

**课程设计(综合实验)报告**

( 2019 – 2020 年度第 2 学期)

名 称： 操作系统综合实验

题 目： 独占设备分配回收

院 系： 计算机系

班 级： 软件1802

学 号： 201809000121

学生姓名： 王毅涛

指导教师： 王德文

设计周数： 分散进行

成 绩：

日期： 年 月 日

**一、 综合实验的目的与要求**

本实验模拟模拟设备的分配与回收。

通过实验过程，加深实验者对设备管理方案的理解。

**二、实验正文**

实验是模拟，因此我们把设备管理抽象化，此处我设计了设备表，下辖若干设备类表，某一设备类表管理若干同一类型的设备。设备表负责总体管理和命令行交互界面。用户根据提示输入信息，然后再输入相关信息。如果选择“分配”，那么提示输入符合要求的名称，相对号，类型号。然后查询相应类表中是不是有足够多的，空闲且设备完好的设备，如果有就分配，然后标记为正在使用。如果选择了释放，先输入作业名和类号，定位到对应类号的类型表，遍历设备表比较输入的作业名，找到就将其设定为不在使用，设备占用名清空。选择退出即可退出主程序，选择打印可以打印基本信息。

代码和流程图见附录。

**三、综合实验总结或结论**

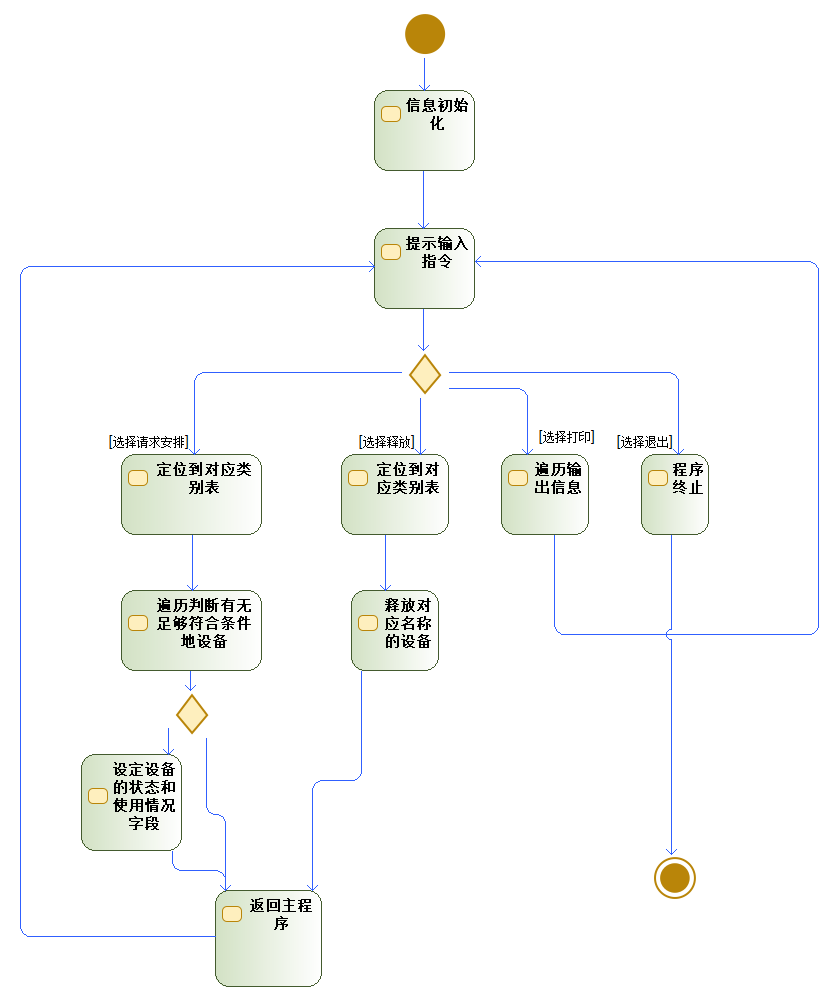
实验目的基本达成。本实验的难点在于合理有效地统筹组织各个功能类

**四、参考文献**

[1] 汤小丹, 梁红兵.《计算机操作系统》. 西安电子科技大学出版社, 第四版. 2014-05

**附录（设计流程图、程序、表格、数据等）**

流程图：



代码：

EquipmentChart:

