**ＯＳ專案書面報告**

**資科三**

103703031 彭麒家

負責JAVA multithread程式設計 以及實作影片錄製

103703030 邱天

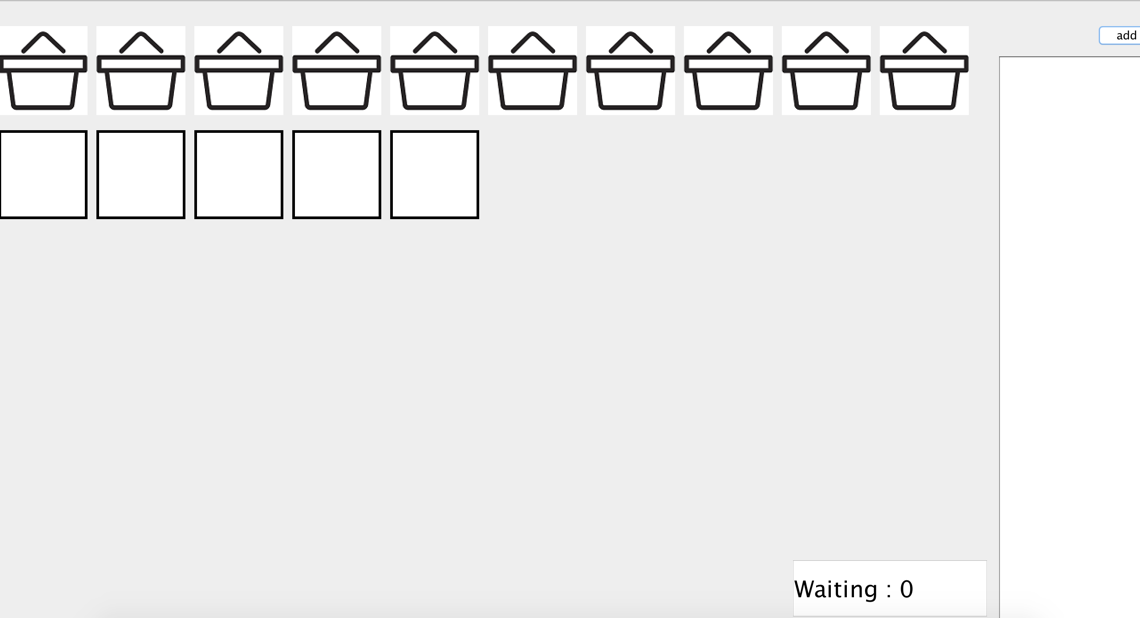
負責JAVA Deadlock程式設計 以及ＧＵＩ界面

**題目：swimming pool:**

程式設計語言選擇：JAVA IDE : Eclipse

使用ＪＡＶＡ多直行續 實作出Swimming Pool

ＧＵＩ介面：



**ＧＵＩ界面以及程式流程介紹：**

Swimming Pool Problem

上排為籃子 10個 若有人使用 籃子則會上色

下排為房間數 5個 若有人使用 該方格則會打一個Ｘ

人潮過多時 則會 輸出該人物找不到房間

直到房間空出後再進入找房間

右下角 waiting 人數：若人潮過多 籃子不夠用了！！

（防止Deadlock產生）

最右邊 console欄 顯示客人在該泳池俱樂部的流程

最右上角 add 為ＩＮＰＵＴ鈕

每按一次 多來一個客人

**程式詳細解說：**

程式當中使用while迴圈來進行synchronization的達成，我們在while迴圈當中設置五個ＣＡＳＥ，這五個ＣＡＳＥ讓依序進來這個由游泳池俱樂部的人，能進行不同的動作，CASE 1-5分別是 找房間中，找籃子中，游泳中，找房間回家，回家摟，中途為了避免ＤＥＡＤＬＯＣＫ的產生，若有人找不到房間亦或者是找不到籃子，會在ＣＯＮＳＯＬＥ欄顯是她目前找不到，等有空出來事才會再進去找。

而其中我們設置了ＷＡＩＴＩＮＧ的功能，我們假設程式當中有五個房間以及十個籃子，若此時籃子已經被十個人佔用了，我們會強制此要進去的人不准進來，因為會發生ＤＥＡＤＬＯＣＫ，等到籃子有被空出來，此人才能進去游泳池開始找房間。

我們程式當中繼承了ＪＡＶＡ的ＴＨＲＥＡＤ，因此基本上只要ＮＥＷ一個ＴＨＲＥＡＤ物件，程式即可以開始執行，並不像Ｃ＋＋需要宣告很多才能開始使用ＴＨＲＥＡＤ，再加上ＥＣＬＩＰＳＥ有內件的ＧＵＩ介面讓我們實作圖型介面動畫的執行，因此我最終選擇ＪＡＶＡ來近型此專案的實作，而非Ｃ＃或者Ｃ＋＋。

