

GROUP PROJECT

INVENTORY MANAGEMENT
PROGRAM

START

แนวความคิด



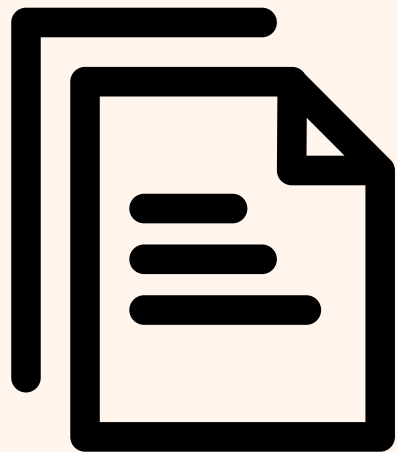
โปรแกรมนี้ออกแบบมาเพื่อจัดการคลังสินค้า
เพิ่มรายการสินค้าและกำจัดสินค้าตามเวลาที่เหมาะสม
(หมดอายุ) โดยอ้างอิงจาก **Theme : dispose**
โดยที่โปรแกรมใช้หลักการทำงานของ Linked List
ในการจัดเก็บข้อมูลของสินค้าเพื่ออ่าน
และแก้ไขผ่าน Console



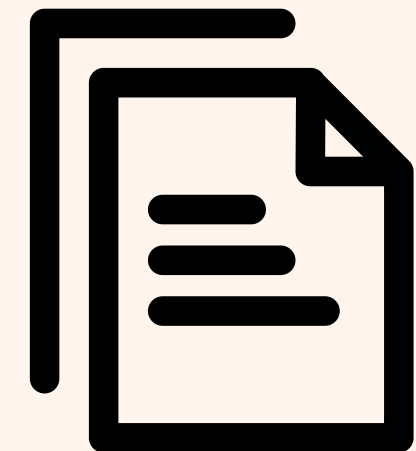
dispose

หลักการทำงานของโปรแกรมโดยสังเขป

Load ข้อมูลจาก
Txt file



Save ข้อมูลลง
Txt file



การเริ่มโปรแกรม

ตัวเลือกที่ 1 : Load data

ตัวเลือกที่ 2 : Show All Data

ตัวเลือกที่ 3 : Add, Delete, Edit your stock

ตัวเลือกที่ 4 : Sorting Your Stock

ตัวเลือกที่ 9 : Exit

ตัวเลือกที่ 99 : Then type "dev" to switch developer mode

Sorting Implementation

Insertion Sort Algorithms

เป็นคอนเซ็ปต์ในการเรียงข้อมูล
ข้อมูลของแต่ละคลาสจะถูกเรียงจาก
น้อยไปมาก

- เรียงโดยใช้ Node ID
- เรียงโดยใช้วันที่ผลิต (mfg)
- เรียงโดยใช้วันหมดอายุ (exp)
- เรียงโดยใช้น้ำหนักรวม (weight_All)
- เรียงโดยใช้ปริมาตรรวม (volume_All)



การเพิ่ม ลบ และแก้ไขข้อมูล

- Add Stock
- Delete Stock
- Edit Stock
- Back to Main Menu

การเพิ่ม ลบ และแก้ไขข้อมูล

```
Please select the category you wish to add
1 : Food
2 : Drinks
3 : Daily Use Product
4 : Specific Use Product
5 : Back
Input number : 
```

```
[!] Food's data is empty.
Please Enter NodeID you wish to delete
Enter 0 for Back to main menu
Enter NodeID : 
```



Function ในการจัดการ Linked List

```
void show_data();  
void insert(food*&);  
food* move_next();
```



Function การรับค่า

```
void ask(int &choice, string text = "Number Input");  
void ask(string &choice, string text = "String Input");  
void ask(double &choice, string text = "Double Input");  
void ask(char &choice, string text = "Char Input");
```

```
ask(s_choice, "Enter Text");  
Enter Text : ....(cin >> s_choice)....
```