1. **package** fourExclusiveDevice;
3. **import** java.util.Scanner;
5. **public** **class** EquipmentChart {
6. Scanner in =**new** Scanner(System.in);
7. **int** typeMax;
8. **int** arrayLen;
9. Typechart [] typechartArray;//存放类表的表，不同表项之间不同类别,号码就是类别
10. **public** **void** initialize() {
11. System.out.println("请输入有多少类设备,然后回车");
12. **int** typenum\_tmp=in.nextInt();
13. **this**.typeMax=typenum\_tmp;
14. **this**.typechartArray=**new** Typechart[typenum\_tmp];
15. **for** (**int** i=0;i<typenum\_tmp;i++) {//i是表示种类，j表示该类别下的局部编号
16. System.out.println("请输入"+i+"号类有多少个设备,然后回车");
17. **int** typetotal=in.nextInt();//i号类的设备总数目
18. **this**.typechartArray[i]=**new** Typechart(typetotal);//初始化
19. **this**.typechartArray[i].setChartType(i);
20. **for** (**int** j=0;j<typetotal;j++) {
21. System.out.println("为第"+i+"类的设备"+j+"输入一个设备名吧,请避免重名！");
22. **this**.typechartArray[i].chart[j]=**new** Equipment(j, i, in.next());
23. }
24. System.out.println("有几个坏机器？");
25. **int** badNum=in.nextInt();//几个坏设备
26. **for** (**int** k=0;k<badNum;k++) {
27. System.out.println("输入损毁的设备号，0开始，"+(typetotal-1)+"结束，不符合条件的不会录入");
28. **this**.typechartArray[i].markTheBad(in.nextInt());
29. }
30. System.out.println("OK");
31. }
32. **this**.arrayLen=**this**.typechartArray.length;
33. **return**;
34. }//initialize
35. /////////////////////////////////////MAIN//////////////////////////////
36. **public** **void** Main() {
37. **this**.initialize();
38. **while** (**true**) {
39. System.out.println("选1分配，选2回收，选3退出 4打印 输入后按下回车");
40. **int** instruct=in.nextInt();
41. **if** (instruct==1) {
42. **this**.allocate();
43. }
44. **else** **if** (instruct==2) {
45. **this**.release();
46. }
47. **else** **if** (instruct==3) {
48. System.out.println("正在退出程序");
49. **return**;
50. }
51. **else** **if** (instruct==4) {
52. **this**.printme();
53. }
54. **else** {
55. System.out.println("输入格式不正确，请重新输入");
56. **continue**;
57. }
58. }
59. }
60. ////////////////////////////////////MAIN///////////////////////////////
61. **public** **void** allocate() {
62. System.out.println("输入作业名");
63. String name=in.next();
64. System.out.println("输入类型（数字） 0开始"+(**this**.typeMax-1)+"结束");
65. **int** type=in.nextInt();
66. **if** (type<0||type>=**this**.typeMax) {
67. System.out.println("类型号输入错误，返回到主菜单");
68. **return**;
69. }
70. **else** {;}
71. System.out.println("输入相对号,0开始,"+(**this**.typechartArray[type].getChart().length-1)+"结束");
72. **int** addr=in.nextInt();
73. **if**(addr<0||addr>=**this**.typechartArray[type].getChart().length) {
74. System.out.println("相对号输入错误，返回到主菜单");
75. **return**;
76. }
77. **else** {
78. **this**.typechartArray[type].allocate(name,addr);
79. }
80. }//allocate
82. **public** **void** release() {
83. System.out.println("输入作业名");
84. String name=in.next();
85. System.out.println("输入类型（数字） 0开始"+(**this**.typeMax-1)+"结束");
86. **int** type=in.nextInt();
87. **if** (type<0||type>=**this**.typeMax) {
88. System.out.println("类型号输入错误，返回到主菜单");
89. **return**;
90. }
91. **else** {
92. **this**.typechartArray[type].release(name);
93. }
94. }//release
96. **public** **void** printme() {
97. **for** (**int** i=0;i<**this**.typeMax;i++) {
98. System.out.println("以下是类型"+i);
99. Typechart tc=**this**.typechartArray[i];
100. System.out.println("可用资源 "+tc.available+"资源总数"+tc.total);
101. **for** (Equipment eq:**this**.typechartArray[i].chart) {
102. System.out.println("占用进程名称:"+eq.getOccupant()+"是否良好"+eq.isGood()+"是否正在使用"+eq.isInUse());
103. }
104. }
105. }

108. }
109. //分配设备要求输入作业名、设备类名和相对号；回收设备要求输入作业名和设 备类名

TypeChart:

