

**2019年春季学期  
计算机学院《软件构造》课程**

**Lab 6实验报告**

|  |  |
| --- | --- |
| 姓名 | 任永鹏 |
| 学号 | 1173710107 |
| 班号 | 1737101 |
| 电子邮件 | [805402160@qq.com](mailto:805402160@qq.com) |
| 手机号码 | 18800420869 |

**目录**

[1 实验目标概述 1](#_Toc11791305)

[2 实验环境配置 1](#_Toc11791306)

[3 实验过程 1](#_Toc11791307)

[3.1 ADT设计方案 1](#_Toc11791308)

[3.2 Monkey线程的run()的执行流程图 3](#_Toc11791309)

[3.3 至少两种“梯子选择”策略的设计与实现方案 3](#_Toc11791310)

[3.3.1 策略1 3](#_Toc11791311)

[3.3.2 策略2 4](#_Toc11791312)

[3.4 “猴子生成器”MonkeyGenerator 4](#_Toc11791313)

[3.5 如何确保threadsafe？ 5](#_Toc11791314)

[3.6 系统吞吐率和公平性的度量方案 5](#_Toc11791315)

[3.7 输出方案设计 5](#_Toc11791316)

[3.8 猴子过河模拟器v1 6](#_Toc11791317)

[3.8.1 参数如何初始化 6](#_Toc11791318)

[3.8.2 使用Strategy模式为每只猴子选择决策策略 6](#_Toc11791319)

[3.9 猴子过河模拟器v2 7](#_Toc11791320)

[3.9.1 对比分析：固定其他参数，选择不同的决策策略 7](#_Toc11791321)

[3.9.2 对比分析：变化某个参数，固定其他参数 7](#_Toc11791322)

[3.9.3 分析：吞吐率是否与各参数/决策策略有相关性？ 8](#_Toc11791323)

[3.9.4 压力测试结果与分析 9](#_Toc11791324)

[3.10 猴子过河模拟器v3 9](#_Toc11791325)

[4 实验进度记录 11](#_Toc11791326)

[5 实验过程中遇到的困难与解决途径 11](#_Toc11791327)

[6 实验过程中收获的经验、教训、感想 12](#_Toc11791328)

[6.1 实验过程中收获的经验和教训 12](#_Toc11791329)

[6.2 针对以下方面的感受 12](#_Toc11791330)

# 实验目标概述

本次实验训练学生的并行编程的基本能力，特别是 Java 多线程编程的能力。 根据一个具体需求，开发两个版本的模拟器，仔细选择保证线程安全（threadsafe） 的构造策略并在代码中加以实现，通过实际数据模拟，测试程序是否是线程安全 的。另外，训练学生如何在 threadsafe 和性能之间寻求较优的折中，为此计算吞 吐率和公平性等性能指标，并做仿真实验。 λ Java 多线程编程 λ 面向线程安全的 ADT 设计策略选择、文档化 λ 模拟仿真实验与对比分析

# 实验环境配置

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

在这里给出你的GitHub Lab6仓库的URL地址（Lab6-学号）。

<https://github.com/ComputerScienceHIT/Lab6-1173710107>

# 实验过程

请仔细对照实验手册，针对三个问题中的每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但千万不要把你的源代码全部粘贴过来！）。

## ADT设计方案

设计了哪些ADT、各自的作用、属性、方法；

给出每个ADT的specification；

（可选）以类图形式给出多个类之间的关系。

有三个adt

Monkey,ladder,pedal

Monkey

有

**private** **int** number; 号码

**private** **int** direction;方向

**private** **int** speed;速度

**private** **int** sleep;sleep时间

**private** **int** ladderNumber;梯子的编号

**private** **int** rungNumber;踏板的编号

**private** **int** realspeed;真实的速度

**private** **int** age;猴子的年龄

除了set get以及构造器之外

还有findladder（）用于给猴子寻找梯子

Moveonladder（）用于让猴子在梯子上面移动

Ladder

List<pedal> list = **new** ArrayList<pedal>();梯子上的踏板的集合

**private** **int** ID = -1;梯子的编号

**private** **int** direction = -1;梯子上面的猴子的编号

**private** **int** monkeyAmount;梯子上面的猴子的数量

除了set get之外

只剩一个构造器了

Pedal

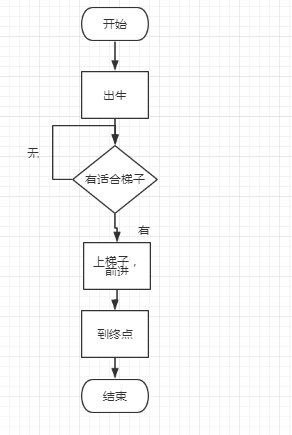
**private** **boolean** isempty = **true**;踏板上面是否有猴子

