Національний університет “Києво-Могилянська академія”

## Лабораторна робота № 1

з дисципліни

“Алгортитми і структури даних**”**

Роботу виконали:

*Рибка Максим Костянтинович*

студент І курсу НаУКМА  
Факультет Інформатики  
Спеціальність: інженерія програмного забезпечення

*Мокрий Михайло Вікторович*

студент І курсу НаУКМА  
Факультет Інформатики  
Спеціальність: інженерія програмного забезпечення

Київ — 2018

**ПЛАН**

1. Постановку задачі.
2. Розподіл ролей в групі.
3. Код гри (основні класи).
4. Вирішення основних задач.
5. Проблеми в роботі.
6. Висновки.
7. **Постановка задачі**

Написати програму, що буде формувати список студентів та викладачів університету НаУКМА.

Відповідно мають бути реалізовані такі можливості роботи, як:

1. Створити/видалити/редагувати факультет.
2. Створити/видалити/редагувати кафедру факультету.
3. Додати/видалити/редагувати студента/викладача до кафедри.
4. Знайти студента/викладача за ПІБ, курсом або групою.
5. Вивести всіх студентів впорядкованих за курсами.
6. Вивести всіх студентів/викладачів факультету впорядкованих за алфавітом.
7. Вивести всіх студентів кафедри впорядкованих за курсами.
8. Вивести всіх студентів/викладачів кафедри впорядкованих за алфавітом.
9. Вивести всіх студентів кафедри вказаного курсу.
10. Вивести всіх студентів кафедри вказаного курсу впорядкованих за алфавітом.

Вимоги:

* Повинні бути реалізовані усілякі можливі варіанти захисту від невірного введення даних, або заборонених дій.
* При написанні програми необхідно обов’язково використовувати об'єкти і обмін даними між ними.
* Продумати ієрархію класів.
* Вся інформація вводиться з клавіатури.
* Для роботи користувача повинно пропонуватися меню з набором можливих дій.
* Робота може виконуватися в групі, максимум з 2**студентів**.
* До роботи має бути доданий звіт про виконання лабораторної роботи з описом усіх написаних класів, а також реалізованих можливостей.
* При груповій роботі до звіту має бути доданий чіткий розподіл функцій та повноважень, що були реалізовані кожним учасником групи окремо.

**2. Розподіл ролей у группі**

1. Розробка ієрархії – Максим Рибка

2. Методи сортування – Мокрий Михайло

**3. Код та структура лабораторної роботи**

**Структура лабораторної роботи:**

Arrays

Faculty

Faculty

Faculty

Specialty

Department

Specialty

Department

Professor

Professor

Student

Professor

Student

Student

**Код працюючої лабораторної роботи:**

Лабораторна має 10 класів (включно з тестером і абстрактним класом)

[class Tester](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory 1/src/Tester.java)

Клас реалізує виконання програми.

[lab\_package:](https://github.com/MikhailoMokryy/Laboratory1/tree/master/Laboratory%201/src/lab_package)

[abstract class People](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory 1/src/lab_package/People.java)

[Абстрактний клас який характрезує клас](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory 1/src/lab_package/People.java)

[Student та Professor.](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory 1/src/lab_package/People.java)

[class Student extends People](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/Student.java)

Клас має конструктор з характериками студента, також

кожному студенту присвоюється один факультет.

class Professor extends People

Клас має конструктор з характериками викладача, також

Кожному студенту викладачу присвоюється одна кафедра.

[class Faculty](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/Faculty.java)

У цьому класі створюється масив з об'єктів класу Specialty та Department, а також методи створення, редагування, видалення факультетів і кафедр.

[class FacultyObject](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/FacultyObject.java)

Цей клас реалізує роботу зі студентами, тобто методи створення, редагування, видалення викладачів і студентів, а також сортування.

[class Department extends FacultyObject](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/Department.java)

Клас містить конструктор і поля об’єкта “кафедра”.

[class Specialty extends FacultyObject](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/Specialty.java)

Клас містить конструктор і поля об’єкта “спеціальність”.

[class Arrays](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory%201/src/lab_package/Arrays.java)

Всередині цього класу створюються об’єкти класу Student і Professors, також створюються об’єкти класів Faculty, Specialty та Department і присвоюється студентам і викладачам.

[final class DataInput](https://github.com/MikhailoMokryy/Laboratory1/blob/master/Laboratory 1/src/lab_package/DataInput.java)

Клас реалізує зчитування інформації з консолі, захист від “NumberFormatException” та інших помилок.

**4.Вирішення основних задач**

Основною поставленою нами задачею була реалізація коректної роботи всіх методів та зв’язок між класами, який би змінювався в залежності від вибору користувача. Також, не менш важливим фактором був захист від некоректного вводу даних.

**5.Проблеми в роботі**

1. Реалізація

Головною проблемою була побудова такої структури програми, яка б правильно працювала та задовольняла всі поставлені умови.

2. Втрата даних.

У ході написання коду проблемним виявилось створення та видалення кафедр\студентів\викладачів та коректного додавання\видалення їх з масивів.

3.Робота з студентами\викладачами.

Так як ми записували всіх студентів та викладачів у окремі масиви, проблематичним було витягнення об’єкту з цих масивів для редагування їхніх даних.

**6.Висновки, фото працюючої програми**

За час виконання лабораторної роботи ми отримали досвід працювання в команді, спробували знаходити альтернативні рішення при різних баченнях реалізації різних методів, зуміли попрацювати з ієрархією класів.