Introduction to programing HW7 Part3 Coding

108072147 林汶螢

1.

1.cpp > DonationList

```cpp
#include <iostream>
using namespace std;
class DonationList{
    private:
        int     numDonations;
        double  *donations;
        double  **arrPtr; //array of pointer
        void    selectSort();
    public:
        DonationList(int num, double gifts[]);
        ~DonationList();
        void show();
        void showSorted();
};
DonationList::DonationList(int num, double gifts[]){
    numDonations = num;
    if (num > 0){
        // Allocate an array of doubles
        donations = new double[num];
        // Allocate an array of pointers-to-doubles.
        this->arrPtr = new double*[num];
        // Initialize the arrays
        for (int count = 0; count < numDonations; count++)
        {
            *(donations+count) = *(gifts+count);
            this->arrPtr[count] = &donations[count];
        }
        // Now sort the array of pointers
        selectSort();
    }
}
void DonationList::selectSort(){
    int maxIndex;
    double *maxElem;
    for (int scan = 0; scan < (numDonations - 1); scan++)
    {
        maxIndex = scan;
        maxElem = *(arrPtr+scan);
        for(int i = scan + 1; i < numDonations; i++)
        {
            if (*(arrPtr+i) > maxElem)
            {
                maxElem = *(arrPtr + i);
                maxIndex = i;
            }
        }
        *(arrPtr+maxIndex) = *(arrPtr+scan);
        *(arrPtr+scan) = maxElem;
    }
}
```

1.cpp > DonationList

```cpp
void DonationList::show(){
    for (int count = 0; count < numDonations; count++)
        cout << donations[count] << " ";
    cout << endl;
}
void DonationList::showSorted(){
    for (int count = 0; count < numDonations; count++)
        cout << *(arrPtr[count]) << " ";
    cout << endl;
}
DonationList::~DonationList(){
    if (numDonations > 0){
        delete [ ] donations;
        donations = 0;
        delete [ ] arrPtr;
        arrPtr = 0;
    }
}
int main(int argc, char * argv[]){
    double given[100];
    int i=0,len=0;
    cout <<"Enter donations: (0 or not number means stop) \n";
    cin >> given[i];
    while(given[i] != 0){
            i++; cin >> given[i];
        }
    len = i;
    DonationList donationlist(len,given);
    cout<<"ur input before sorting: \n";
    donationlist.show();
    cout<<"ur input after sorting: \n";
    donationlist.showSorted();
    return 0;
}
```

執行結果:

```
Enter donations: (0 or not number means stop)
30
15
98
34
23
17
87
64
end
ur input before sorting:
30 15 98 34 23 17 87 64
ur input after sorting:
64 87 17 23 34 98 15 30
```

2.

2.cpp

```cpp
#include <iostream>
#include <iomanip>
#include <ctype.h>
#include <sstream>
#include <string>
using namespace std;
bool isNum(string str){
    stringstream sin(str);
    int d;
    char c;
    if(!(sin >> d)){
        return false;
    }
    if (sin >> c){
        return false;
    }
    return true;
}
void selectSort(int num,double *scores){
    int minIndex;
    double minElem;
    for (int scan = 0; scan < (num - 1); scan++){
        minIndex = scan;
        minElem = *(scores+scan);
        for(int i = scan + 1; i < num; i++)
        {
            if (*(scores + i) < minElem)
            {
                minElem = *(scores + i);
                minIndex = i;
            }
        }
        *(scores+minIndex) = *(scores+scan);
        *(scores+scan) = minElem;
    }
}
double average(double* scores, int num){
    double sum;
    for (int i=0; i<num; i++) sum += *(scores+i) ;
    return sum/num;
}
void show(int num,double *scoreArray){
    for (int count = 0; count < num; count++)
        cout << scoreArray+count << " ";
    cout << endl;
}
int main(int argc, char *argv[]){
    int num=-1;
    double input=0;
    string sinput;
    stringstream ss;
    bool asknum = true;
    while(num < 0){
        ss.clear();
        cout<<"Please input the number of scores u want to record."<<endl;
        getline(cin,sinput);
        stringstream ss(sinput);
        if(ss >> num ) {
            if(num > 0){
                break;
            }
            else{
            cout<<"You should input nonnegative number."<<endl;
            num = -1;
            }
        }
        else{
            cout<<"You should input nonnegative number."<<endl;
            num = -1;
        }
    }
    ss.clear();
    double *scoreArray = new double[num];
    for (int i = 0; i < num; i++){
        do {
            ss.clear();
            cout<<"Enter the score: ";
            getline(cin,sinput);
            ss << sinput;
            ss >> input;
            if(input < 0 || !isNum(sinput)){
                i--;
                cout<<"U should enter nonnegative number."<<endl;
                ss.clear();
                break;
            }
            else{
                *(scoreArray+i) = input;
                break;
            }
        }while(input >=0 && isNum(sinput));
    }
    selectSort(num, scoreArray);
    double avg = average(scoreArray , num);
    cout<<"\nThe sorted scores: \n";
    for (int i = 0; i < num; i++){
        cout<<scoreArray[i]<<" ";
        if((i+1)%10 == 0){
            cout<<endl;
            i-1;
        }
    }
    cout<<"\nAverages:"<<setw(20)<<right<<avg<<endl;
    delete  [] scoreArray;
    return 0;
}
```

