Лабораторна робота №1

З курсу "Алгоритми та структкри данних"

Національний Університет України
"Києво-Могилянська Академія"
Петрик Ярослав, Рожко Андрій
м. Київ 2019

Постановка задачі

Написати програму, що буде формувати список студентів та викладачів університету НаУКМА.
Відповідно мають бути реалізовані такі можливості роботи, як:

- Створити/видалити/редагувати факультет.
- Створити/видалити/редагувати кафедру факультета.
- Додати/видалити/редагувати студента/викладача до кафедри.
- Знайти студента/викладача за ПІБ, курсом або групою.
- Вивести всіх студентів впорядкованих за курсами.
- Вивести всіх студентів/викладачів факультета впорядкованих за алфавітом.
- Вивести всіх студентів кафедри впорядкованих за курсами.
- Вивести всіх студентів/викладачів кафедри впорядкованих за алфавітом.
- Вивести всіх студентів кафедри вказаного курсу.
- 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 обовязково 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 звіта має бути доданий чіткий розподіл функцій та повноваженнь, що були реалізовані кожним учасником групи окремо.

Розподіл ролей

Андрій Рожко:

- Архітектура об’єктів та їх взаємодій
- Модифікація (внесення, видалення, редагування) елементів за допомогою командного інтерфейсу
- Реалізація консольного інтерфейсу
- Code Review

Петрик Ярослав:

- Реалізація пошуку
- Реалізація сортування та виведення данних
- Організація структур данних (ArrayList в данному випадку)
- Прототипування та впровадження основ взаємодії консольного інтерфейсу з користувачем
- Code Review

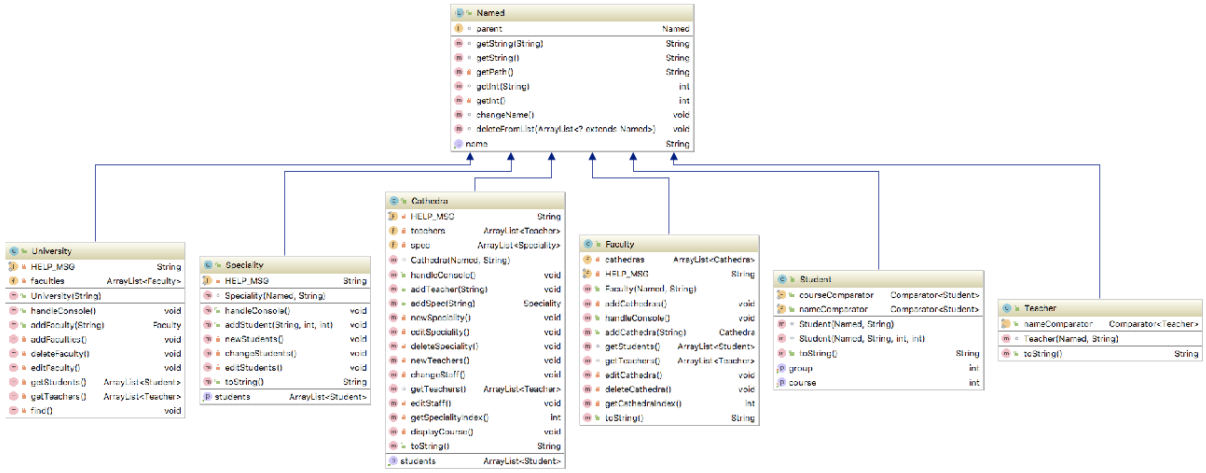
Опис реалізованих можливостей

- До структури університету можливо додати/видалити/редагувати:
  - Факультети
  - Кафедри факультетів
  - Спеціальності кафедр
  - Викладачі кафедр
  - Студенти спеціальностей
- Пошук:
  - Викладача/студента за прізвищем чи іменем, або за їх частиною
  - Викладача за кафедрою
  - Студента за курсом
  - Студента за групою
- Виведення:
  - Студентів/викладачів факультету у алфавітному порядку
  - Студентів університету згрупованими за курсом
  - Студентів кафедр згрупованими за курсом
  - Студентів вказаного курсу у алфавітному порядку
  - Студкнтів/викладачів кафедр у алфавітному порядку

Структура програми

Програми складається з двох пакетів: Lab та Utility, та з класу Main у кореневому пакеті

Пакет Lab



Абстрактний клас Named, що є батьком всіх інших містить поле name та parent а також реалізує всі методи, що є спільними між іншими класами

Клас University є репрезентацією університету, містить в собі масив факультетів та методи для керування структурою університету / пошуку / виведення данних

