МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ “ХПІ”

Кафедра “Обчислювальна техніка та програмування”

Розрахункове завдання з дисципліни

«Програмування ч.2»

Пояснювальна записка

ЄСПД ГОСТ 19.404-79(СТЗВО-ХПІ-2.01-2018 ССОНП)

Розробники

###### Виконав: Саламатін Р.О.

студент групи КІТ-120Д

Перевірив: Пасько Д. А.

Харків 2021

ЗАТВЕРДЖЕНО

ЄСПД ГОСТ 19.404-79(СТЗВО-ХПІ-2.01-2018 ССОНП)

Розрахункове завдання з дисципліни

«Програмуваня ч.2»

Пояснювальна записка

Листів 15

ЄСПД ГОСТ 19.404-79(СТЗВО-ХПІ-2.01-2018 ССОНП)

Харків 2021

**Вступ**

**Тема роботи:**

Розробка інформаційно-довідкової системи.

**Мета роботи:**

Закріпити отримані знання з дисципліни «Програмування ч.2» шляхом використання типового комплексного завдання.

1. Призначення та галузь застосування;
2. Постановка завдання до розробки;
3. Опис вхідних та вихідних даних;
4. Опис складу технічних та програмних засобів;
5. Список джерел інформації;
6. Додаток, який складається з розробленого коду.

**Призначення та галузь застосування**

**Об'єктно-орієнтоване програмування** — одна з [парадигм](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка розглядає програму як множину [«об'єктів»](https://uk.wikipedia.org/wiki/%D0%9E%D0%B1'%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), що взаємодіють між собою. Основу ООП складають чотири основні концепції: [інкапсуляція](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D1%96%D1%8F), [успадкування](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [поліморфізм](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BC%D0%BE%D1%80%D1%84%D1%96%D0%B7%D0%BC_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та абстракція. Однією з переваг ООП є краща [модульність](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) програмного забезпечення (тисячу функцій процедурної мови, в ООП можна замінити кількома десятками класів із своїми методами). Попри те, що ця парадигма з'явилась в [1960-х](https://uk.wikipedia.org/wiki/1960-%D1%82%D1%96) роках, вона не мала широкого застосування до [1990-х](https://uk.wikipedia.org/wiki/1990-%D1%82%D1%96), коли розвиток комп'ютерів та комп'ютерних мереж дав змогу писати надзвичайно об'ємне і складне програмне забезпечення, що змусило переглянути підходи до написання програм.

**Постановка завдання до розробки**

При виконанні завдання з розробки інформаційно-довідкової системи необхідно виконати наступне:

1. З розділу «Розрахункове завдання/Індивідуальне завдання», відповідно до варіанта завдання, обрати прикладну галузь(варіант 3).
2. Для прикладної галузі розробити розгалужену ієрархію класів, що описана у завданні та складається з одного базового класу та двох спадкоємців. Клас повинен мати перевантажені оператори.
3. Розробити клас-список, що буде включати до себе масив вказівників до базового класу. А також базові методи роботи з списком: а) очистка списку б) відображення списку в) додавання/видалення/оновлення.
4. Розробити клас-контролер, що буде включати колекцію розроблених класів, та наступні методи роботи з колекцією: а) читання даних з файлу б) запис даних у файл.
5. Розробити клас меню, який має відображати діалогове меню для демонстрації реалізованих функцій класу контролера.
6. Виконати перевірку вхідних даних за допомогою регулярних виразів.
7. Оформити документацію: пояснювальну записку.

**Опис вхідних та вихідних даних**

**Вхідні дані:**

1. Ім'я працівника (string fullName).
2. Компанія в котрій працює працівник(string company).
3. Електронна поштова скринька (string mail).
4. Характеристика працівника (string characteristic).
5. Стаж працівника у місяцях (int workExperience).
6. Наявність страховки у працівника (bool insurance).

**Вихідні дані:**

1. Список, котрий складається із Працівників спадкоємців, а саме Програмістів та Перекладачв

**Опис складу технічних та програмних засобів**

1. Створено базовий клас та двох спадкоємців відповідно до свого індивідуального завдання(7-ий варіант).
2. Створено методи читання даних з файлу та запис даних у файл .
3. Створено методи додавання та видалення, та редагування елементу за вибором користувача.
4. Створено функцію меню для зручного користування програмою.
5. Створено методи пошуку працівника зі страховкою.
6. Створено метод сортування учнів по класу (Зростанню і зменшенням).
7. Застосовані регулярні вирази для перевірки вхідних даних.
8. Виконано перевантаження операторів.

