

Тема: ООП. Вступ до ООП.

Мета:

Загальне завдання.

Для предметної галузі з розділу “Розрахункове завдання / Індивідуальні завдання” розробити два класи: - клас, що відображає сутність “базового класу”, у тому числі: - конструктор за замовчуванням, копіювання та конструктор з аргументами (реалізація конструкторів повинна бути продемонстрована за допомогою списків ініціалізацій); - деструктор; - клас, що має у собі динамічний масив об’єктів базового класу та має в собі методи додавання, видалення елемента, отримання елемента по індексу (або ідентифікатору), вивід усіх елементів на екран. Рекомендовані сигнатури методів: - додавання:

```
void CList::addPhone(Phone& phone);
```

- видалення:

```
void CList::removePhone(int index);
```

- отримання по індексу:

```
CPhone& CList::getPhone(int index);
```

- виведення усіх елементів (при цьому цей метод повинен викликати метод CList::getPhone(int index), щоб не було дублювання коду):

```
void CList::showAll();
```

- метод 1 обходу колекції. Приклад сигнатури такого методу (У наведеному прикладі реалізоване завдання пошуку самого дешевого телефону з заданою діагоналлю (повертається один телефон):

```
CPhone& findCheapestPhone(float diagonal);
```

Індивідуальне завдання.

Обрати прикладну галузь за варіантом відповідно до номера у журналі групи.

Хід роботи

```
class Employee
{
private:
    string company;
    string mail;
    string fullName;
    string characteristic;
    int workExperience;
    bool insurance;
```

Рис.1 - поля базового класу.

```
Employee::Employee(const Employee& obj) :company(obj.company), mail(obj.mail), fullName(obj.fullName),
characteristic(obj.characteristic), workExperience(obj.workExperience), insurance(obj.insurance)
{
    cout << "\n\t\tThere was a COPY constructor here\n";
}
```

Рис.2 - конструктор копіювання зі списком ініціалізації.

```
public:
    Employee();
    Employee(const string company, const string mail, const string fullName, const string
    Employee(const Employee &obj);
    ~Employee();

    Employee& operator=(const Employee& obj);
    friend bool operator==(const Employee& obj1, const Employee& obj2) { ... }
    friend bool operator!=(const Employee& obj1, const Employee& obj2) { ... }
    friend std::ostream& operator<< (std::ostream& out, const Employee& obj) { ... }
    friend std::istream& operator>> (std::istream& in, Employee& obj) { ... }

    string printEmployee();

    void setCompany(string company);
    void setMail(string mail);
    void setFullName(string fullName);
    void setCharac(string charac);
    void setWorkExp(int workExp);
    void setInsurance(bool insurance);

    string getCompany();
    string getMail();
    string getFullName();
    string getCharac();
    int getWorkExp();
    bool getInsurance();
```

Рис.3 - методи для базового класу.

```

class List
{
private:
    std::vector<Employee> list;

public:
    void addEmployee(Employee& employee);
    void addObjects();
    void removeEmployee(const int index);
    void showAll();

    int getSize();
    Employee& getEmployee(const int index);
    Employee& getEmployeeByWorkExp(const int workExp);
    string check(regex reg);

    Employee& operator[](const int index);
    friend std::ostream& operator << (std::ostream& out, const List& obj) { ... }

    void readFromFile();
    void writeToFile();
};

```

Рис.4 - для динамічного масиву було використано бібліотеку vector, на рисунку зображені методи для роботи з класом списком.

При розробці програми використані рекомендовані сигнатури методів.

```

    A list of employees.Select an action from the following.

1.Write list to screen.
2.Generate and add employee.
3.Delete employee by index.
4.Search for employees with insurance.
5.Reading employees from a file.
6.Writing employees in file.
0.Exit.

Your choice: █

```

Рис.5 - початкове меню(для реалізації створено клас Helper).

```

        A list of employees.Select an action from the following.

1.Write list to screen.
2.Generate and add employee.
3.Delete employee by index.
4.Search for employees with insurance.
5.Reading employees from a file.
6.Writing employees in file.
0.Exit.

Your choice: 2

        There was a DEFAULT constructor here

Create a new employee now.
    Enter full name : Dan Danan Da
    Enter mail : DanDa@gmail.com
    Enter company name : Google
    Describe the positive qualities in 3 words : Happy,Sad,Fun
    Enter work experience in months : 10
    Availability of insurance (1 - yes, 0 - no) : 1

        There was a COPY constructor here
A new employee has been added to the list. Would you like to add another one?
Your choice (1 - yes, 0 - no):0

        There was DESTRUCTOR here

Successful.