Клас Faculty є репрезентацією факультету, містить масив кафедр, та методи для керуванням структурою факультету / виведення данних

Клас Cathedra є репрезентацією кафедри, містить масив спеціальностей, масив викладачів, та методи для керуванням структурою кафедри / виведення данних

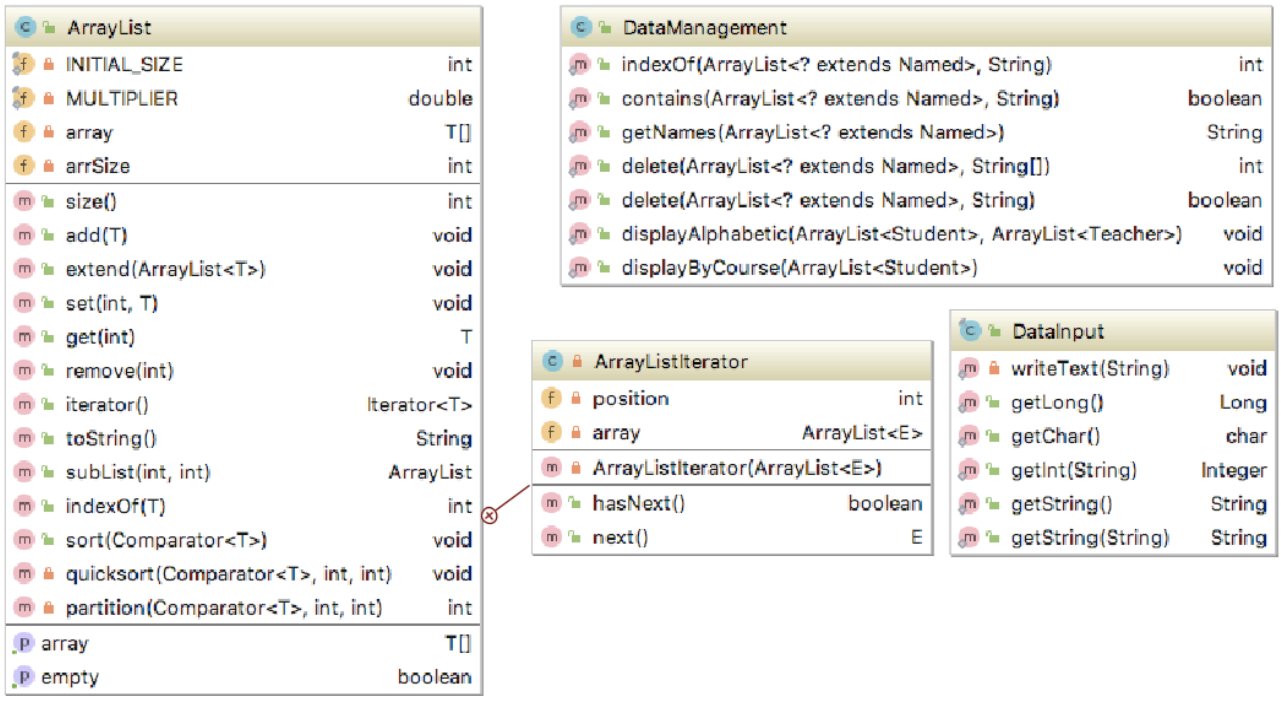
Клас Faculty є репрезентацією факультету, містить масив кафедр, та методи для керуванням структурою факультету / виведення данних

Клас Speciality є репрезентацією спеціальності, містить масив студентів, та методи для керуванням структурою спеціальності

Клас Student є репрезентацією студенту, містить поля курсу та групи

Клас Teacher є репрезентацією викладача

Пакет Utility



Клас ArrayList є репрезентацією структури данних динамічного масиву (списку). Містить методи для взаємодіє з ним, включаючи сортування масиву за допомоги алгоритму Quicksort та за фактором визначеним у компараторі, за допомоги якого і здійснюється порівняння.

Клас ArrayListIterator є вкладеним в клас ArrayList і є реалізацією об’єкта ітерації