Реалізація програми для виконання завдання

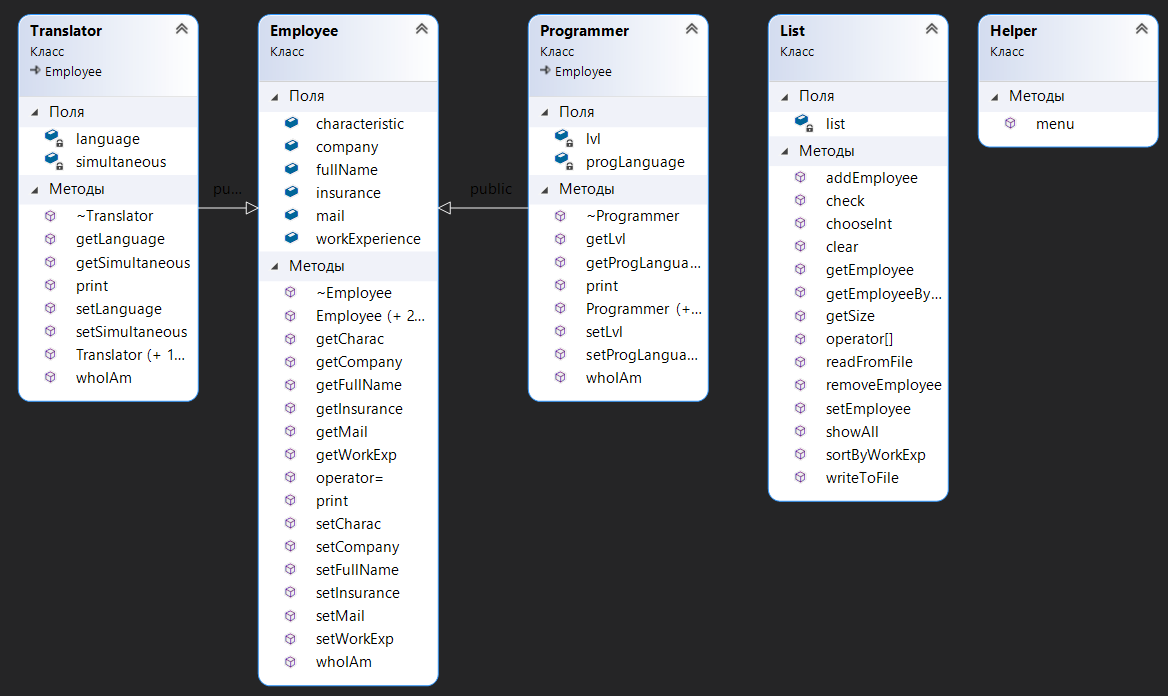
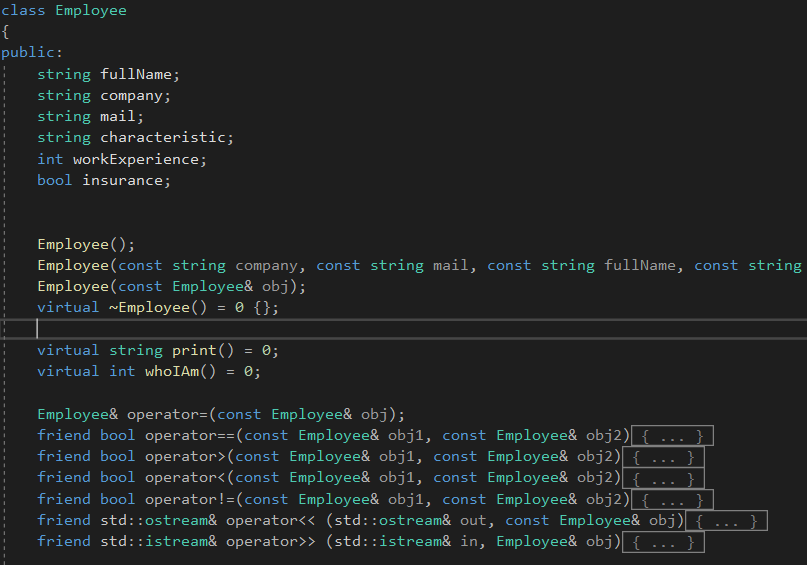
Рис.1 Діаграма класів для даної програми

Рис 2. Базовий абстрактний клас Працівника з віртуальними методами та реалізованими перевантаженими методами порівняння елементів(які впорядковуються через поле стажу роботи)

