

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Об'єктно-орієнтоване програмування ч.2»

Звіт з лабораторної роботи №10
Тема: «Обробка параметризованих контейнерів»

Виконав:
ст. гр. КИТ-118в
Кліщов Б. Р.

Перевірив:
Пугачев Р.В.

Харків – 2020

Мета: Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

Кліщов Б. Р., КИТ-118в

1.2 Загальне завдання

Використовуючи програму рішення завдання лабораторної роботи №9:

- Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.
- Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
- Автоматичний режим виконання програми задається параметром командного рядка **-auto**. Наприклад, `java ClassName -auto`.
- В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.
- Забороняється використання алгоритмів з Java Collections Framework.

1.3 Задача

Бюро знайомств: Сортуння за датою реєстрації, за кількістю властивостей в розділі "відомості про себе", за кількістю властивостей в розділі "вимоги до партнера".

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Були розроблені додаткові функції для класа-контейнера. Розроблене діалогове меню та можливість зчитування даних з файлу.

2.2 Ієрархія та структура класів

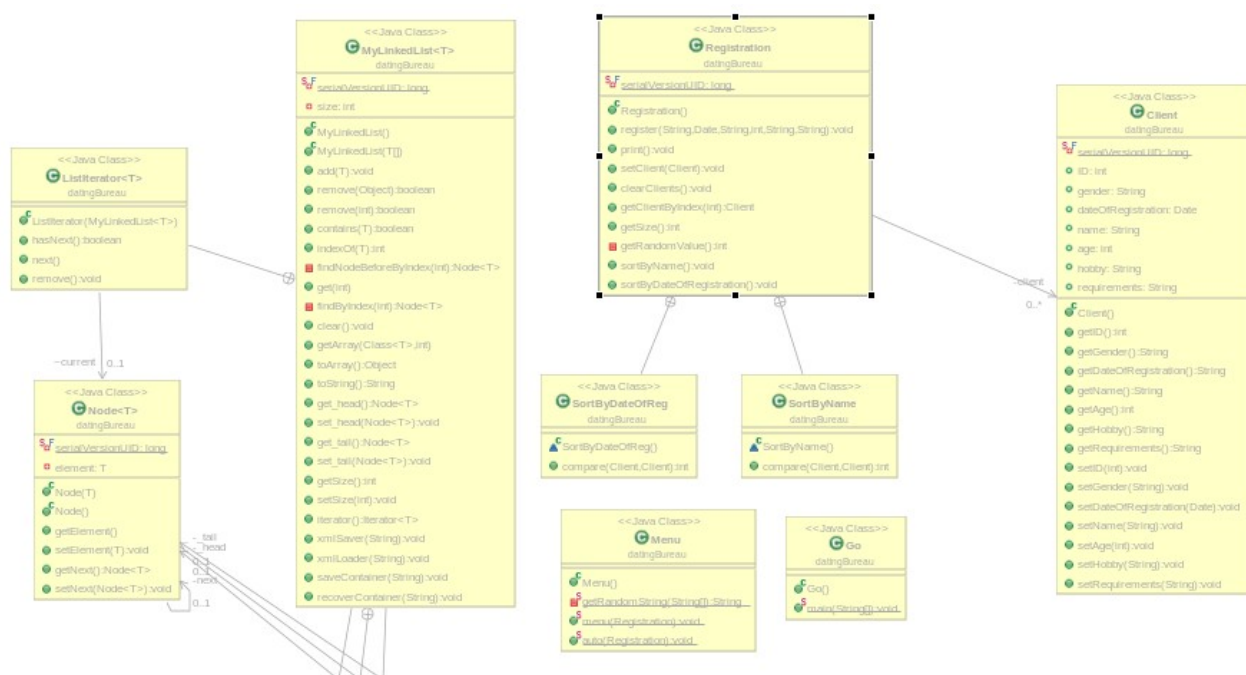


Рисунок 1. Діаграма класів

2.3 Важливі фрагменти програми

```
public static void auto(Registration registration) {
    Random random = new Random ();
    MyLinkedList <Client> myList = new MyLinkedList <Client> ();
    String[] gender = {"male", "female"};
    String[] name = {"John", "Victor", "James"};
    String[] hobby = {"fishing", "football", "chess"};
    String[] requirements = {"honestly", "kindness", "sense of humor"};

    for(int i = 0; i < 3; i++) {
        registration.register(
            getRandomString(gender),
            new Date(),
            getRandomString(name),
            random.nextInt(29) + 16,
            getRandomString(hobby),
            getRandomString(requirements));
    }
    registration.print();
    System.out.println("Adding to container...");
    for(int i = 0; i < registration.getSize(); i++) {
        myList.add(registration.getClientByIndex(i));
    }
    if(myList.getSize() != 0) {
        System.out.println("Complete!!!");
    } else { System.out.println("Error"); }
}
```