```

Рис.6 - приклад створення та додавання елементу у список.

```

Your choice: 1

Company:Google
Mail:DanDa@gmail.com
Fullname:Dan Danan Da
Characteristic:Happy,Sad,Fun
Work experience in years:10
Insurance:1

Successful.

```

Рис.7 - приклад використання методу для виводу всіх елементів

```
Your choice: 3

Choose index(0-2):1

                There was DESTRUCTOR here

Successful.

                A list of employees.Select an action from the following.

1.Write list to screen.
2.Generate and add employee.
3.Delete employee by index.
4.Search for employees with insurance.
5.Reading employees from a file.
6.Writing employees in file.
0.Exit.

Your choice: 1

Company:Yandex
Mail:Nenet@gmail.com
Fullname:Nenet
Characteristic:Good
Work experience in years:10
Insurance:1

Successful.
```

Рис.8 - приклад використання методу видалення елемента зі списку. Спочатку додали ще один елемент, який зараз вивели у консоль

```
Your choice: 4

Company:Yandex
Mail:Nenet@gmail.com
Fullname:Nenet
Characteristic:Good
Work experience in years:10
Insurance:1

Successful.
```

Рис.9 - приклад використання методу пошуку співробітників зі страхуванням.

Your choice: 5

There was DESTRUCTOR here

Here was a constructor with PARAMETRS

There was a COPY constructor here

There was DESTRUCTOR here

Here was a constructor with PARAMETRS

There was a COPY constructor here

There was DESTRUCTOR here

Here was a constructor with PARAMETRS

There was a COPY constructor here

There was a COPY constructor here

There was a COPY constructor here

There was DESTRUCTOR here

There was DESTRUCTOR here

There was DESTRUCTOR here

Here was a constructor with PARAMETRS

There was a COPY constructor here

There was a COPY constructor here

There was a COPY constructor here

There was a COPY constructor here

There was DESTRUCTOR here

There was DESTRUCTOR here

There was DESTRUCTOR here

Рис. 10 - приклад зчитування з файлу.

Your choice: 1

Company:example
Mail:rabotnik141gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:0

Company:example
Mail:rabotnik500gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:0

Company:example
Mail:rabotnik778gmail.com
Fullname:example
Characteristic:example
Work experience in years:10
Insurance:0

Company:example
Mail:rabotnik264gmail.com
Fullname:example
Characteristic:example
Work experience in years:27
Insurance:1

Company:example
Mail:rabotnik881gmail.com
Fullname:example
Characteristic:example
Work experience in years:25
Insurance:1

Company:example
Mail:rabotnik591gmail.com
Fullname:example
Characteristic:example
Work experience in years:17
Insurance:0

Successful.

Рис.11 - вміст списку після зчитування.

```
Your choice: 6

Successful.

      A list of employees.Select an action from the following.

1.Write list to screen.
2.Generate and add employee.
3.Delete employee by index.
4.Search for employees with insurance.
5.Reading employees from a file.
6.Writing employees in file.
0.Exit.

Your choice: 1

Company:Dark
Mail:Sadasd@gmail.com
Fullname:Sadad
Characteristic:Bad
Work experience in years:10
Insurance:1

Company:Google
Mail:Bmth@gmail.com
Fullname:Bmth
Characteristic:Good
Work experience in years:9
Insurance:0

Successful.
```

Рис.12 - вміст консолі і запис у файл.



text – Блокнот

Файл	Правка	Формат	Вид	Справка
<hr/>				
Dark	Sadasd@gmail.com	Sadad	Bad	10 1
Google	Bmth@gmail.com	Bmth	Good	9 0

Рис.12.1 - вміст файлу після запису.

Також виконані додаткові умови завдання, а саме:

- усі поля “базового класу” повинні бути приватними та мати публічні гетери та сетери (модифікатори доступу), використовувати механізм інкапсуляції;
- усі функції, що не повинні змінювати поля поточного об’єкта, повинні бути константними;
- усі аргументи функцій, що не змінюються, по можливості також повинні бути константними. Якщо їх не можна зробити константними, у такому разі повинно бути обґрунтування цього;
- у класі-списку метод додавання елемента не повинен вводити дані з клавіатури або файлу, а повинен приймати вже готовий об’єкт для додавання. Метод вводу даних має бути відокремленим;

Висновок: в даній лабораторній роботі було створено три класи: базовий, який описує співробітника, клас список в якому реалізовано методи з його роботи і додатковий клас помічник. Було використано механізм інкапсуляції для розробки даної програми.