

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА  
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра  
теорії оптимальних процесів

**ВСТУП ДО ПРОГРАМУВАННЯ.  
НАВЧАЛЬНА (ОБЧИСЛЮВАЛЬНА) ПРАКТИКА**  
**Звіт виконання лабораторної роботи №7**  
**“Прості класи.”**

**Виконав(а):**

Студента групи ПМа-11  
Манько О.В

**Перевірили:**

доц. кафедри ТОП Мельничин А. В.,  
ас. кафедри ТОП Гавдяк М. М.

## Мета роботи:

Описати та реалізувати клас.

## Умова завдання.

Створити клас КВИТАНЦІЯ з наступними полями: назва послуги, номер квитанції, оплачена сума. Включити в нього необхідні конструктори та методи доступу до полів класу. Реалізувати методи введення і виведення в потік (файловий та консоль). Включити функції сортування за різними критеріями та пошуку за назвою. Вхідні дані зчитувати з текстового файлу. Програма повинна містити меню для перевірки всіх можливостей.

–

## Код програми:

**main.cpp**

```
#include "hdr.h"

int main()
{
    ifstream infile("data.txt");
    ofstream outfile("output.txt");

    string name;

    bool found = false;

    int count;
    int action;
    int choice;

    // перевірка чи є файл
    if (!infile.is_open()) {
        cout << "Error opening file!" << endl;
        return 1;
    }

    cout << "--- WELCOME ---\n";

    // введення кількості послуг які є у файлі
    do
    {
        cout << "How many receipts(>0) in file: ";
        cin >> count;
    } while (count <= 0);

    // створення динамічного масиву
    receipt* mass_receipts = new receipt[count];

    // цикл читає та записує дані в масив
    // після циклу файл закривається
    for (int i = 0; i < count; i++)
    {
        mass_receipts[i].data_input(infile);
    }
    infile.close();

    // меню програми
    cout << "\n--- Select action ---";
    do
    {
        cout << "\n1.Find by name";
        cout << "\n2.Sort and show";
        cout << "\nEnter choice: ";
        cin >> action;
    } while (action != 1 && action != 2);

    // якщо був вибір 1
    if (action == 1)
```

```

{
    cout << "Enter service name: ";
    cin >> name;

    cout << "\n--- Search Results ---\n";

    onfile << "--- Search Results ---\n\n";

    /*
    цикл який перевіряє кожне ім'я
    виводить його на екран
    якщо воно співпадає з введеним з клавіатури
    */
    for (int i = 0; i < count; i++)
    {
        if (mass_receipts[i].GetSN() == name)
        {
            mass_receipts[i].print();
            mass_receipts[i].fprint(onfile);
            found = true;
        }
    }
    /*
    вивід на екран повідомлення
    про те що у файлі нема послуги
    яку ввели з клавіатури
    */
    if (!found)
    {
        cout << "\nNo services with name " << name;
    }
    return 0;
}

// якщо був вибір 2
if (action == 2)
{
    do
    {
        cout << "\nSort by:";
        cout << "\n1.Name";
        cout << "\n2.ID";
        cout << "\n3.Cost";
        cout << "\nEnter choise: ";
        cin >> choise;
    } while (choise != 1 && choise != 2 && choise != 3);

    cout << "\n--- Sorted list ---\n\n";
    onfile << "--- Sorted list ---\n\n";
}

// сортування за назвою послуги
if (choise == 1)
{
    sort_by_name(mass_receipts, count);
}

// сортування за номером квитанції
if (choise == 2)
{
    sort_by_id(mass_receipts, count);
}

// сортування за оплаченою сумою
if (choise == 3)
{
    sort_by_amount(mass_receipts, count);
}

// цикл виводить дані в консоль і в файл
for (int i = 0; i < count; i++)
{
    mass_receipts[i].print();

```

```

        mass_receipts[i].fprintf(onfile);
    }

    // очищення пам'яті
    delete[] mass_receipts;

    return 0;
    /*
    */
}

```

## hdr.h

```

#pragma once
#include <iostream>
#include <string>
#include <fstream>

using namespace std;

class receipt
{
private:
    string service_name;    // назва послуги
    int id_receipt;         // номер квитанції
    float paid_amount;      // оплачена сума
public:

    string GetSN();
    int GetID();
    float GetPA();

    void SetSN(string);
    void SetID(int);
    void SetPA(float);

    // ввід даних з файлу
    void data_input(istream& inf);

    // вивід даних на консоль
    void print();

    // вивід даних в файл
    void fprintf(ofstream&);

    /*
    Методи сортування викликані friend бо:
        - Вони є глобальними
        - Мають доступ до приватних полів
    */

    friend void sort_by_name(receipt*, int);
    friend void sort_by_id(receipt*, int);
    friend void sort_by_amount(receipt*, int);

    // конструктор і деструктор(~)
    receipt();
    ~receipt();
};