monkey inMonkey = **null**;踏板上有的话，猴子

只有set get

## Monkey线程的run()的执行流程图

这里无需考虑具体采用的梯子选择策略。



## 至少两种“梯子选择”策略的设计与实现方案

### 策略1

先选择没有猴子的梯子，若所有梯子上都有猴子，则优先选择没有与我对向而行的猴子的梯子；如果满足该条件的梯子有很多，则随机选择。

### 策略2

先选择整体推进速度最快的梯子，就是梯子上离我距离最近的猴子的真实行进速度最快。如果梯子上都没有猴子，则随便选择一个梯子。

## “猴子生成器”MonkeyGenerator

如何设计和实现。

**private** **static** Map<Integer, monkey> *monkeys* = **new** ConcurrentHashMap<Integer, monkey>();

**private** **static** Map<Integer, ladder> *ladders* = **new** ConcurrentHashMap<Integer, ladder>();

声明两个变量

然后

/\*\*

\* 产生所有的猴子

\*

\* **@param** k 每次产生所有猴子的数量

\* **@param** time 产生猴子的时间

\* **@param** sum 总共要产生的猴子的总数

\* **@param** maxSpeed 猴子的最大速度

\*

\* **@return** 返回猴子的集合

\*/

**public** **static** Map<Integer, monkey> generateMonkeys(**int** k, **int** time, **int** sum, **int** maxSpeed)

然后根据参数，先构建ladder,后构建monkey

## 如何确保threadsafe？

对所有的猴子上锁，即每次只能有一只猴子进入到线程中去做决策。

## 系统吞吐率和公平性的度量方案

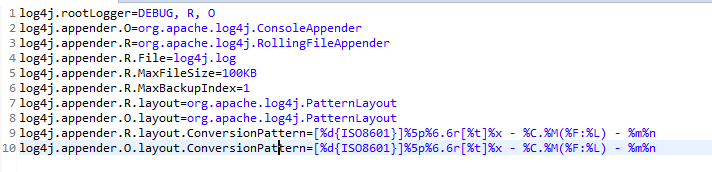
“吞吐率”是指：假如𝑁只猴子过河的总耗时为𝑇秒，那么每只猴子的 平均耗时为𝑋 = ் ே秒，则吞吐率𝑇ℎ = ே ்表征每秒钟可过河的猴子数目。

“公平性”是指：如果 Monkey 对象 A 比 B 出生得早，那么 A 应该不 晚于 B 抵达对岸，则为“公平”；若 A 比 B 晚到对岸，则为“不公 平”。设 A 和 B 的产生时间分别为𝑌௔和𝑌௕，抵达对岸的时间分别为𝑍௔ 和𝑍௕，那么公平性𝐹(𝐴, 𝐵) = ൜ 1, 𝑖𝑓 (𝑌௕ − 𝑌௔ ) ∗ (𝑍௕ − 𝑍௔ ) ≥ 0 −1, 𝑜𝑡ℎ𝑒𝑟𝑤𝑖𝑠𝑒 。对𝑁 只猴子两两计算其之间的公平性并综合到一起，得到本次模拟的整体 公平性𝐹 = ∑ ி(஺,஻) (ಲ,ಳ)∈౸ େಿ మ , Θ = {(𝐴, 𝐵)|𝐴 ≠ 𝐵, (𝐵, 𝐴) ∉ Θ}，其取值范围 为[−1,1]。

