

**Lab report**

|  |  |
| --- | --- |
| **Course**: | Class Libraries and Data Structures |
| **Semester**: | 1st semester of the academic year **2019-2020** |
| **Major**: | Software Engineering |
| **Class**: | 2018 |
| **Student Name**: | SONG,Xingjian（宋行健） |
| **Student ID:** | 222019321062006 |
| **Teacher:** | ZHAO, Hengjun (赵恒军) |

**School of Computer and Information Science**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | | Queue and Simulation | | | |
| Date | | Dec 19，2019 | Type | | □Confirmatory  √ Design  □Comprehensive |
| 1. **Objective & Requirements**    1. Understand the concept of container adapter    2. Know the implementation the queue container adapter in the STL    3. Grasp the use of queue container in a real application    4. Know the concept of simulation and can use simulation to solve a real problem    5. Know about the queueing theory and the exponential distribution theory | | | | | |
| 1. **Experimental environment (**platform and software**)**   Windows 7 (or higher versions) + Visual Studio 2010 (or higher versions) | | | | | |
| 1. **Experimental content and design** (Main Content, Procedure, Codes and Results)   Task 1  Improve the car wash simulation problem in the following ways based on the codes and slides sent to you:   * 1. Do not restrict the capacity of the car wash station.   2. The time interval between arrivals should be generated from an exponential distribution randomly. An exponential distribution has a distribution function   The parameter is based on the mean arrival interval from the user’s input, that is,   * 1. The service time for each car should be generated from an exponential distribution   ,  where  with mean service time provided by the user. Note that mean service time should be less than mean arrival time which means should be larger than   * 1. To generate a sequence satisfying exponential distributions, you could adopt the formula:   or  with *p* a random value in (0, 1)   * 1. Output the average waiting time and maximal queue length, using a large amount of simulation data. See if your calculated average waiting time equals:  1. 首先将队列长度CAPACITY设置成一个极大值，使得不会出现OVERFLOW的情况，并创建关于车辆到达时间间隔指数分布随机值的参数为1/20；      1. 在循环开始之前创建一个模拟全局变量TIME来模拟车辆到达的时间轴，通过随机生成的时间间隔tempT进行计算得出每辆车的到达时间。        1. 在Car类中创建服务时间变量serviceTime，设置成私有，然后创建关于车辆服务时间时间间隔指数分布随机值的参数为1/15          1. 将之前固定的服务时间改成指数分布的随机生成数      1. 在主函数中设置随机数生成的初始种子为当前时间      1. 最后输出模拟的平均等待时长和理论估计的平均等待时长，进行比较      1. 设置模拟样本数量为10万 | | | | | |
| 1. **Result analysis and discussion**（Analysis of experimental results and summing up the harvest and the existing problems）   **实验总结：**   1. 在本实验中要注意变量精度的问题，因为参数是小于1的小数，而且运算模拟的过程中为了模拟的效果，将所有的变量类型都设置为了double。另外需要注意的是在函数调用并传参的地方也要设置为double，不然会有精度损耗。 2. 开始时，我是在rand()函数前设置随机变量rand()的种子，每次调用一次rand()函数都会设置一次种子，这样生成的随机数并不是随机的。上网查阅资料后，发现C++的随机数生成器是伪随机，rand()是并不是一个真正的随机数产生器，即可以预测随机序列的顺序。所以开始时快速调用rand()时，它的随机数种子是一直没有变的，所以产生的随机序列是一样的。为了解决这种情况我将设置随机数种子的语句放在了主函数中，只在程序的最开始设置一次种子。 3. 在计算理论平均等待时间的时候，因为变量MIU是Car类中的，所以在调用的时候要加上作用域符号。   **实验结果：**    **通过模拟得到，模拟平均等待时间为45.2591，理论平均等待时间为45，在误差范围内，模拟成功。** | | | | | |
| Comments & Evaluation | Content & Design (A-E) | | |  | |
| Procedure & Codes (A-E) | | |  | |
| Results (A-E) | | |  | |
| Analysis & Discussion (A-E) | | |  | |
| Score (A-E):  Feedback comments: | | | | |