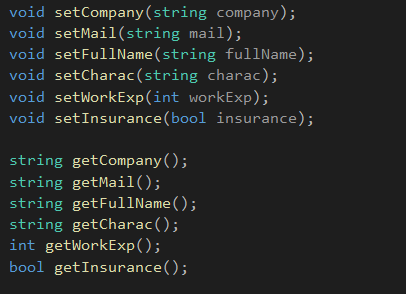


Рис.3 Гетери та сетери базового класу Працівник

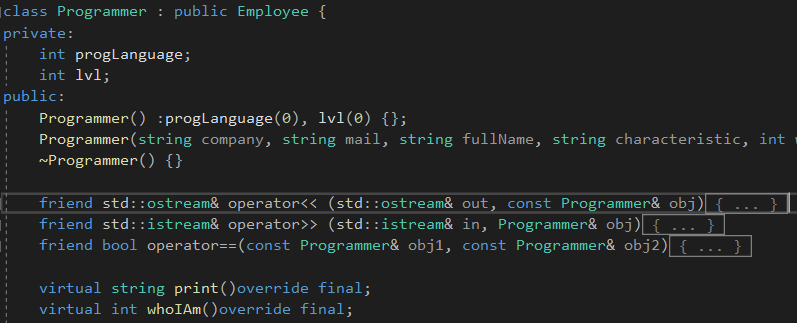
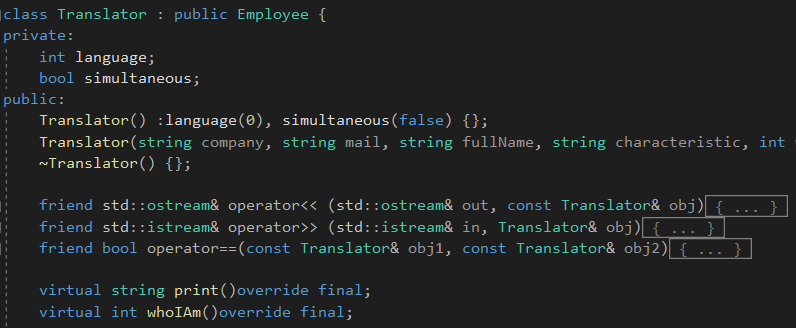
Рис.4 Клас спадкоємець Програміст, який містить захищені поля,конструктори(базовий та з полями) та деструктор, а також перевантажені методи

Рис.5 Клас спадкоємець Перекладач, який містить захищені поля,конструктори(базовий та з полями) та деструктор, а також перевантажені методи