執行結果:

```
Please input the number of scores u want to record.
a
You should input nonnegative number.
Please input the number of scores u want to record.
7
Enter the score: -5
U should enter nonnegative number.
Enter the score: 1
Enter the score: 3
Enter the score: 5
Enter the score: 98
Enter the score: 100
Enter the score: a
U should enter nonnegative number.
Enter the score: a
U should enter nonnegative number.
Enter the score: 0
Enter the score: 0

The sorted scores:
0 0 1 3 5 98 100
Averages:             29.5714
```

3.

```cpp
#include <iostream>
#include <cstring>
#include <sstream>
#include <ctype.h>
using namespace std;
void selectSort(int num,int *scores){
    int minIndex;
    double minElem;
    for (int scan = 0; scan < (num - 1); scan++){
        minIndex = scan;
        minElem = *(scores+scan);
        for(int i = scan + 1; i < num; i++)
        {
            if (*(scores + i) < minElem)
            {
                minElem = *(scores + i);
                minIndex = i;
            }
        }
        *(scores + minIndex) = *(scores + scan);
        *(scores+scan) = minElem;
    }
}
int median(int *a, int n){
    if(n % 2 !=0){
        return *(a+n / 2);
    }
    else
        return (*(a+n / 2) + *(a+n / 2 - 1)) / 2;
}
bool isNum(string str)
{
    stringstream sin(str);
    int d;
    char c;
    if(!(sin >> d)){
        return false;
    }
    if (sin >> c){
        return false;
    }
    return true;
}
int mode(int *a, int n){
    int cnt = 0, maxcnt = 0,index = 0;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            if (*(a+j) == *(a+i))
                cnt++;
        }
        if(cnt > maxcnt){
            maxcnt=cnt;
            index = i;
        }
        cnt = 0;
    }
    if(maxcnt == 1) return -1;
    return *(a+index);
}
int main(int argc,char* argv[]){
    int * pies;
    string input;
    stringstream ss;
    int myint;
    pies = new int[30];
    for (int i = 0; i < 30; i++){
        do {
            cout<<"Enter the number of pies u ate in a year : ";
            getline(cin, input);
            if(!isNum(input) ){
                i--;
                cout<<"U should enter nonnegative number."<<endl;
                myint = 0;
                break;
            }
            ss<< input;
            ss>> myint;
            ss.clear();
            if(myint < 0 ){
                i--;
                cout<<"U should enter nonnegative number."<<endl;
                myint = 0;
                break;
            }
            else{
                *(pies+i) = myint;
                break;
            }
        }while(myint >= 0 && isNum(input));
    }
    selectSort(30, pies);
    cout<<"\nThe pies people consumed after sorted: \n";
    for (int i = 0; i < 30; i++){
        cout<<*(pies + i)<<" ";
        if(*(pies + i)<10) cout<<" ";
        if((i+1)%10 == 0){
            cout<<endl;
            i-1;
        }
    }
    int Mode = mode(pies, 30);
    cout<<"The median of the pies: "<<median(pies, 30)<<endl;
    if (Mode == -1) cout<<"There is no mode.\n";
    else cout<<"The mode of the pies: "<<mode(pies, 30)<<endl;
    delete[] pies;
    return 0;
}
```

執行結果

```
Enter the number of pies u ate in a year : 0
Enter the number of pies u ate in a year : 0
Enter the number of pies u ate in a year : 0
Enter the number of pies u ate in a year : 7
Enter the number of pies u ate in a year : 8
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 15
Enter the number of pies u ate in a year : 67
Enter the number of pies u ate in a year : 34
Enter the number of pies u ate in a year : 30
Enter the number of pies u ate in a year : 96
Enter the number of pies u ate in a year : 98
Enter the number of pies u ate in a year : 24
Enter the number of pies u ate in a year : 11
Enter the number of pies u ate in a year : 7
Enter the number of pies u ate in a year : 7
Enter the number of pies u ate in a year : 7
Enter the number of pies u ate in a year : 2
Enter the number of pies u ate in a year : 3
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9
Enter the number of pies u ate in a year : 9

The pies people consumed after sorted:
0  0  0  2  3  7  7  7  7  8
9  9  9  9  9  9  9  9  9  9
9  9  11 15 24 30 34 67 96 98
The median of the pies: 9
The mode of the pies: 9
```