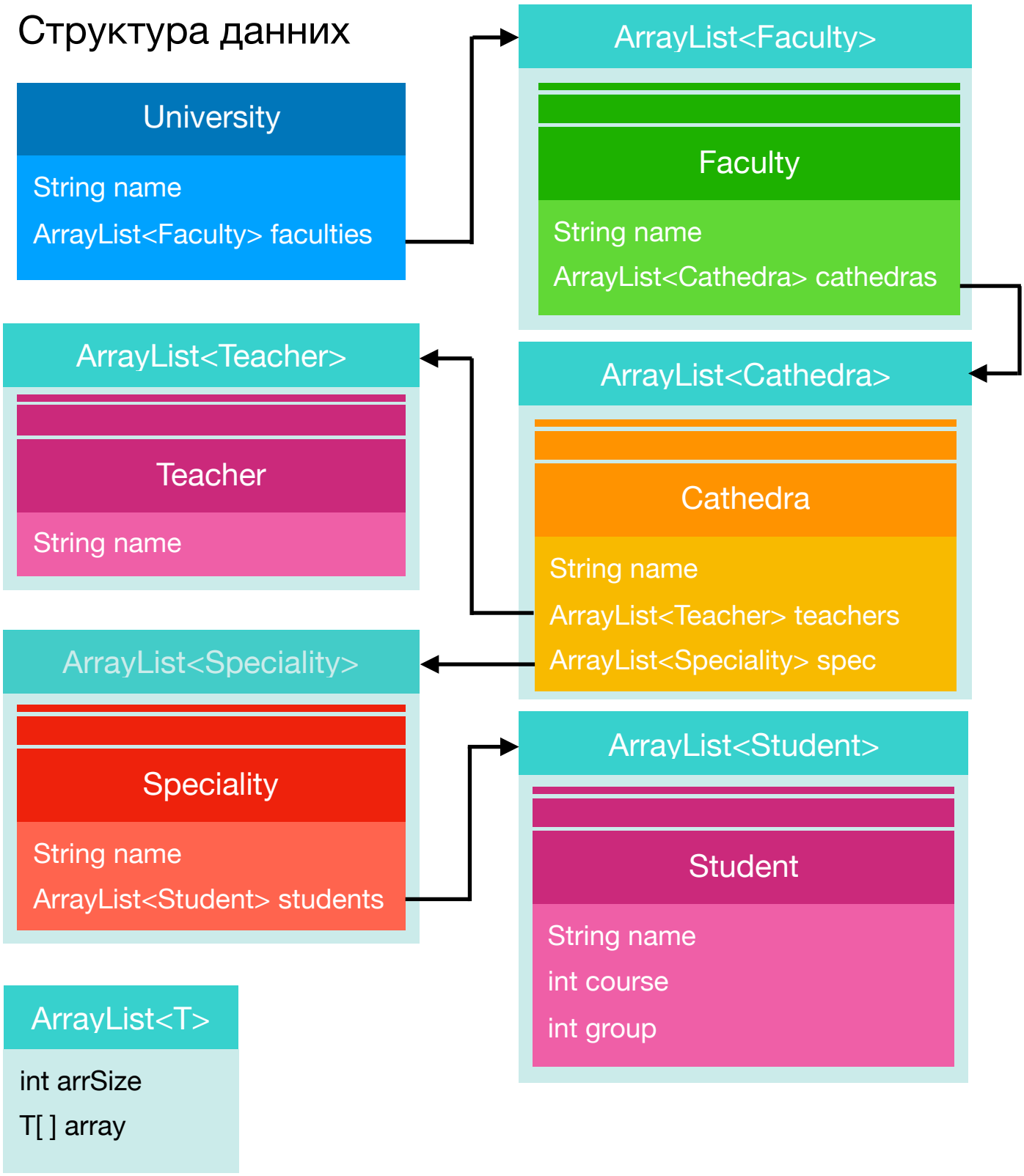
Клас DataManagement є допоміжним класом з методами упорядження данних специфічних до цього проекту

Клас DataInput є допоміжним класом, що містить методи для отримання данних з консолі

Пакет default

Клас Main, що є вхідною точкою програми. Реалізує метод main, створює об’єкт університету, наповнює його початковими данними та передає керування консолю

Структура данних



Вирішення основних задач

Динамічне збереження данних

Використовуючи структуру данних ArrayList можливо зберігати динамічну кількість об’єктів у пам’яті, маючи змогу постійно додавати до кінця нових членів списку виділяючи масив більшого розміру.

Відносини між об’єктами

Встановлено об’єкти мають можливість перейняти керування консолью у батьківського об’єкту (див. Структура данних) задля здійснення змін над собою

Взаємодія з користувачем

Інтерфейс має декілька шарів, кожен з яких прив’язаний до певного об’єкту (див. Структура данних). Кожен шар має команд, включаючи перехід на рівень вище, чи нижче. Щоб не загубитися серед павутини об’єктів користувач бачить якому рівню він дасть команду за допомоги покажчика шляху вигляду University/Faculty/Cathedra/>

Пошук

На загальному списку студентів/викладачів здійснюється пошук серед полів імені у студентів та імені, кафедри у викладачів. При вводі цифри також здійснюється пошук серед полів курсу та групи у студентів

Сортування

Використано алгоритм Quicksort у класу ArrayList. Задля уніфікації сортування між усіма можливими типами об’єктів в середині списку, використовуються класи що імплементують інтерфейс Comparator. Таким чином можливо порівняти два об’єкти між собою. Також це дає змогу відсортувати один і той же об’єкт за різними характеристиками (як наприклад сортування студентів за іменем чи за курсом) реалізуючи лише ще один Comparator

Проблеми в роботі

Підтримання найновішої версії коду серед учасників

Використання пакету утиліт git разом із сервісом gitHub дало змогу оперативно працювати одночасно. Потребувало вивченню утиліти git.