## 输出方案设计

日志

利用之前lab4的配置文件



要写日志的类



要写什么logger.就可以了

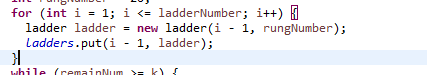
## 猴子过河模拟器v1

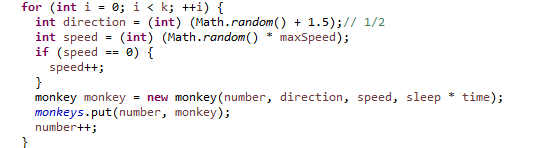
### 参数如何初始化

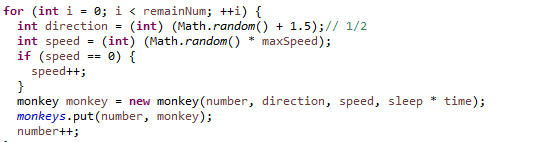
在



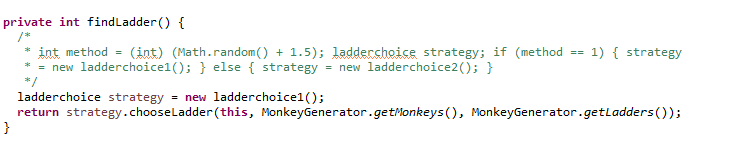
根据输入的参数可以进行初始化，先初始化ladder



然后初始化monkey  




### 使用Strategy模式为每只猴子选择决策策略

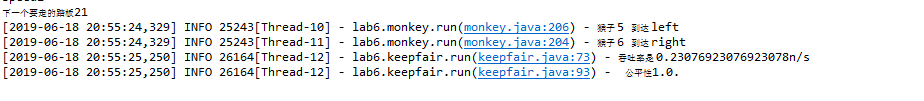


内容被注释掉了，不想该代码了，直接看吧/xyx

## 猴子过河模拟器v2

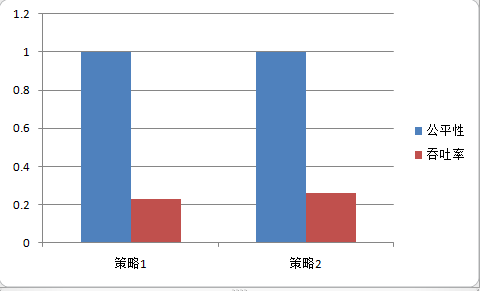
在不同参数设置和不同“梯子选择”模式下的“吞吐率”和“公平性”实验结果及其对比分析。