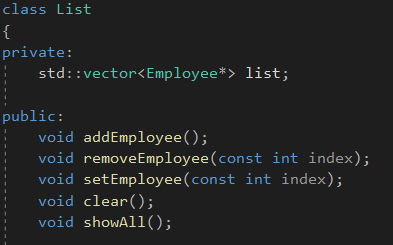
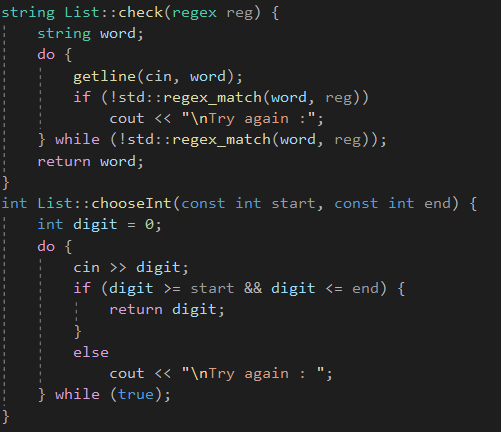
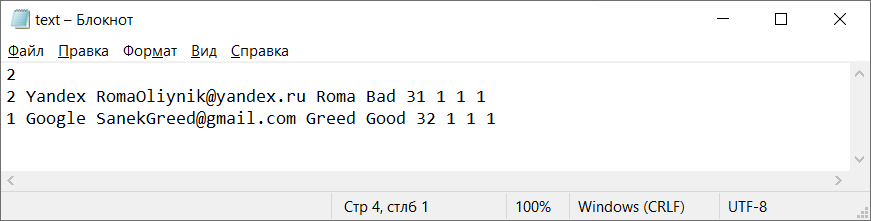
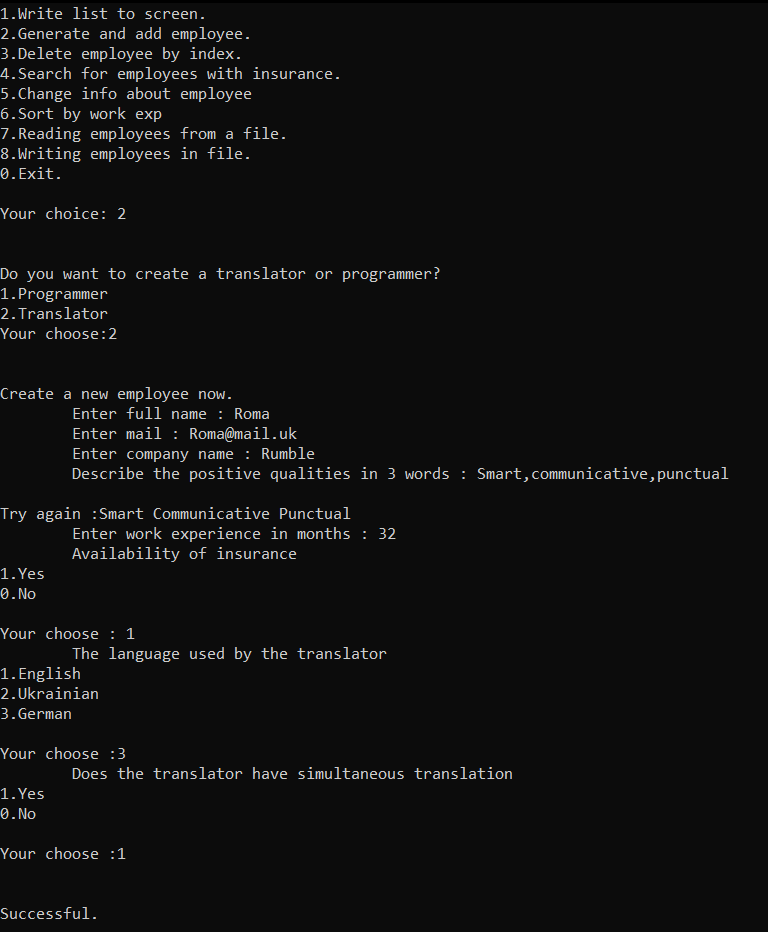
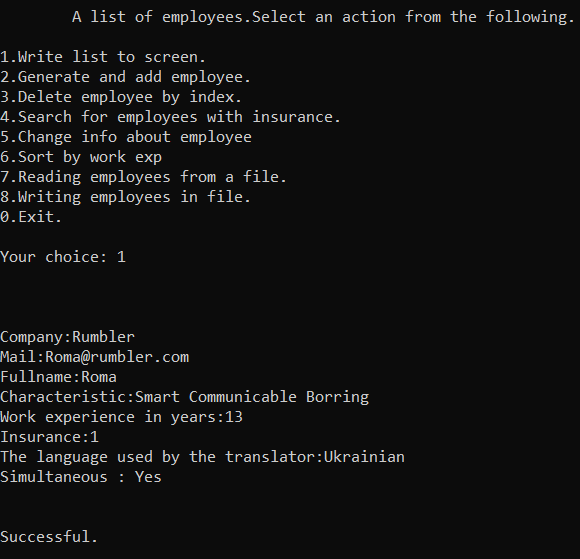
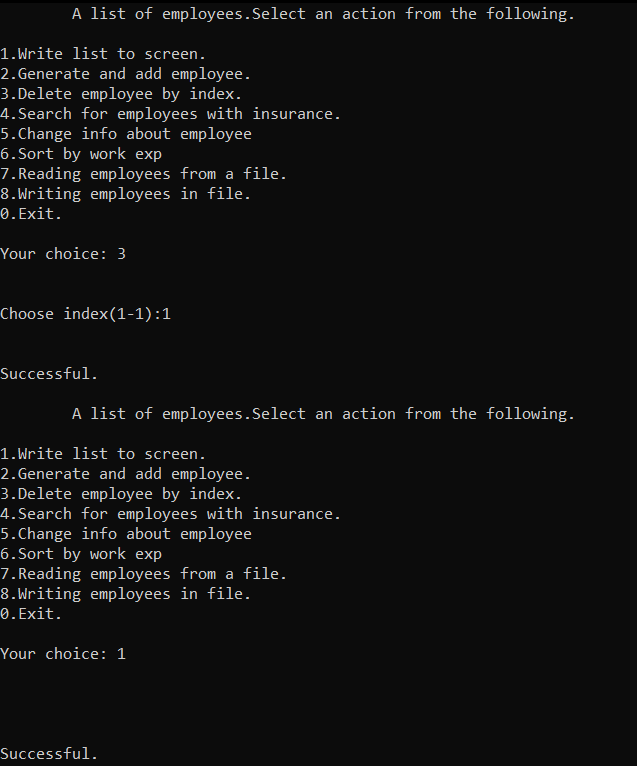
