

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ(ІСТ)

Звіт

з лабораторної роботи № 8 з дисципліни
«Основи програмування. Частина 2»

„Побудова та використання програми для адміністрування”

Виконав(ла)

ІІІ-43 Дяченко Д.О.

(шифр, прізвище, ім'я, по батькові)

Перевірів(ла)

Вітковська І. І.

(прізвище, ім'я, по батькові)

Київ 2025

Мета: розробити програму для адміністрування даних. Дослідити типи зв'язків між класами.

Варіант 5

БІРЖА ПРАЦІ: ВЕДЕННЯ ОСОБОВИХ СПРАВ БЕЗРОБІТНИХ В ЦЕНТРИ ЗАЙНЯТОСТІ

Функціональні вимоги до програмного забезпечення

1. Управління вакансіями і резюме
 - 1.1. Можливість додавати вакансію і резюме до категорії
 - 1.2. Можливість видаляти вакансію і резюме з категорії
 - 1.3. Можливість змінювати дані вакансій і резюме
 - 1.4. Можливість переглянути дані конкретної вакансії і резюме
 - 1.5. Можливість переглянути відсортований список всіх вакансій
 - 1.6. Можливість переглянути відсортований список всіх резюме
2. Управління безробітними
 - 2.1. Можливість додавати безробітних
 - 2.2. Можливість видаляти безробітних
 - 2.3. Можливість змінювати дані безробітних
 - 2.4. Можливість переглянути дані конкретного безробітного
 - 2.5. Можливість переглянути список всіх безробітних
 - 2.5.1. Можливість відсортувати список по імені
 - 2.5.2. Можливість відсортувати список по прізвищу
3. Управління фірмами замовниками
 - 3.1. Можливість додавати замовників
 - 3.2. Можливість видаляти замовників
 - 3.3. Можливість змінювати дані замовників
 - 3.4. Можливість переглянути дані конкретного замовника
 - 3.5. Можливість переглянути список всіх замовників
 - 3.5.1. Можливість відсортувати список по імені
 - 3.5.2. Можливість відсортувати список по прізвищу
4. Пошук
 - 4.1. Можливість пошуку по ключовому слову серед вакансій
 - 4.2. Можливість пошуку по ключовому слову серед безробітних

Вихідний код програми (мова програмування C#):

<https://github.com/DanyloDiachenko/OOP-Sharp>

```
using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
    static void Main(string[] args)
    {
        TestJobManagementSystem();
    }

    static void TestJobManagementSystem()
    {
        JobManagementSystem system = new JobManagementSystem();

        Category itCategory = new Category("IT");
        system.AddCategory(itCategory);

        Company techCorp = new Company("TechCorp", "info@techcorp.com");
        system.AddCompany(techCorp);

        Company designStudio = new Company("DesignStudio", "info@designstudio.com");
        system.AddCompany(designStudio);

        Vacancy developerVacancy = new Vacancy("Розробник", "Розробка ПЗ", techCorp);
        system.AddVacancyToCategory("IT", developerVacancy);

        Vacancy designerVacancy = new Vacancy("Дизайнер", "Розробка UI/UX", designStudio);
        system.AddVacancyToCategory("IT", designerVacancy);

        Resume ivanResume = new Resume("Резюме Івана", "C#, .NET", "Іван", "Петренко", "ivan@example.com");
        system.AddResumeToCategory("IT", ivanResume);

        Resume mariaResume = new Resume("Резюме Марії", "Photoshop, Figma", "Марія", "Іваненко",
"maria@example.com");
        system.AddResumeToCategory("IT", mariaResume);

        Console.WriteLine("=== Тестування методів JobManagementSystem ===");
    }
}
```

```
Console.WriteLine("Додано вакансію 'Розробник' та резюме 'Резюме Івана' до категорії IT.");

system.RemoveVacancyFromCategory("IT", developerVacancy);
system.RemoveResumeFromCategory("IT", ivanResume);
Console.WriteLine("Видалено вакансію 'Розробник' та резюме 'Резюме Івана' з категорії IT.");

system.UpdateVacancy(developerVacancy, "Старший розробник", "Розробка ПЗ", techCorp);
system.UpdateResume(ivanResume, "Резюме Івана", "C#, .NET, SQL", ivanResume.JobSeeker);
Console.WriteLine("Оновлено вакансію до 'Старший розробник' та резюме з новими навичками 'C#, .NET, SQL'.");

List<Vacancy> sortedVacancies = system.GetSortedVacancies();
List<Resume> sortedResumes = system.GetSortedResumes();
system.PrintVacancies(sortedVacancies);
system.PrintResumes(sortedResumes);

JobSeeker petro = new JobSeeker("Петро", "Сидоренко", "petro@example.com");
system.AddJobSeeker(petro);
system.UpdateJobSeeker(petro, "Петро", "Сидоренко", "petro.updated@example.com");
system.RemoveJobSeeker(petro);
Console.WriteLine("Додано шукача 'Петро', оновлено контакт 'Петро', видалено 'Петро'.");

JobSeeker anton = new JobSeeker("Антон", "Коваленко", "anton@example.com");
JobSeeker maria = new JobSeeker("Марія", "Коваленко", "maria@example.com");
system.AddJobSeeker(anton);
system.AddJobSeeker(maria);
Console.WriteLine("=====");
List<JobSeeker> sortedByName = system.GetSortedJobSeekersByName();
system.PrintJobSeekers(sortedByName);
Console.WriteLine("=====");
List<JobSeeker> allSeekers = system.GetAllJobSeekers();
system.PrintJobSeekers(allSeekers);
Company newCompany = new Company("NewTech", "newtech@example.com");
system.AddCompany(newCompany);
system.UpdateCompany(techCorp, "TechCorp Ltd", "contact@techcorp.com");
system.RemoveCompany(newCompany);
Console.WriteLine("Додано 'NewTech', оновлено 'TechCorp' до 'TechCorp Ltd', видалено 'NewTech'.");

Company? foundCompany = system.GetCompany("TechCorp Ltd");
List<Company> allCompanies = system.GetAllCompanies();
List<Company> sortedCompanies = system.GetSortedCompaniesByName();
```

```
Console.WriteLine($"Отримано компанію: {foundCompany?.Name}");
system.PrintCompanies(allCompanies);
system.PrintCompanies(sortedCompanies);

List<Vacancy> foundVacancies = system.SearchVacancies("розробник");
List<JobSeeker> foundSeekers = system.SearchJobSeekers("Іван");
system.PrintVacancies(foundVacancies);
system.PrintJobSeekers(foundSeekers);
}
}
```

Випробування програми:

Висновок: для реалізації поставленої задачі було використано зв'язки агрегації та використання. Програма працює коректно, нестандартних ситуацій не виникає. Поставлена задача виконана успішно.