Function การเพิ่ม ลบ และแก้ไข

```
void add_stock();
```

```
void delete_stock(int);
```

```
void show_delete_stock();
```

```
void edit_stock();
```

```
int return_NodeID();
```

```
void edit_NodeID(int);
```

```
void set_next(food *);
```



Function การเรียงลำดับข้อมูล

```
void sorting_function()
```

```
int compareDates(const string& date1 , const string& date2)
```

```
void LLFood::sorting_by_NodeID()
```

```
void LLFood::sorting_by_mfd()
```

```
void LLFood::sorting_by_exp()
```

```
double return_weightAll();
```

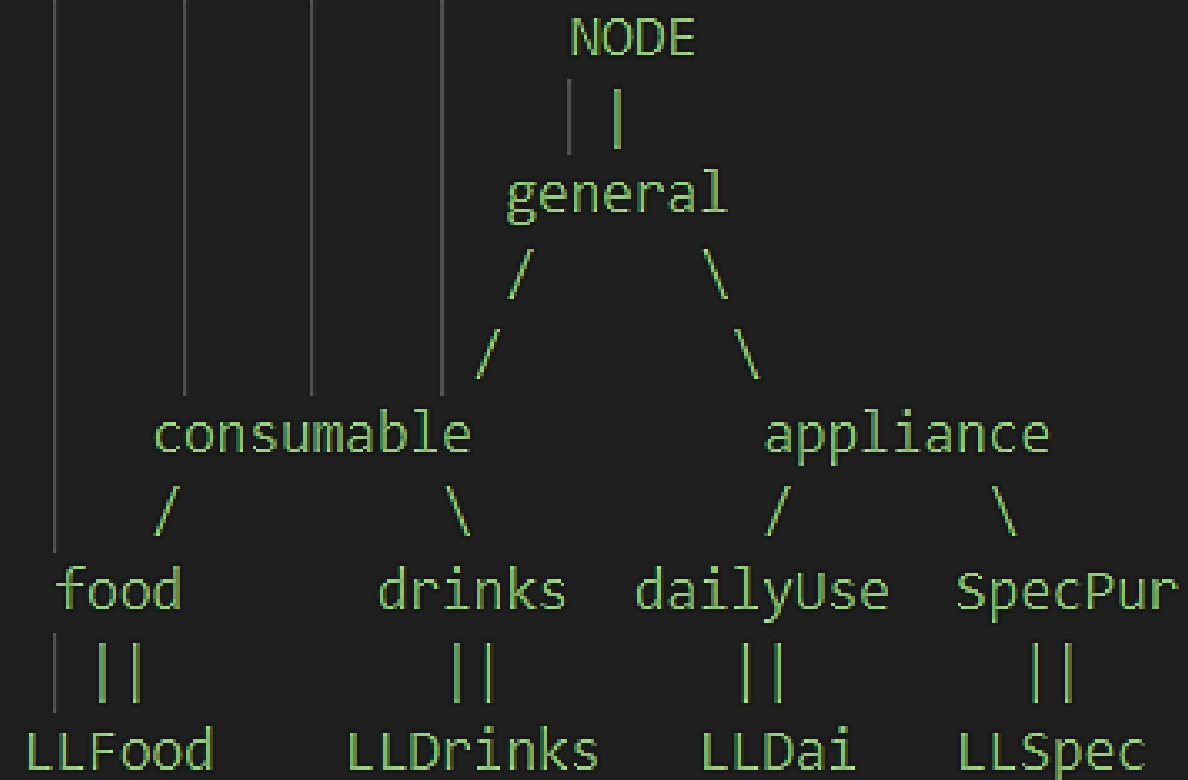
```
string return_mfg();
```

```
string return_exp();
```