Рисунок 2. Авто-мод

```
int choose;
Scanner in;
do {
    System.out.println("1 - REGISTER NEW CLIENT");
    System.out.println("2 - OUTPUT CLIENTS INFO");
    System.out.println("3 - GENERIC CLASS");
    System.out.println("4 - SERIALIZE OBJECT");
    System.out.println("5 - DESERIALIZE OBJECT");
    System.out.println("6 - SORT BY NAME");
    System.out.println("7 - SORT BY DATE OF REGISTRATION");
    System.out.println("8 - EXIT PROGRAM\n");

    while(true) {
        System.out.print("Your choice: ");
        in = new Scanner(System.in);
        choose = in.nextInt();
        if(choose < 1 || choose > 8) {
            System.out.println("Invalid number of operation.");
            continue;
        }
        break;
    }
}
```

Рисунок 3. Меню

```

public void sortByName() {
    Collections.sort(this.client, new SortByName());
}
public void sortByDateOfRegistration() {
    Collections.sort(this.client, new SortByDateOfReg());
}

class SortByName implements Comparator<Client> {
    public int compare(Client a, Client b) {
        return a.getName().compareTo(b.getName());
    }
}

class SortByDateOfReg implements Comparator<Client> {
    public int compare(Client a, Client b) {
        return a.getDateOfRegistration().compareTo(b.getDateOfRegistration());
    }
}

```

Рисунок 4. Методи сортування

2.4 Опис алгоритмів

Програма містить два моди: автоматичний та за вибором користувача. Так як в лабораторній роботі №7 використовувався ArrayList, то параметризований клас контейнер було вирішено використовувати як контейнер для зберігання результату по вибору користувача. Замість методу сортування за кількістю хоббі та за кількістю потреб до свіврозмовника було розроблено метод сортування за ім'ям, так як в полях хобі та потреби передбачалось довільний ввід тексту. В автомоді демонструється додавання рандомних користувачів з імпровізованої "бази даних". Після цього запускається процес запису даних до параметризованого контейнера, створеного в попередній лабораторній роботі, та читання з нього, а вже потім сортування. Для запуску автоматичного режиму потрібно запустити програму з консолі з аргументом "-auto".

3 ВАРІАНТИ ВИКОРИСТАННЯ

```
ID: 27835718
Gender: male
Date of registration: Sun May 10 22:01:58 EEST 2020
Name: Bohdan
Age: 19
Hobby: cooking
Requirements: honest

ID: 27376262
Gender: female
Date of registration: Sun May 10 22:02:43 EEST 2020
Name: Kate
Age: 29
Hobby: football
Requirements: polite

ID: 27779566
Gender: male
Date of registration: Sun May 10 22:03:32 EEST 2020
Name: Albert
Age: 12
Hobby: Minecraft
Requirements: playing-skills
```

Рисунок 5. Додавання та відображення елементів

```
ID: 27779566
Gender: male
Date of registration: Sun May 10 22:03:32 EEST 2020
Name: Albert
Age: 12
Hobby: Minecraft
Requirements: playing-skills

ID: 27835718
Gender: male
Date of registration: Sun May 10 22:01:58 EEST 2020
Name: Bohdan
Age: 19
Hobby: cooking
Requirements: honest

ID: 27376262
Gender: female
Date of registration: Sun May 10 22:02:43 EEST 2020
Name: Kate
Age: 29
Hobby: football
Requirements: polite
```

Рисунок 6. Сорткування

```

public class Go {
    public static void main(String[] args) throws FileNotFoundException, ClassNotFoundException, IOException {
        Registration registration = new Registration();
        for(String str : args) {
            if (str.equals("-auto")) {
                Menu.auto(registration);
                return;
            }
        }
        Menu.menu(registration);
    }
}

```

Рисунок 7. Вхід в автоматичний режим

```

Gender: female
Date of registration: Sun May 10 22:05:51 EEST 2020
Name: Victor
Age: 30
Hobby: fishing
Requirements: honestly

ID: 48826743
Gender: male
Date of registration: Sun May 10 22:05:51 EEST 2020
Name: John
Age: 37
Hobby: chess
Requirements: kindness

ID: 64173581
Gender: male
Date of registration: Sun May 10 22:05:51 EEST 2020
Name: John
Age: 16
Hobby: football
Requirements: kindness

Adding to container...
Complete!!!
Reading from container...
Complete!!!

```

Рисунок 8. Результат запису та читання з контейнеру

ВИСНОВКИ

Я розширив функціональності параметризованих класів.