4.

```cpp
#include <iostream>
#include <sstream>
using namespace std;
class Stats{
    private:
        string city;
        string monthYear;
        double *rainfall;
        int cnt;
        int most, least;
        const string month[12]={"January", "February", "March",
                                "April", "May", "June", "July",
                                "August","September","October",
                                "November","December"
                                };
    public:
        Stats(string city, string monthYear){
            this -> city = city;
            this -> monthYear = monthYear;
            this -> rainfall = new double[30];
            for(int i=0; i<30; i++){
                *(rainfall+i) = 0;
            }
            this -> cnt = 0;
            this -> most = 0;
            this -> least = 0;
        }
        ~Stats(){
            delete [] rainfall;
        }
        double total(){
            double sum = 0;
            for(int i=0; i< this->cnt; i++){
                sum += this->rainfall[i];
            }
            return sum;
        }
        double average(){
            return this->total()/this->cnt;
        }
        double lowest(){
            double L=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] < L){
                    least = i;
                    L = this->rainfall[i];
                }
            }
            return L;
        }
        double highest(){
            double M=rainfall[0];
            for(int i=0; i< this->cnt; i++){
                if(this->rainfall[i] > M){
                    most = i;
                    M = this->rainfall[i];
                }
            }
            return M;
        }
        bool storeValue(double rainfall){
            if (rainfall<0 || cnt>30) return false;
            this->rainfall[this->cnt] = rainfall;
            cnt++;
            return true;
        }
    }
    void selectionSort(int array[], int size, int cnt[]){
        int start, minIndex, minValue, minCNT;
        for (start = 0; start < (size - 1); start++){
            minIndex = start;
            minValue = array[start];
            minCNT = cnt[start];
            for (int index = start + 1; index < size; index++){
                if (array[index] < minValue){
                    minValue = array[index];
                    minIndex = index;
                    minCNT   = cnt[index];
                }
            }
            array[minIndex] = array[start];
            array[start] = minValue;
            cnt[minIndex] = cnt[start];
            cnt[start] = minCNT;
        }
    }
    string displayDescending(){
       string ret;
        ret += " Rainfall Report Display by descending in " + this->city +" County : \n";
        ret += "============================================================\n";
        int copyRainfall[this->cnt], cnt[this->cnt];
        for(int i=0; i<this->cnt; i++){
            copyRainfall[i] = this->rainfall[i];
            cnt[i] = i;
        }
        selectionSort(copyRainfall,this->cnt, cnt);
        for(int i=this->cnt-1; i >= 0; i--){
            stringstream temp;
            string strtemp;
            temp<< copyRainfall[i];
            temp>> strtemp;
            temp.clear();
            ret += convert(cnt[i])+"  Rainfall: "+ strtemp +"\n";
        }
        return ret;
    }
    string convert(){
        stringstream ss;
        int monthnum;

        ss << this->monthYear.substr(4,2);
        ss >> monthnum;
        return this->month[monthnum] + ", " + this->monthYear.substr(0,4);
    }
    string convert(int cnt){
        stringstream ss;
        int monthnum, yearnum;
        string yearstr;
        ss << this->monthYear.substr(4,2);
        ss >> monthnum;
        ss.clear();
        ss << this->monthYear.substr(0,4);
        ss >> yearnum;
        ss.clear();
        yearnum += (monthnum + cnt)/12;
        monthnum = (monthnum + cnt)%12;
        ss << yearnum;
        ss >> yearstr;
        return this->month[monthnum] + ", " + yearstr;
    }
    string displayReport(){
```

```
            string ret;
            ret += this->convert() + " -" + this->convert(cnt) + " Rain Report for " + this->city +" County\n";
            stringstream temp;
            string strtemp;
            temp<< total();
            temp>> strtemp;
            temp.clear();
            ret += "Total rainfall in this period: " + strtemp + " inches\n";
            temp<< average();
            temp>> strtemp;
            temp.clear();
            ret += "Average monthly rainfall: " + strtemp + " inches\n";
            temp<< lowest();
            temp>> strtemp;
            temp.clear();
            ret += "The least rain fell in " + this->convert(least) + " with " + strtemp + " inches\n";
            temp<< highest();
            temp>> strtemp;
            temp.clear();
            ret += "The most rain fell in " + this->convert(most) + " with " + strtemp + " inches\n";
            return ret;
        }
};
int main (int argc, char* argv[]){
    string city,monthYear;
    double rainfall;
    int cont;
    string s;
    char c;

    cout<<"Please enter the city u want to analyze:";
    getline(cin,city);
    cout<<"Please enter the month u started to analyze:";
    cin>> monthYear;
    Stats stats(city, monthYear);

    cont = 1;
    while(cont){
        do{
            cout<<"\nHow's the rainfall in "<<city<<" ?\n";
            cin>>rainfall;
        }while(!stats.storeValue(rainfall));
        getline(cin,s);
        do{
            cout<<"Do u want to continue?\n";
            cout<<"Please enter Y/N only.\n";
            getline(cin,s);
            if(s[0]=='Y') cont =1;
            else cont = 0;
        }while(s.size()>1 || (s[0] != 'Y' && s[0] != 'N'));
    }
    cout<<endl<<stats.displayReport()<<endl;
    cout<<endl<<stats.displayDescending()<<endl;
    return 0;
}
```