### 对比分析：固定其他参数，选择不同的决策策略





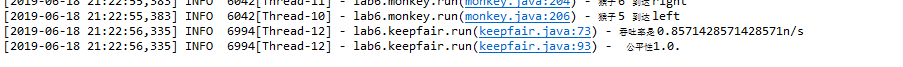


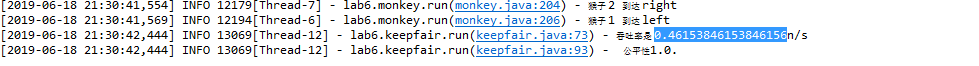


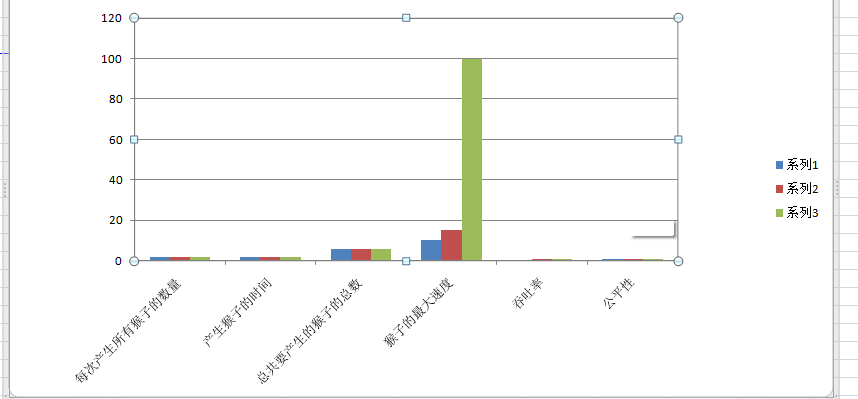
### 对比分析：变化某个参数，固定其他参数

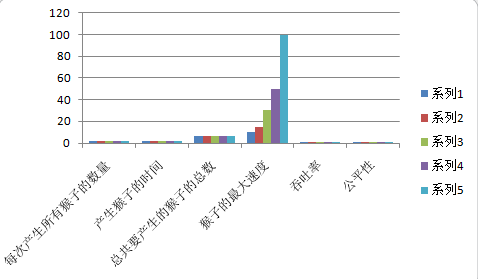












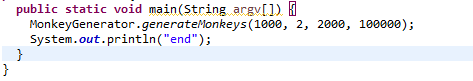
### 分析：吞吐率是否与各参数/决策策略有相关性？

从上面的数据可以看出吞吐率和测略，参数都有关系

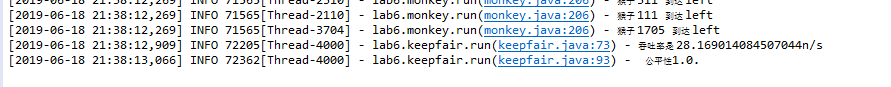
吞吐率和各个参数的关系很明显。主要体现在桥越多，单个猴子速度越快，过河越快。

二种策略中第二种策略是效率最好的，因为该策略保证了空的梯子数目最多，让反方向的猴子等待的时间最少，甚至可以直接上梯子。

### 压力测试结果与分析



吞吐率变大



## 猴子过河模拟器v3

针对教师提供的三个文本文件，分别进行多次模拟，记录模拟结果。

|  |  |  |
| --- | --- | --- |
|  | 吞吐率 | 公平性 |
| Competiton\_1.txt |  |  |
| 第1次模拟 | 0.7246376811594203 | 1.0 |
| 第2次模拟 | 0.7228915662650602 | 1.0 |
| 第3次模拟 | 0.7334963325183375 | 1.0 |
| 第4次模拟 | 0.7299270072992701 | 1.0 |
| 第5次模拟 | 0.7246376811594203 | 1.0 |
| 第6次模拟 | 0.7263922518159807 | 1.0 |
| 第7次模拟 | 0.7109004739336493 | 1.0 |
| 第8次模拟 | 0.7211538461538461 | 1.0 |
| 第9次模拟 | 0.7425742574257426 | 1.0 |
| 第10次模拟 | 0.7407407407407407 | 1.0 |
| 平均值 | 0.728 | 1.0 |
| Competiton\_2.txt |  | 1.0 |
| 第1次模拟 | 0.7874015748031497 | 1.0 |
| 第2次模拟 | 0.8025682182985554 | 1.0 |
| 第3次模拟 | 0.7911392405063879 | 1.0 |
| 第4次模拟 | 0.7886435331232144 | 1.0 |
| 第5次模拟 | 0.8025682182985554 | 1.0 |
| 第6次模拟 | 0.7874015748031497 | 1.0 |
| 第7次模拟 | 0.7987220447284437 | 1.0 |
| 第8次模拟 | 0.7987220447284437 | 1.0 |
| 第9次模拟 | 0.7874015748031497 | 1.0 |
| 第10次模拟 | 0.8025682182985554 | 1.0 |
| 平均值 | 0.795 | 1.0 |
| Competiton\_3.txt |  | 1.0 |
| 第1次模拟 | 0.5952380952380952 | 1.0 |
| 第2次模拟 | 0.5780346820809249 | 1.0 |
| 第3次模拟 | 0.5882352941176471 | 1.0 |
| 第4次模拟 | 0.5882352941176471 | 1.0 |
| 第5次模拟 | 0.6024096385542169 | 1.0 |
| 第6次模拟 | 0.5617977528089888 | 1.0 |
| 第7次模拟 | 0.5813953488372093 | 1.0 |
| 第8次模拟 | 0.5813953488372093 | 1.0 |
| 第9次模拟 | 0.5780346820809249 | 1.0 |
| 第10次模拟 | 0.5952380952380952 | 1.0 |
| 平均值 | 0.585 | 1.0 |

# 实验进度记录

请使用表格方式记录你的进度情况，以超过半小时的连续编程时间为一行。

每次结束编程时，请向该表格中增加一行。不要事后胡乱填写。

不要嫌烦，该表格可帮助你汇总你在每个任务上付出的时间和精力，发现自己不擅长的任务，后续有意识的弥补。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 6.8 | 一天 | 熟悉实验内容写好adt | 完成 |
| 6.9 | 一天 | 写V1 | 没写完 |
| 6.10-6.16 | 很多天 | 写实验 | 中间出了点事情，没写。。。 |
| 6.17-6.18 | 两天 | 写实验，写报告 | Over，个人时间安排中间出了点事情，实验写的很垃圾。。。 |

# 实验过程中遇到的困难与解决途径

|  |  |
| --- | --- |
| 遇到的难点 | 解决途径 |
| 写的时候那个logger之前还会整，结果这次怎么也整不对。。之前就是一直按照这个链接弄得，结果那个配置文件怎么也显示不上  <https://blog.csdn.net/qq_18783269/article/details/62044105> | 找了其他资料也不好用，自己慢慢瞎整出来的 |
| 一开始测试的时候，我的猴子就一直在第一个踏板上，怎么也不动 | 求助同学，研究代码发现是go里面出现的问题，后面解决 |
|  |  |

# 实验过程中收获的经验、教训、感想

## 实验过程中收获的经验和教训

## 针对以下方面的感受

1. 多线程程序比单线程程序复杂在哪里？你是否能体验到多线程程序在性能方面的改善？

能

1. 你采用了什么设计决策来保证threadsafe？如何做到在threadsafe和性能之间很好的折中？

将关键的可变对象进行上锁，每次只允许一个进程访问。

1. 你在完成本实验过程中是否遇到过线程不安全的情况？你是如何改进的？

没

1. 关于本实验的工作量、难度、deadline。

实验还行，比之前的lab3工作量小很多，辛亏小很多，要不我写不完了。。。。。之前一周忙各种事情，就有点不想写实验，导致今晚还在整，个人问题，值得深思！

1. 到此为止你对《软件构造》课程的意见和建议。

Good，就是ppt多点例子，我理解起来也快点

1. 还有一周就要期末考试了，你准备如何复习？

把平时写博客上的雨课堂看看，然后看看ppt做做往年题。。。。每门课都是考前几天速成，相当真实！！！