Підтримання чистоти коду

Кожен новий комміт був перевірений з боку іншого учасника. Таким чином виявлено 10-ки багів, і код (хоч і був часто переписаний після завантаження) зберігав свою читоту. Також декомпіляція та використання більш узагальнених функцій допомагає у можливості коду бути усвідомленим іншою людиною

Інтуїтивність користувацького інтерфейсу

Модель користування "Питання-відповідь" обмежена, адже користувач іде "по рельсах" програми. Реалізовано шарову систему на прикладі інтерфейсу bash

Заборона використання вбудованих структур данних

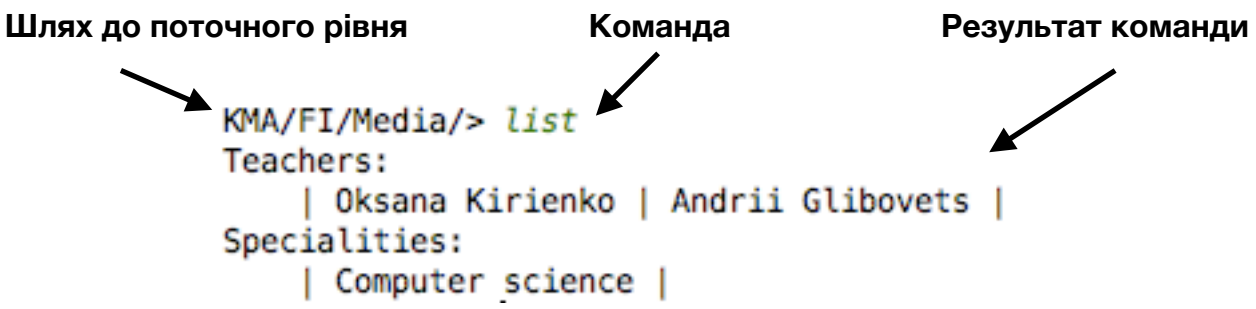
Реалізована копія ArrayList, що не поступається за функціоналом вбудованій версії

Інструкція користувача

Після запуску ви побачите інтерфейс накшталт такого:

```
Available commands:
Name    - change the name of university
Add     - add new faculties
Edit    - edit existing faculties
Delete  - delete any existing faculty
List    - list all faculties
Find    - find teachers or students throughout the university
Display - display all students grouped by course
Help    - show this message again
Stop    - stop execution of program
KMA/>
```

Ви можете надрукувати команду help щоб побачити які команди доступні на цьому рівні. По замовчуванню ви опинитесь на рівні університету, щоб перйти на рівень факультету чи глибше, введіть команду edit. Щоб піднятись на рівень вище введіть команду stop.
При запуску програми в середині системи вже будуть данні такі як факультети FI, FES, FS, кафедри на них та студенти й викладачі.



На рівні університету можливо здійснити пошук за іменем, групою та курсом серед студентів, чи за іменем та кафедрою серед викладачів за допомогою команди find.
На рівнях університету, факультету та кафедри можливо здійснити вивід студентів/викладачів у алфавітному порядку чи групуючи за курсом за допомогою варіацій команд display.
В свою чергу команди add, delete, edit додають, видаляють, редагують членів того чи іншого об’єкту (наприклад на рівні університету можливо редагувати список факультетів, а на рівні спеціальності — студентів).
Командою list можливо вивести всіх членів рівня.
Команда Name змінює ім’я поточного рівня.

Висновок

За допомоги цієї лабораторної роботи ми змогли посилити навички співпраці, та дізнались про можливі засоби, щодо її покращення (git). Закріпили знання з алгоритміки (пошук та сортування об’єктів), композиції (будівництво архітектури програми) та структцур данних (ArrayList).
В результаті ми отримали багаторівневу систему здатню зберігати, модифікувати, та виводити данні про університет.

Початковий код

Може бути знайдений у вільному доступі за посиланням:
https://github.com/Malien/java-p2-l1