**《数据结构》课程实践报告**

|  |  |
| --- | --- |
| 课程实践题目： | **栈和队列主题实践报告** |
| 姓 名： | **刘柏田** |
| 院系： | **计算机** |
| 专 业： | **软件工程** |
| 年 级： | **大二** |
| 学 号： | **199050418** |
| 指导教师： | **任雪萍** |

年 月 日

**(任雪萍编写)**

**目 录**

1 该课程设计的目的………………………………………………………………x

2 任务分析…………………………………………………………………………x

3分析设计…………………………………………………………………………x

4调试分析…………………………………………………………………………x

5测试结果…………………………………………………………………………x

6小结………………………………………………………………………………x

7用户手册…………………………………………………………………………x

8附录………………………………………………………………………………x

9参考文献…………………………………………………………………………x

**各部分要求如下**：

1. **该课程设计的目的**

熟练使用C或C++或java编写程序，强化模块设计理念；

解决的实际问题（与题目对应）。

提高综合运用所学的理论知识和方法，独立分析和解决问题的能力（与题目相结合）。

初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

了解并掌握数据结构与算法的设计方法，具备初步的独立分析和设计能力；

1. **任务分析**

无歧义的任务说明：

1. 某汽车轮渡口，过渡口每次能载10辆车过江。过江时车辆分为客车类和火车类，上船有如下规定：同类船先到先上，客车先于货车上渡船，且每上4辆客车，才允许上一辆货车；若等待的客车不足4辆，则以货车替代；若无货车等待则允许客车都上船。设计一个算法模拟渡口管理。
2. 8\*8棋盘的使用栈解决的骑士游历问题

系统实现的功能：（1）能够根据算法的要求对当前车辆的行进进行判断和放置，并在足够的输入之后对结果有处理的输出（2）根据给定的起始位置求解一种能遍历棋盘的走法

（1）

测试数据：（包括合法和非法输入，及相应的合理输出，文字形式描述）

无

输出：

（2）

输入：0 0

输出

游历完成：

1 16 27 22 3 18 47 56

26 23 2 17 46 57 4 19

15 28 25 62 21 48 55 58

24 35 30 45 60 63 20 5

29 14 61 34 49 44 59 54

36 31 38 41 64 53 6 9

13 40 33 50 11 8 43 52

32 37 12 39 42 51 10 7

1. **分析设计**

需要处理的数据的逻辑结构(线性结构、树形结构、图状结构和集合)：线性结构

合适的存储结构（顺序存储、链式储存、索引顺序存储和散列存储等）： 链式存储 顺序存储

设计好的数据类型： 队列 二维数组

与功能对应的模块划分定义（定义时，写出函数头）：

模块调用关系：

1. **调试分析**

碰到的调试问题：

骑士游历的回溯太过于多 运行时间太过漫长 对回溯的情况无法准确把握

相应的解决方案：

使用贪心策略，对算法进行优化和解决，但是对于多解的情况并没有很好的处理

算法的时间复杂性：O(m\*m)

空间复杂性：O(m\*m)

教师提出的修改意见：对车辆的总量有限制 对与超过的或者过多的车辆停止输入

对骑士游历的过程中回溯的过程优化 减少回溯所需要的时间

修改结果： 增加回溯过程时候的判断和限制，增加对车辆进入时对已有车辆的判断

1. **测试结果**
2. 输入数据:
3. 无
4. 输出数据：
5. 货车 : 1号
6. 货车 : 2号
7. 货车 : 3号
8. 货车 : 4号
9. 客车 : 1号
10. 货车 : 5号
11. 货车 : 6号
12. 客车 : 2号
13. 客车 : 3号
14. 客车 : 4号
15. 输出排序结果
16. 客车 : 1号
17. 客车 : 2号
18. 客车 : 3号
19. 客车 : 4号
20. 货车 : 1号
21. 货车 : 2号

部分非法输入和测试结果：

测试数据： 8 8

输出：游历失败

结论：（程序是否正确?稳健？）

1. **小结**

设计是否合理？(从提高程序质量的角度考虑，比如：有无冗余代码等)

没有考虑到有上限或者其中一些小限制的地方，这些内容是能够减少后续无用的处理量的

可以改进的设想（具体分析，包括改进目标和改进手段等）：

（1）对方法的编写的时候更应该考虑到通用的解法，对数据的限制不应该写的特别死

（2）对于涉及到回溯的算法，要考虑代价和能不能尽量的减少代价

1. **用户手册**

机器硬件及操作系统有无要求？

无 ，Java对跨平台支持度高

需要什么软件运行测试？

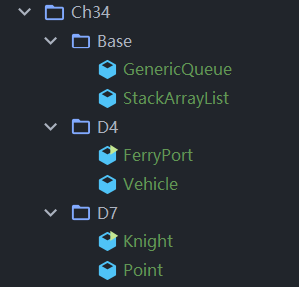
JetBrains Inteligence Idea

运行后，该怎么操作？（说明每一个步骤）

打开项目后运行src\Ch34\D4\FerryPort 或者 \D7\Knight 中的 main 函数

1. **附录**

程序清单如下：



1. **参考文献**

参考的任何资料（包括网上的资料）