我們沒有使用到poisson ，取而代之的是，我們利用每個動作執行時間為三千毫秒，來進行實作，而利用手動加入人數的方式讓人進來，利用模擬方式，實施Daemon。

**實作影片連結網址：**

<https://youtu.be/6H9BprYXviQ>

**程式：**

**Demo.java:**

**package** osGUI;

**public** **class** Demo {

**public** **static** **int**[] *cubicle* = **new** **int**[5];

**public** **static** **int**[] *basket* = **new** **int**[10];

**public** **static** **int** *basket\_num*;

**public** **int** *number*;

Shared Variable

**public** Demo(){

start();

}

**public** **void** start() {

*number* = 0;

*basket\_num* = 0;

}

**public** **void** add(){

*number*++;

InThread inThread = **new** InThread(*number*);

inThread.start();

}

}

**Inthread.java**

**public** **class** InThread **extends** Thread {

**private** **int** number;

**private** **int** situ;

**private** **int** temp;

**public** InThread(**int** num) {

number = num;

situ = 0;

}

**static** **private** Object *obj* = **new** Object();

**public** **static** **void** staticWait() {

**synchronized** (*obj*) {

**try** {

*obj*.wait();

} **catch** (Exception e) {}

}

}

**public** **static** **void** staticNotify() {

**synchronized** (*obj*) {

*obj*.notify();

}

}

**public** **static** **void** leave(){

Demo.*basket\_num*--;

*staticNotify*();

}

**public** **void** run () {

**if**(Demo.*basket\_num* > 9){

GUItest.*textField*.setText("Waiting : " + ++GUItest.*waiting\_num*);

}

**while**(Demo.*basket\_num* > 9){

*staticWait*();

}

**if**(GUItest.*waiting\_num* > 0){

GUItest.*textField*.setText("Waiting : " + --GUItest.*waiting\_num*);

Critical Section

}

Demo.*basket\_num*++;

**while**(situ<5) {

**switch**(situ) {

**case** 0:{

System.***out***.println(number + "找房間中");

GUItest.*ta*.append(number + "找房間中" + "\r\n");

**for**(**int** i=0; i<5; i++) {

**if**(Demo.*cubicle*[i] == 0) {

Demo.*cubicle*[i] = number;

temp = i;

situ++;

GUItest.*cubicle\_change*(i, **true**);

**break**;

}

**if**(i==4){

System.***out***.println(number + "找不到房間");

GUItest.*ta*.append(number + "找不到房間" + "\r\n");

}

}

**break**;

}

**case** 1:{

System.***out***.println(number + "找籃子中");

GUItest.*ta*.append(number + "找籃子中" + "\r\n");

**for**(**int** i=0; i<10; i++) {

**if**(Demo.*basket*[i] == 0) {

Demo.*basket*[i] = number;

situ++;

GUItest.*basket\_change*(i, **true**);

**break**;

}

**if**(i==9){

System.***out***.println(number + "找不到籃子");

GUItest.*ta*.append(number + "找不到籃子" + "\r\n");

}

}

**break**;

}

**case** 2:{

**for**(**int** i=0; i<5; i++) {

**if**(Demo.*cubicle*[i] == number) {

Demo.*cubicle*[i] = 0;

GUItest.*cubicle\_change*(i, **false**);

}

}

System.***out***.println(number + "游泳中");

GUItest.*ta*.append(number + "游泳中" + "\r\n");

situ++;

**break**;

}

**case** 3:{

System.***out***.println(number + "找房間回家");

GUItest.*ta*.append(number + "找房間回家" + "\r\n");

**for**(**int** i=0; i<5; i++) {

**if**(Demo.*cubicle*[i] == 0) {

Demo.*cubicle*[i] = number;

temp = i;

situ++;

GUItest.*cubicle\_change*(i, **true**);

**break**;

}

**if**(i==4){

System.***out***.println(number + "找不到房間回家");

GUItest.*ta*.append(number + "找不到房間回家" + "\r\n");

}

}

**break**;

}

**case** 4:{

System.***out***.println(number + "回家囉");

GUItest.*ta*.append(number + "回家囉" + "\r\n");

**for**(**int** i=0; i<10; i++) {

**if**(Demo.*basket*[i] == number) {

Demo.*basket*[i] = 0;

GUItest.*basket\_change*(i, **false**);

Demo.*cubicle*[temp] = 0;

GUItest.*cubicle\_change*(temp, **false**);

*leave*();

}

}

situ++;

**break**;

}

}

**try** {

Thread.*currentThread*();

Thread.*sleep*(3000);

}

**catch**(InterruptedException e) {

e.printStackTrace();

}

}

}

}

有

Notify()

離開更衣間

清空籃子

換衣服

找更衣間

離開泳池

游泳

離開更衣間

找更衣間

裝進籃子

換衣服

沒有

Wait()

先詢問是否有籃子

進入泳池