```

## hdr.cpp

```
#include "hdr.h"

string receipt::GetSN()
{
    return service_name;
}

int receipt::GetID()
{
    return id_receipt;
}

float receipt::GetPA()
{
    return paid_amount;
}

void receipt::SetSN(string N)
{
    service_name = N;
}

void receipt::SetID(int A)
{
    id_receipt = A;
}

void receipt::SetPA(float P)
{
    paid_amount = P;
}

void receipt::data_input(istream& inf)
{
    inf >> service_name;
    inf >> id_receipt;
    inf >> paid_amount;
}

void receipt::print()
{
    cout << "Service name: " << service_name << "\n";
    cout << "Receipt ID: " << id_receipt << "\n";
    cout << "Paid amount: " << paid_amount << "\n\n";
}

void receipt::fprint(ofstream& onf)
{
    onf << "Service name: " << service_name << "\n";
    onf << "Receipt ID: " << id_receipt << "\n";
    onf << "Paid amount: " << paid_amount << "\n\n";
}

receipt::receipt()
{
    service_name = "Nothing";
    id_receipt = 0;
    paid_amount = 0;
}

receipt::~receipt()
{
}

/*
Зовнішній цикл відповідає за кількість проходів по масиву
Після кожного виконання циклу гарантується
що найбільший (або "найважчий") елемент з тих
що залишилися перейде в кінець масиву
Внутрішній цикл проходить по елементах, порівнюючи пари сусідів
j - це індекс поточного елементу
```

```

який порівнюється з наступним (j + 1)

(count - i - 1):
    (- 1): тому що всередині ми звертаємось до arr[j + 1].
    Якби ми йшли до самого кінця (count)
    то j + 1 вийшов би за межі масиву

    (- i): оскільки після кожного проходу зовнішнього циклу (i)
    найбільші елементи вже відсортувалися в кінець
    немає сенсу їх знову перевіряти

інші функції по сортуванню по аналогії
*/

void sort_by_name(receipt* arr, int count)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            // якщо лівий елемент більший за правий
            if (arr[j].service_name > arr[j + 1].service_name)
            {
                receipt temp = arr[j];    // зберігаємо лівий елемент у
тимчасову змінну
                arr[j] = arr[j + 1];      // на місце лівого ставимо правий
(менший)
                arr[j + 1] = temp;        // на місце правого ставимо
збережений лівий
            }
        }
    }
}

void sort_by_id(receipt* arr, int count)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            if (arr[j].id_receipt > arr[j + 1].id_receipt)
            {
                receipt temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void sort_by_amount(receipt* arr, int count)
{
    for (int i = 0; i < count - 1; i++)
    {
        for (int j = 0; j < count - i - 1; j++)
        {
            if (arr[j].paid_amount > arr[j + 1].paid_amount)
            {
                receipt temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

```

## Результат виконання програми:

```
--- WELCOME ---
How many receipts(>0) in file: 30

--- Select action ---
1.Find by name
2.Sort and show
Enter choise: 2

Sort by:
1.Name
2.ID
3.Cost
Enter choise: 1

--- Sorted list ---

Service name: Air_Conditioning
Receipt ID: 125
Paid amount: 620

Service name: Building_Insurance
Receipt ID: 116
Paid amount: 410

Service name: Cleaning
Receipt ID: 115
Paid amount: 190

Service name: Electricity
Receipt ID: 102
Paid amount: 780

Service name: Electricity
Receipt ID: 124
Paid amount: 450

Service name: Fire_Safety
Receipt ID: 114
Paid amount: 145.3

Service name: Full_Service
Receipt ID: 130
Paid amount: 1500

Service name: Garbage_Removal
Receipt ID: 104
Paid amount: 120

Service name: Gas
Receipt ID: 105
Paid amount: 640.2

--- WELCOME ---
How many receipts(>0) in file: 30

--- Select action ---
1.Find by name
2.Sort and show
Enter choise: 1
Enter service name: Water

--- Search Results ---
Service name: Water
Receipt ID: 101
Paid amount: 350.5

Service name: Water
Receipt ID: 107
Paid amount: 210.75

Service name: Water
Receipt ID: 119
Paid amount: 200
```

## Висновки:

На даній лабораторній роботі я навчився описувати та реалізовувати класи в мові програмування C++.