Input -1

```
Please enter the city u want to analyze:YY
Please enter the month u started to analyze:202001

How's the rainfall in YY ?
0
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
317
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
5
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
2
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
0
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
1314
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
100
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
666
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
8
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
69
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
609
Do u want to continue?
Please enter Y/N only.
Y
```

Output

```
February, 2020 -July, 2021 Rain Report for YY County
Total rainfall in this period: 6553 inches
Average monthly rainfall: 385.471 inches
The least rain fell in February, 2020 with 0 inches
The most rain fell in July, 2020 with 1314 inches


 Rainfall Report Display by descending in YY County :
=====================================================================
July, 2020  Rainfall: 1314
June, 2021  Rainfall: 888
September, 2020  Rainfall: 666
April, 2021  Rainfall: 666
January, 2021  Rainfall: 612
December, 2020  Rainfall: 609
February, 2021  Rainfall: 606
March, 2021  Rainfall: 603
March, 2020  Rainfall: 317
August, 2020  Rainfall: 100
May, 2021  Rainfall: 88
November, 2020  Rainfall: 69
October, 2020  Rainfall: 8
April, 2020  Rainfall: 5
May, 2020  Rainfall: 2
June, 2020  Rainfall: 0
February, 2020  Rainfall: 0
```

Input -2

```
How's the rainfall in YY ?
612
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
606
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
603
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
666
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
88
Do u want to continue?
Please enter Y/N only.
Y

How's the rainfall in YY ?
888
Do u want to continue?
Please enter Y/N only.
N
```

5.

```cpp
#include <iostream>
#include <sstream>
#include <iomanip>
using namespace std;
string binsDescription[9] = {"Turkey", "Chocolate", "Egg", "Yoyo",
                             "ChickenBreast", "Gummy Bear", "Almond",
                             "Cookie", "Spaghetti" };
class InvBin{
    private:
        string description;     // Item name
        int qty;                // Quantity of items
                                // in this bin
    public:
        InvBin (string d = "empty", int q= 0){
            // 2-parameter constructor
            // with default values
            description= d;
            qty= q;
        }
        // It will also have the following public member functions. They
        // will be used by the BinManager class, not the client program.
        void setDescription(string d){
            this->description = d;
        }
        string getDescription(){
            return this->description;
        }
        void setQty(int q){
            this->qty = q;
        }
        int getQty(){
            return this->qty;
        }
};

class BinManager{
    private:
        InvBin *bin;
        int numBins;
    public:
        BinManager(){
            numBins = 0;
        }
        BinManager(int size, string d[], int q[]){
            bin = new InvBin[30];
            this -> numBins = size;
            for(int i=0; i<size; i++){
                bin[i].setDescription(d[i]);
                bin[i].setQty(q[i]);
            }
        }
        ~BinManager(){
            delete []bin;
        }
        string getDescription(int index){
            return this->bin[index].getDescription();
        }
        int getQuantity(string itemDescription){
            int index = this->search(itemDescription);
            return this->bin[index].getQty();
        }
        bool addParts(string itemDescription, int q){
            int binIndex = this->search(itemDescription);
            if(q<1 || binIndex>=this->numBins || binIndex<0) return false;
            this->bin[binIndex].setQty(this->bin[binIndex].getQty() + q);
            return true;
        }
        bool removeParts(string itemDescription, int g){
            int binIndex = this->search(itemDescription);
            if(g<1 || binIndex>=this->numBins || binIndex<0) return false;
            int newQty = this->bin[binIndex].getQty() - g;
            if(newQty >= 0){
                this->bin[binIndex].setQty(newQty);
                return true;
            } else {
```