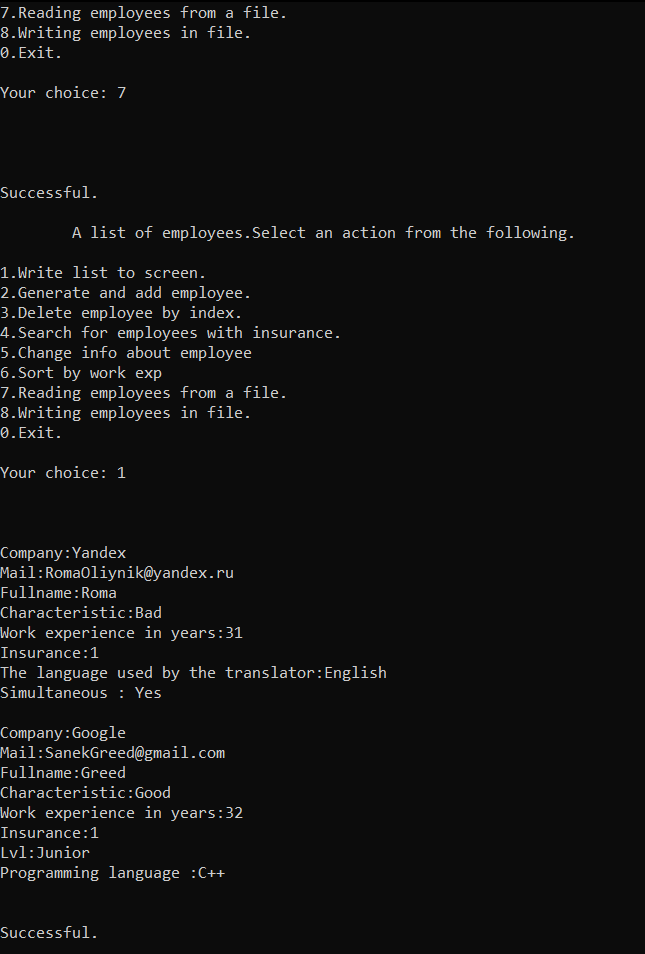
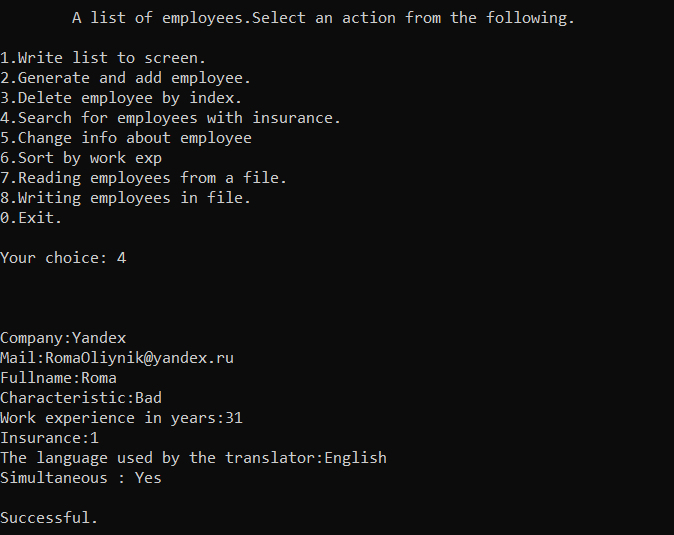
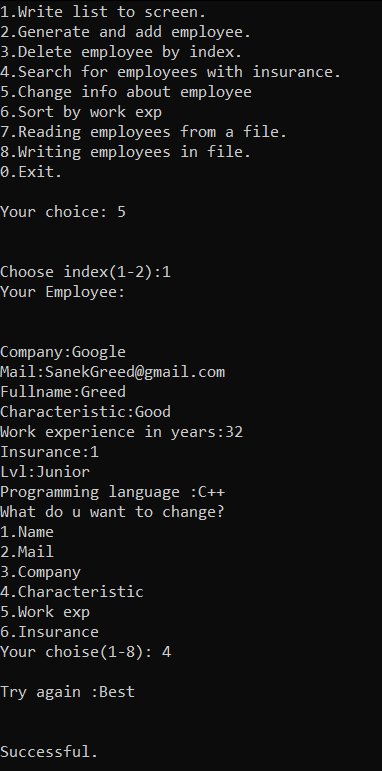
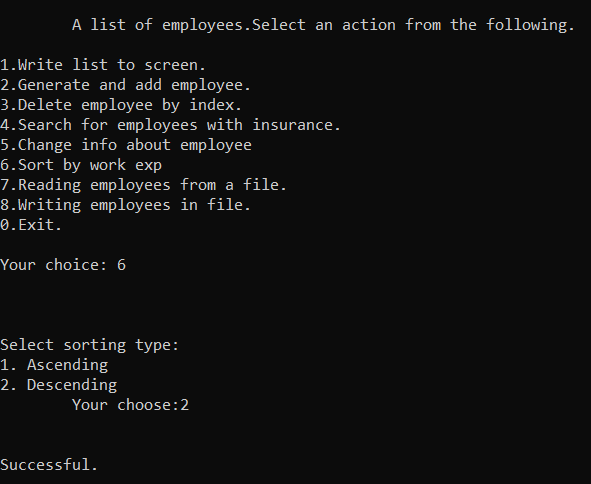
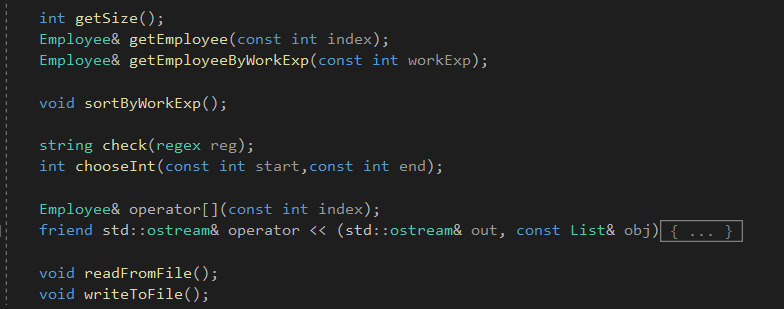