1. **package** fourExclusiveDevice;
3. **public** **class** Typechart {
4. Equipment [] chart;//存放设备
5. **int** total;//总共存放的设备数目
6. //int globalStartAddr;//在设备表里的起始位置
7. **int** chartType;//INT表示的种类
8. **int** available;//目前可用的（好的，且没有分配）
10. **public** **void** markTheBad(**int** index) {
11. **if** (index<0||index>=**this**.chart.length) {
12. System.out.println("超出界限，拒绝录入");
13. **return**;
14. }
15. **else** {
16. **this**.chart[index].setGood(**false**);
17. **this**.available--;
18. }
19. }


23. **public** Typechart(**int** total) {//一开始都是好的空闲的。
24. **super**();
25. **this**.total = total;
26. **this**.chart=**new** Equipment[total];
27. **this**.available=**this**.total;
28. }
30. **public** Typechart() {
31. **super**();
32. // TODO Auto-generated constructor stub
33. }
34. **public** Equipment[] getChart() {
35. **return** chart;
36. }
37. **public** **int** getTotal() {
38. **return** total;
39. }
40. **public** **int** getChartType() {
41. **return** chartType;
42. }
43. **public** **int** getAvailable() {
44. **return** available;
45. }
46. **public** **void** setChart(Equipment[] chart) {
47. **this**.chart = chart;
48. }
49. **public** **void** setTotal(**int** total) {
50. **this**.total = total;
51. }
52. **public** **void** setChartType(**int** chartType) {
53. **this**.chartType = chartType;
54. }
55. **public** **void** setAvailable(**int** available) {
56. **this**.available = available;
57. }


61. **public** **void** allocate(String name, **int** addr) {
62. // TODO Auto-generated method stub
63. Equipment tmpeq=**this**.chart[addr];
64. **if** (tmpeq.isGood()&&!tmpeq.isInUse()) {
65. tmpeq.setOccupant(name);
66. tmpeq.setInUse(**true**);
67. **this**.available--;
68. }
69. **else** {
70. System.out.println("Equipment not available at this moment \nPlease release it or wait until it's repaired.");}
71. }


75. **public** **void** release(String name) {
76. // TODO Auto-generated method stub
77. **for** (Equipment tmpeq:**this**.chart) {
78. **if** (tmpeq.compareOccupantName(name)) {
79. tmpeq.setInUse(**false**);
80. tmpeq.setOccupant("");
81. **this**.available++;
82. **return**;
83. }
84. **else** {**continue**;}
85. }
86. System.out.println("No result released Please check your input.");
87. **return**;
88. }

91. }

Equipment:

1. **package** fourExclusiveDevice;
3. **public** **class** Equipment {
4. **int** globalNum;//绝对编号
5. **int** typeNum;//相对编号
6. **int** type;//类型
7. **boolean** inUse;//是不是正在使用
8. **boolean** good;//状态是不是良好
9. String Occupant;//占用的作业的名称
10. **public** Equipment(**int** typeNum, **int** type, String occupant) {
11. **super**();
12. **this**.typeNum = typeNum;
13. **this**.type = type;
14. Occupant = occupant;
15. **this**.inUse=**false**;
16. **this**.good=**true**;
17. }
18. **public** **boolean** compareOccupantName(String comp) {
19. **if** (comp.equals(**this**.Occupant)) {**return** **true**;}
20. **else** {**return** **false**;}
21. }
22. **public** **int** getGlobalNum() {
23. **return** globalNum;
24. }
25. **public** **int** getTypeNum() {
26. **return** typeNum;
27. }
28. **public** **int** getType() {
29. **return** type;
30. }
31. **public** **boolean** isInUse() {
32. **return** inUse;
33. }
34. **public** **boolean** isGood() {
35. **return** good;
36. }
37. **public** String getOccupant() {
38. **return** Occupant;
39. }
40. **public** **void** setGlobalNum(**int** globalNum) {
41. **this**.globalNum = globalNum;
42. }
43. **public** **void** setTypeNum(**int** typeNum) {
44. **this**.typeNum = typeNum;
45. }
46. **public** **void** setType(**int** type) {
47. **this**.type = type;
48. }
49. **public** **void** setInUse(**boolean** inUse) {
50. **this**.inUse = inUse;
51. }
52. **public** **void** setGood(**boolean** good) {
53. **this**.good = good;
54. }
55. **public** **void** setOccupant(String occupant) {
56. Occupant = occupant;
57. }

60. }