Output:

```
Bin description     Bin Quantity
 0.Turkey              0
 1.Chocolate           0
 2.Egg                 0
 3.Yoyo                0
 4.ChickenBreast       0
 5.Gummy Bear          0
 6.Almond              0
 7.Cookie              0
 8.Spaghetti           0

What do U want to do?(Add/Remove/Quit)
Remove
Which one do U want to remove?(name)
Egg
How many?
3
No enough quantity to remove.

Bin description     Bin Quantity
 0.Turkey              0
 1.Chocolate           0
 2.Egg                 0
 3.Yoyo                0
 4.ChickenBreast       0
 5.Gummy Bear          0
 6.Almond              0
 7.Cookie              0
 8.Spaghetti           0

What do U want to do?(Add/Remove/Quit)
Add
Which one do U want to add?(name)
Egg
How many?
1000
```

```
Bin description     Bin Quantity
 0.Turkey              0
 1.Chocolate           0
 2.Egg                 1000
 3.Yoyo                0
 4.ChickenBreast       0
 5.Gummy Bear          0
 6.Almond              0
 7.Cookie              0
 8.Spaghetti           0

What do U want to do?(Add/Remove/Quit)
Remove
Which one do U want to remove?(name)
Egg
How many?
334

Bin description     Bin Quantity
 0.Turkey              0
 1.Chocolate           0
 2.Egg                 666
 3.Yoyo                0
 4.ChickenBreast       0
 5.Gummy Bear          0
 6.Almond              0
 7.Cookie              0
 8.Spaghetti           0

What do U want to do?(Add/Remove/Quit)
Quit

===============Bin Manager===============
Bin description     Bin Quantity
 0.Turkey              0
 1.Chocolate           0
 2.Egg                 666
 3.Yoyo                0
 4.ChickenBreast       0
 5.Gummy Bear          0
 6.Almond              0
 7.Cookie              0
 8.Spaghetti           0
```

```
            return false;
        }
    }
    int search(string itemDescription){
        for(int i=0; i<9; i++){
            if( itemDescription == binsDescription[i] ){
                return i;
            }
        }
        return -1;
    }
    int getnumBins(){
        return this->numBins;
    }
};
int getInt(){
    string sin;
    int num;
    getline(cin,sin);
    stringstream ss(sin);
    while(true){
        char ischar;
        if( ss >> num){
            if(!(ss >> ischar)){
                break;
            }
        }
        cout<<"You should input integer."<<endl;
        getline(cin,sin);
        ss.clear();
        ss << sin;
    }
    return num;
}
void displayReport(BinManager &binManager){
    int numBins = binManager.getnumBins();
    cout<<endl
        <<left<<setw(20)<<"Bin description"
        <<left<<setw(22)<<"Bin Quantity"<<endl;
    for(int i = 0; i < numBins; i++){
        cout<<right<<setw(2)<<i<<"."
            <<left<<setw(20)<<binManager.getDescription(i)
            <<left<<setw(22)<<binManager.getQuantity(binsDescription[i])<<endl;
    }
}

int main(int argc, char* argv[]){
    int binsQty[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
    BinManager binManager(9, binsDescription, binsQty);
    string sin;
    int indexSelect, num;
    while(true){
        displayReport(binManager);
        cout<<"\nWhat do U want to do?(Add/Remove/Quit)"<<endl;
        getline(cin,sin);
        if(sin == "Add"){
            cout<<"Which one do U want to add?(name)"<<endl;
            getline(cin,sin);
            cout<<"How many?"<<endl;
            num = getInt();
            if(!binManager.addParts(sin, num)){
                if(num < 0) cout<<"U can't enter negative number."<<endl;
                else cout<<"Wrong Index. \n";
            }
        } else if(sin == "Remove") {
            cout<<"Which one do U want to remove?(name)"<<endl;
            getline(cin,sin);
            cout<<"How many?"<<endl;
            num = getInt();
            if(!binManager.removeParts(sin, num)){
                if(num >binManager.getQuantity(sin)) cout<<"No enough quantity to remove.\n";
                else cout<<"Wrong name. "<<endl;
            }
        } else if(sin == "Quit") {
            break;
        } else {
            cout<<"I don't understand what U mean * A * ! \n";
        }
    }

    displayReport(binManager);
    return 0;
}
```