户端可以询问一个数字 b，服务端会返回a和b的大小关系，客户端需要在限定的次数内猜出数字a。

服务端(Game1Server)的实现在框架代码中提供，无需更改。客户端欲服务端的通信协议规定如下:

- 客户端向服务端发送一条消息“hello”。
- 服务端向客户端发送三行消息，每行消息是一个整数，代表这局游戏的信息，这三行消息从前到后分别是`lower_bound`，`upper_bound`，`limit`，代表lower_bound ≤ a ≤ upper_bound，客户端最多可以猜limit次。
- 客户端向服务端发送若干行消息，每行消息是一个整数b，代表客户端这一次猜的数字，服务端收到消息之后，会返回一行消息，代表本次猜测的结果，具体地：
  - 如果 b < a，服务端返回一行消息“<”；
  - 如果 b > a，服务端返回一行消息">"；
  - 如果 b = a，即客户端猜中了数字 a，那么服务端返回一行消息“Correct!”，然后再向客户端发送一行消息“bye”，游戏结束，服务端关闭。
- 客户端收到消息“bye”之后，游戏结束，客户端关闭。

一个可能的通信例子如下，下面只是示例，不是终端输出的内容:

```
# 被猜测的数字是4
client: hello
server: 1
        10
        10
client: 2
server: <
client: 7
server: >
client  4
server: Correct!
        bye
```

同学需要创建并实现客户端类Game1Client，接口如下：

```
Game1Client(String host, int port) //构造函数，host和port是服务器端的地址和端口。
int run() //客户端开始和服务端交互，交互结束后返回一个数字，代表客户端猜测的数字。
```

数据范围：

- 0 ≤ lower_bound ≤ upper_bound ≤ 10_000_000
- limit ≥ 35

注意：

- Game1Client需要依据通信协议保证正常的交互流程，不遵守通信协议导致的会话无法结束或非正常结束也视为没有通过测试点。终端输出内容需要确保有`Accept`，表示测例通过。
- client的逻辑中没有用户键盘输入的部分，client猜测的数字应当由程序自行生成并通过socket发送至server。
- Game1Client类需要编写在一个独立的文件中，因为测试时Game1Server.java会被覆盖。
- Game1Client不需要继承`Thread`。