Class Template

Structure of Inheritance



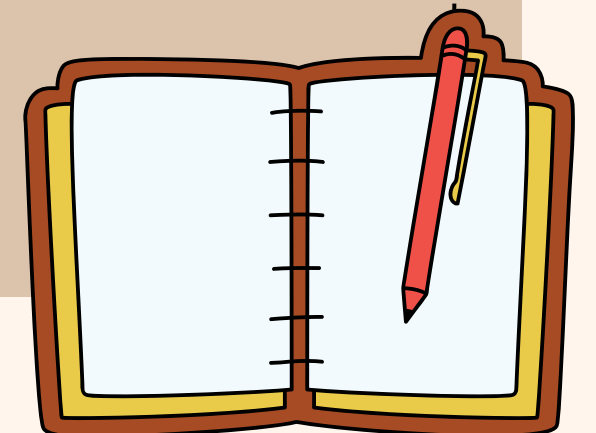
```
class food:public consumable{
private: food *next;
protected: double weightEach;
public:
food(string = "No Name", string = "No" ,....);
~food();
void show_data();
void insert(food*&);
food* move_next();
....
};
```

Requirement

จาก requirement กลุ่มของพวกเราได้ทำตามตรงทุกอย่าง
โดยอธิบายไว้ดังนี้

1) Use Class with Constructor :

- class general:public NODE
- class consumable:public general
- class food:public general
- class drink:public consumable
- class appliance:public general



Requirement

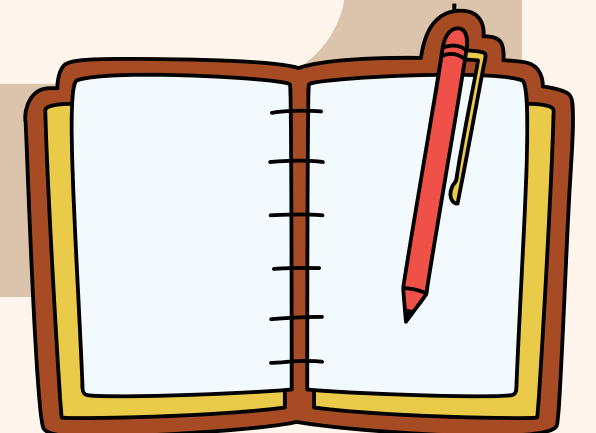
จาก requirement กลุ่มของพวกเราได้ทำตามตรงทุกอย่าง
โดยอธิบายไว้ดังนี้

2) Use Linked List or Stack or Queue or BST : (Linked List)

- LLFood
- LLDrinks
- LLDai
- LLSpec

```
class LLFood{  
    food *hol;  
    int size;  
public:  
    void add_node(food *&);  
    void show_all();  
    ....  
}
```

```
class LLDrinks{  
    drinks *hol;  
    int size;  
public:  
    void add_node(drinks *&);  
    void show_all();  
    ....  
}
```



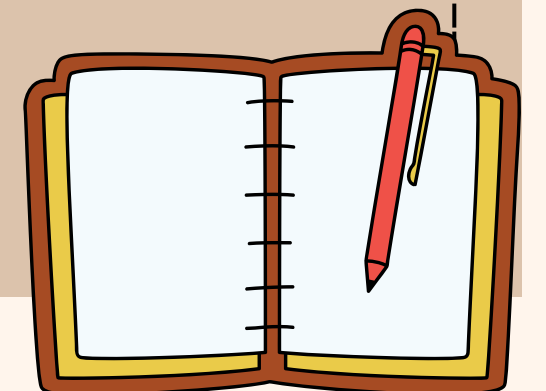
Requirement

3) Use polymorphism :

virtual ~general()=0;

virtual void show_data()=0;

```
public:  
    general(string = "No Name", str  
    virtual ~general() = 0;  
    virtual void show_data() = 0;
```

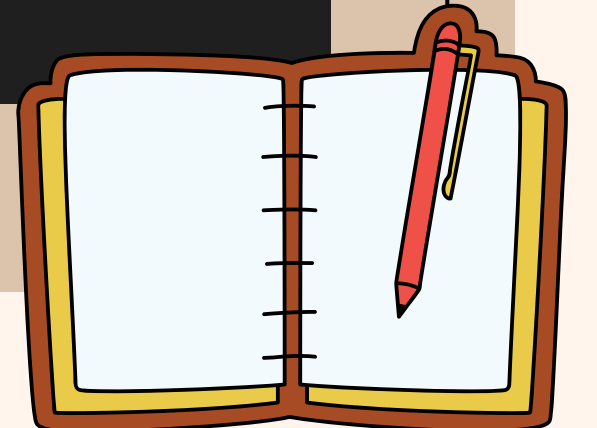


Requirement

4) Exception Handling :

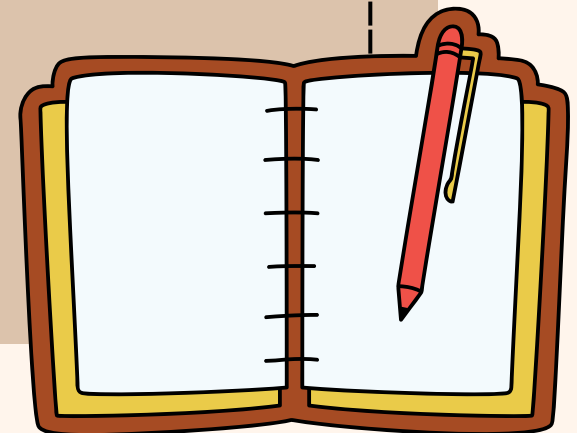
```
    }catch( exception &e ){
        cin.clear();
        cin.ignore(100,'\n');
        clear();
        if(dev) cout << "[Alert!] " << e.what() << endl;
        else cout << "[!] Input format error please try again." << endl;
        flag = 1;
    }
    catch(...){
        cin.clear();
        cin.ignore(100,'\n');
        clear();
        if(dev) cout << "[Alert!] " << "General Exception" << endl;
        else cout << "[!] Input format error please try again." << endl;
        flag = 1;
    }
}while(s_choice != 4 && s_flag == 0 || flag == 1);
```

```
void scin(double &choice, int &choice_check){
    try{
        cin >> choice;
        if(cin.fail()) {
            throw 0;
        }
    }catch( exception &e ){
        cin.clear();
        cin.ignore(100,'\n');
        if(dev) cout << "[Alert!] " << e.what() << endl;
        else cout << "[!] Input format error please try again." << endl;
        choice_check = 1;
    }
    catch(...){
        cin.clear();
        cin.ignore(100,'\n');
        if(dev) cout << "[Alert!] " << "General Exception" << endl;
        else cout << "[!] Input format error please try again." << endl;
        choice_check = 1;
    }
}
```



ข้อจำกัดของโปรแกรม

- ไม่สามารถแก้ไขข้อมูลอย่างละเอียดของแต่ละสินค้าได้ ผู้ใช้จำเป็นต้องกรอกข้อมูลของสินค้าชนิดนั้นใหม่ทั้งหมด
- รูปแบบของวันผลิต และวันหมดอายุไม่ได้มีการกำหนดที่แน่นอน หากมีการรับค่าข้อมูลในรูปแบบที่แตกต่างกัน อาจส่งผลต่อการเรียงลำดับตามวันผลิต หรือวันหมดอายุได้
- ไม่ได้มีการจำกัดจำนวนตัวอักษรตอนรับค่า หากมีการรับค่าข้อมูลที่มีจำนวนตัวอักษรมากเกินไป อาจส่งผลต่อรูปแบบการแสดงผลได้
- ไม่สามารถกรอกช่องว่าง ขณะรับค่าข้อมูลในรูปแบบ String ได้



Q

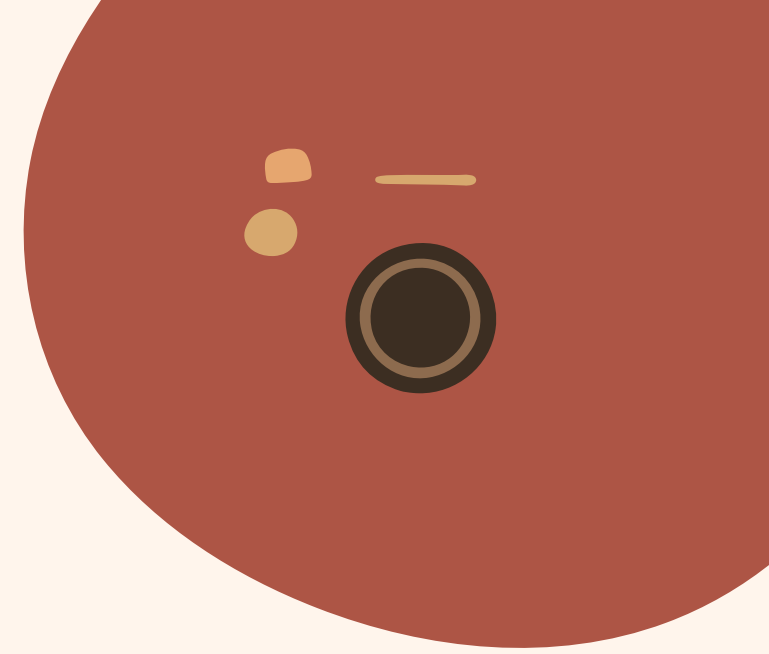
A

Question
Time

?



จัดทำโดย



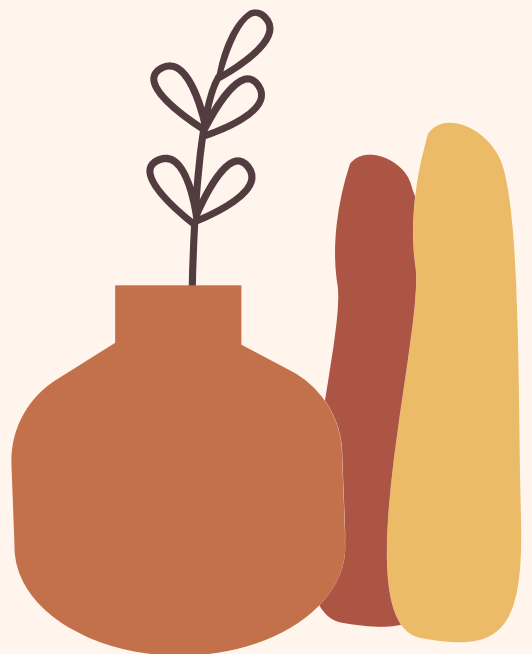
6513120 นายวิชญ์พล วิทยโกมล

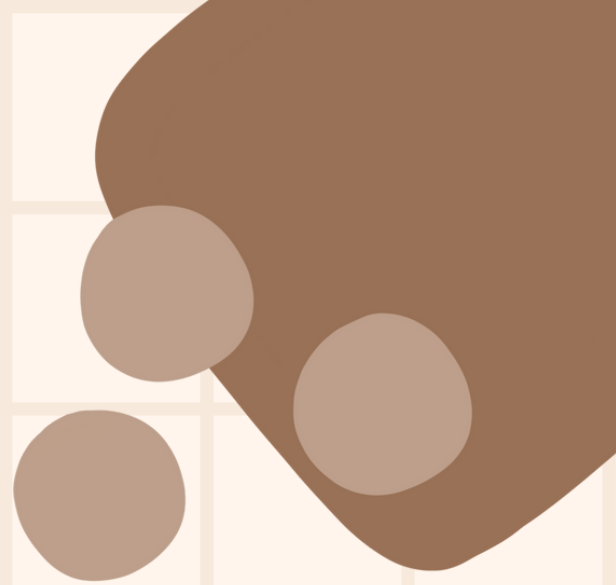
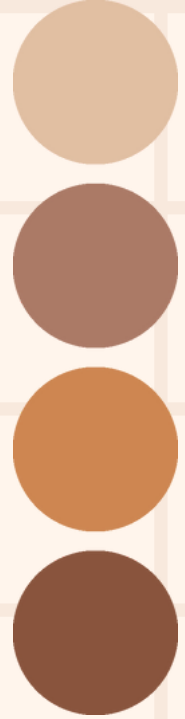
6513121 นายศุภกร ถาวรวงศ์

6513123 นายอนพัทธ์ กิจสมมารณ

6513136 นายมรรณพ ถาบัว

6513162 นายจิณณพัต น้อยวัฒน์





THANK YOU
SO MUCH!