Рис.6 Клас Список, котрий містить колекцію Vector з вказівниками   
на базовий абстрактний клас Працівника, а також має методи роботи з ними  
  
  
  
  
  
  
  
  
  
  
  
  
Рис.7 Методи для роботи зі Списком  
  
  
Рис.8 Реалізація методів для генерації та перевірки вхідних даних з клавіатури  
  
Результати роботи програми:  
  
Рис.8 Вміст файлу зі списком працівників  
(Перша змінна зберігає у собі к-сть співробітників у списку,  
 кожен наступний рядок інформацію про кожного працівника)  
  
Рис.9 Приклад створення нового елементу до списку  
  
  
  
 **  
Рис.10 Приклад виводу списку у консоль  
  
  
Рис.11 Результат видалення та перевірки у списку  
  
  
Рис.12 Результат зчитування з файлу та відображення у консоль(ці дані були записані за допомогою цієї ж програми при попередньому запуску  
  
Рис.13 Результат пошуку Працівників зі страхуванням   
  
  
  
Рис.14 Приклад роботи методу для зміни створених змінних( у процесі покращення )  
  
  
Рис.15 Сортування за стажем роботи**

**Висновок**

Закріплено отримані знання з дисципліни «Програмування» шляхом використання типового комплексного завдання. Створено програму для утримання працівників у списку, котрий має у собі Програмістів та Перекладачів. Для цього було створено базовий абстрактний клас Працівник та його методи. Для нього створено спадкоємців, клас Список та клас Помічник для виводу списку та роботи з ним.

**Список джерел інформації**

1. Ашарина, И.В. Объектно-ориентированное программирование в С++: лекции и упражнения: Учебное пособие для вузов / И.В. Ашарина. - М.: РиС, 2015. - 336 c.
2. Ашарина, И.В. Язык С++ и объектно-ориентированное программирование в С++. Лабораторный практикум: Учебное пособие для вузов / И.В. Ашарина, Ж.Ф. Крупская. - М.: ГЛТ , 2015. - 232 c.
3. Лафоре, Р. Объектно-ориентированное программирование в C++. Классика Computer Science / Р. Лафоре. - СПб.: Питер, 2013. - 928
4. Лабораторний практикум С++ / Давидов В., Молчанов Г.І.  
     
     
     
     
     
   Код: