系統程式\_\_作業二

Linking Loader

老師：黃溪春

班級：資訊二乙

學號：D0372639

姓名：吳冠融

目錄

1. 實作程式語言
2. 演算法
3. 測試資料
4. 程式碼
5. 輸出資料
6. 問題與討論
7. 心得

壹、實作語言：Java

貳、演算法：

類別(class)與副程式：

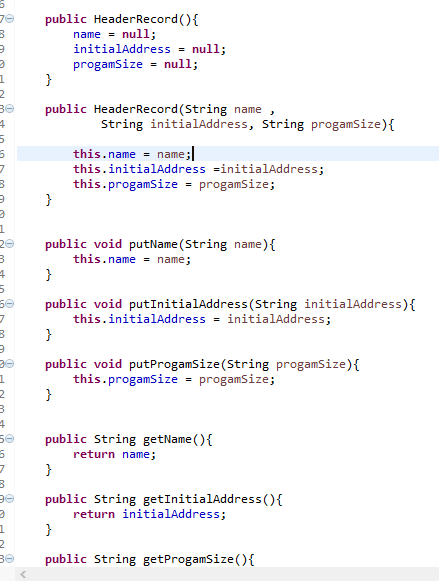
|  |  |
| --- | --- |
| 類別名稱 | 類別功能 |
| HeaderRecord | 儲存程式起始位址、程式大小 |
| TextRecord | 儲存object code |
| ModificationRecord | 儲存欲重定位位址與重定位點 |
| linkingLoader | 實作Linking loader 功能 |
| Compass(String[] subStrs , String CSADDR) | 讀取檔案，將資料分別存入HeaderRecord、TextRecord、ModificationRecord及SYMTABLE |
| Compass2() | 進行重定位功能 |
| PrintSpace(int space,int tIn,int i) | 印出空白符號及資料，使輸出可讀性變佳 |
| SYMTABLE | 建立外部參考位址 |

參、測試資料

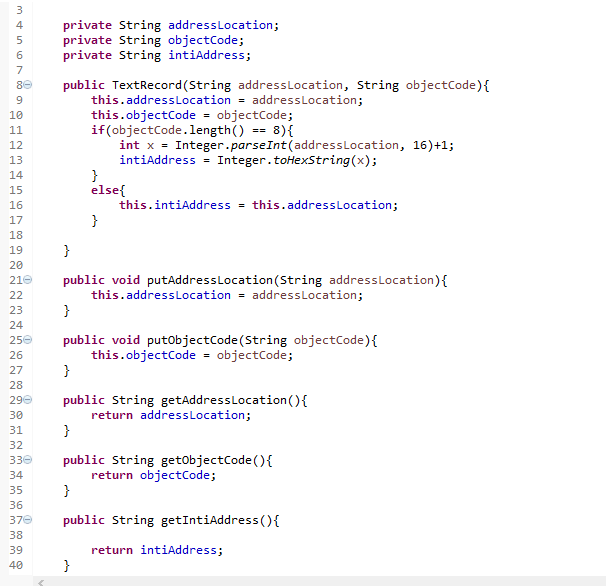
|  |
| --- |
| H^COPY ^000000^001033  D^BUFFER^000033^BUFEND^001033^LENGTH^00002D  R^RDREC ^WRREC  T^000000^1D^172027^4B100000^032023^290000^332007^4B10000^3F2FEC^032016^0F2016  T^00001D^0D^010003^0F200A^4B100000^3E2000  T^000030^03^434F46  M^000004^05^+RDREC  M^000011^05^+WRREC  M^000024^05^+WRREC  E^000000  H^RDREC ^000000^00002B  R^BUFFER^LENGHT^BUFEND  T^000000^1D^B410^B400^B440^77201F^E3201B^332FFA^DB2015^A004^322009^5790000^B850  T^00001D^0E^3B2FE9^13100000^4F00000^F1^000000  M^000018^05^+BUFFER  M^000021^05^+LENGTH  M^000028^06^+BUFFER  M^000028^06^+BUFEND  E  H^WRREC ^000000^00001C  R^LENGTH^BUFFER  T^000000^1C^B410^77100000^E32012^332FFA^53900000^DF2008^B850^3B2FEE^4F0000^05  M^000003^05^+LENGTH  M^00000D^05^+BUFFER  E |

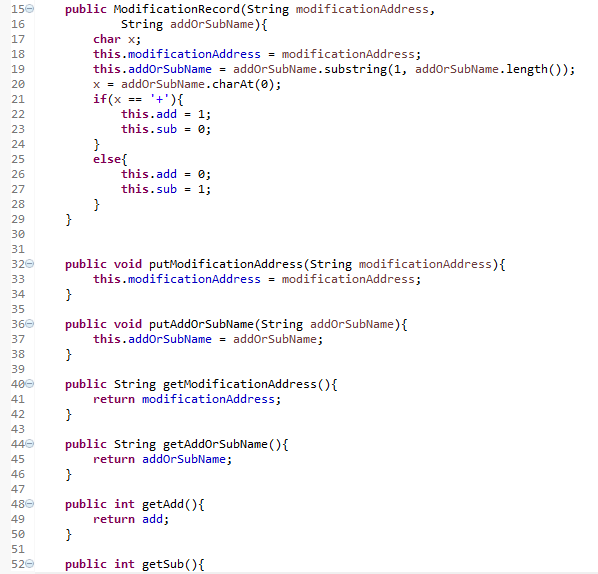
肆、程式碼

一、HeaderRecord：實作建構子、Getter與setter

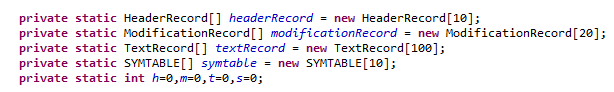


二、TextRecord：實作建構子、Getter與setter



1. ModificationRecord：實作建構子、Getter與setter

四、linkingLoader(主程式和副程式)：

4-1設定變數名稱

4-2設定變數，並開啟欲讀取檔案，建立SYMTABLE。



4-3列印前一行保留下來的值。

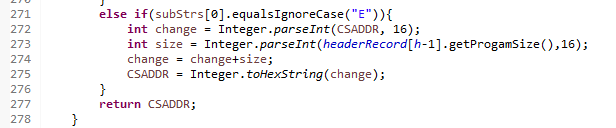
4-4列印值，並計算列數是否已經達到16Bytes

4-5 假如超過16Byte，進行保留工作，並在下一行先行輸出。

4-6資料保留區(‘--’)或已經完全列印完資料後的區間(‘xx’)表示。

四-2副程式1：Compass(String[] subStrs , String CSADDR):判斷Object Code Format 讀入資料為哪一項Format，Header、Text、Define等。





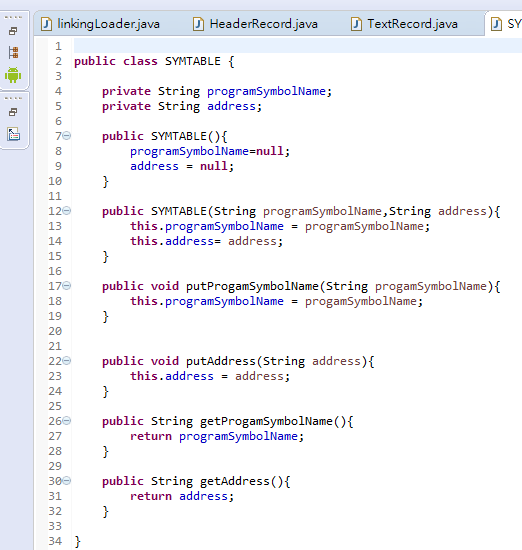
四-3副程式2：Compass2()尋找重定位位址，並且進行重定位功能。





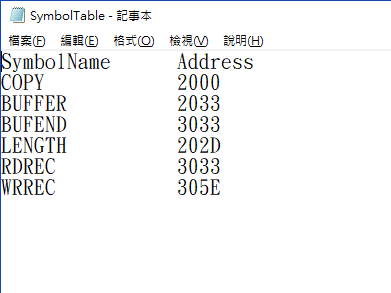
四-4副程式3：PrintSpace(int space,int tIn,int i) 讓程式每印4Bytes後印出一個空白符號，使表格更容易讀懂。

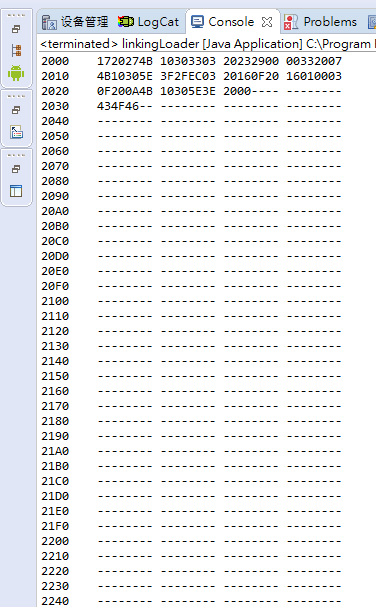


五SYMTABLE：實作建構子、Getter與setter

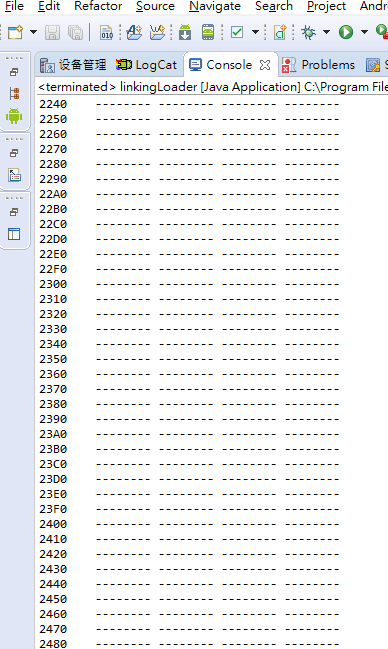
伍、輸出資料

SYMTABLE：

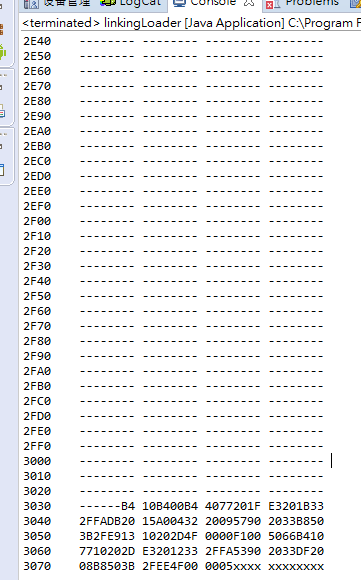


Linking

1-1

1-2

中間資料保留區如圖1-2。

1-3

陸、問題與討論

自己要創造物件使用，所以一開始要對自己的測資所要輸入資料了解，一開始的變數名稱用long，但是後來發覺java的16進位輸出全部是字串，所以之後把所有變數改成字串使用，並且利用內建的Integer進行字串轉整數，整數轉十六進制。

柒、心得

當初想說上一個程式用C++實作，雖然了解C++不少，但是比起java物件導向程式語言，覺得C++更難操作，因此想說用看看之前學過，而且很好用的JAVA，順便熟悉java的操作方法，雖然在轉進制那邊有出一點問題，不過最後突然想起java的操作一切都是字串，即使是整數變數，輸出也是用字串處理，再利用內建的轉換物件，相對於C++來說，更容易上手，前提是必須要熟悉物件規則。