

## Лабораторна робота №7

### Об'єктно-орієнтована декомпозиція

**Мета:** Використання об'єктно-орієнтованого підходу для розробки об'єкта предметної (прикладної) галузі.

#### 1 ВИМОГИ

1. Використовуючи об'єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно прикладної задачі - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

##### 1.1 Розробник

- П.І.Б: Заночкин. Є. Д.
- Група: КІТ-119а
- Варіант: 7

##### 1.2 Завдання

###### Бюро знайомств

Запис про клієнта: стать; реєстраційний номер; дата реєстрації; відомості про себе (довільний набір властивостей: ім'я, зріст, колір очей, дата народження, хобі тощо); вимоги до партнера (довільний набір властивостей).

#### 2 ОПИС ПРОГРАМИ

##### 2.1 Було використано наступні засоби:

public int getId() – геттер;  
public void setId(int id) – сеттер.

##### 2.2 Ієрархія та структура класів

Було створено 5 класів:

- public class Main – містить метод main;
- public class Client – клас, що містить поля клієнта;
- public class ClientList – містить масив типу Client та метод виведення масиву на екран;
- public class InfoAboutYourself – містить інформацію, що відноситься до поля «відомості про себе»;
- public class PartnerRequirements – містить інформацію, що відноситься до поля «вимоги до партнера».

## 2.3 Важливі фрагменти програми

### Клас Client

```
public class Client
{
    /*
     * Змінні
     */
    private String gender;
    private int id;
    private String registrationDate;
    private InfoAboutYourself information;
    private PartnerRequirements requirements;
    /*
     * Конструктор класу
     */
    public Client(String gender, int id, String date, InfoAboutYourself info, PartnerRequirements
requirements)
    {
        this.gender = gender;
        this.id = id;
        this.registrationDate = date;
        this.information = info;
        this.requirements = requirements;
    }
    /*
     * Геттери та сеттери
     */
    public String getClientGender()
    {
        return gender;
    }
    public void setClientGender(String gender)
    {
        this.gender = gender;
    }
    public int getId()
    {
        return id;
    }
    public void setId(int id)
    {
        this.id = id;
    }
    public String getDate()
    {
        return registrationDate;
    }
    public void setDate(String date)
    {
        this.registrationDate = date;
    }
    public InfoAboutYourself getInformation()
    {
        return information;
    }
    public void setInformation(InfoAboutYourself info)
    {
        this.information = info;
    }
    public PartnerRequirements getRequirements()
    {
        return requirements;
    }
}
```

```

    }
    public void setRequirements(PartnerRequirements requirements)
    {
        this.requirements = requirements;
    }
}

```

## Клас InfoAboutYourself

```

public class InfoAboutYourself
{
    /*
     * Змінні
     */
    private String name;
    private int age;
    private int height;
    private String eyeColour;
    private String hobby;
    /*
     * Конструктор класу
     */
    public InfoAboutYourself(String name, int age, int height, String eyeColour, String hobby)
    {
        this.name = name;
        this.age = age;
        this.height = height;
        this.eyeColour = eyeColour;
        this.hobby = hobby;
    }
    /*
     * Геттери та сеттери
     */
    public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name = name;
    }
    public int getAge()
    {
        return age;
    }
    public void setAge(int age)
    {
        this.age = age;
    }
    public int getHeight()
    {
        return height;
    }
    public void setHeight(int height)
    {
        this.height = height;
    }
    public String getEyeColour()
    {
        return eyeColour;
    }
    public void setEyeColour(String eyeColour)
    {
        this.eyeColour = eyeColour;
    }
}

```

```

    }
    public String getClientHobby()
    {
        return hobby;
    }
    public void setClientHobby(String hobby)
    {
        this.hobby = hobby;
    }
}

```

## Клас PartnerRequirements

```

public class PartnerRequirements
{
    /*
     * Змінні
     */
    private String gender;
    private int minAge;
    private int maxAge;
    /*
     * Конструктор класу
     */
    public PartnerRequirements(String gender, int minAge, int maxAge)
    {
        this.gender = gender;
        this.minAge = minAge;
        this.maxAge = maxAge;
    }
    /*
     * Геттери та сеттери
     */
    public String getPartnerGender()
    {
        return gender;
    }
    public void setPartnerGender(String gender)
    {
        this.gender = gender;
    }
    public int getMinAge()
    {
        return minAge;
    }
    public void setMinAge(int minAge)
    {
        this.minAge = minAge;
    }
    public int getMaxAge()
    {
        return maxAge;
    }
    public void setMaxAge(int maxAge)
    {
        this.maxAge = maxAge;
    }
}

```

## Клас ClientList

```

public class ClientList
{
    Client array[] = new Client[2];
    /*
     * Метод print, що виводить інформацію про клієнта на екран

```

```

*/
public void print()
{
    for(int i = 0; i < array.length; i++)
    {
        System.out.println("ID - " + array[i].getId() + "\nДата регистрации - " +
array[i].getDate() + "\nПол - " + array[i].getClientGender() + "\n");
        System.out.println("Информация о себе:\nИмя - " +
array[i].getInformation().getName() + "\nВозраст - " + array[i].getInformation().getAge() +
"\nРост - " + array[i].getInformation().getHeight() + "\nЦвет
глаз - " + array[i].getInformation().getEyeColour() +
"\nХобби - " + array[i].getInformation().getClientHobby() +
"\n");
        System.out.println("Требования к партнёру:\nПол - " +
array[i].getRequirements().getPartnerGender() +
"\nМинимальный возраст - " +
array[i].getRequirements().getMinAge() + "\nМаксимальный возраст - " +
array[i].getRequirements().getMaxAge());
        System.out.println("-----");
    }
}
}

```

### 3 Результаты работы программы

|  |   |
|--|---|
| <pre> ID - 1 Дата регистрации - 01.05.2018 Пол - Мужской  Информация о себе: Имя - Егор Возраст - 18 Рост - 185 Цвет глаз - Синий Хобби - Видео игры  Требования к партнёру: Пол - Женский Минимальный возраст - 18 Максимальный возраст - 25 ----- ID - 2 Дата регистрации - 12.12.2020 Пол - Женский  Информация о себе: Имя - Катя Возраст - 17 Рост - 170 Цвет глаз - Зелёный Хобби - Вышивка, пение  Требования к партнёру: Пол - Мужской Минимальный возраст - 18 Максимальный возраст - 25 ----- </pre> | <pre> ID - 1 Дата регистрации - 01.05.2018 Пол - Мужской  Информация о себе: Имя - Егор Возраст - 99 Рост - 185 Цвет глаз - Синий Хобби - Видео игры  Требования к партнёру: Пол - Женский Минимальный возраст - 18 Максимальный возраст - 25 ----- ID - 2 Дата регистрации - 12.12.2020 Пол - Женский  Информация о себе: Имя - Оля Возраст - 17 Рост - 170 Цвет глаз - Красный Хобби - Вышивка, пение  Требования к партнёру: Пол - Мужской Минимальный возраст - 18 Максимальный возраст - 30 ----- </pre> |
| а)   | б)  |

Рисунок 7.1 – Результат работы программы у середовищі Eclipse

### Висновок

Під час виконання лабораторної роботи було набуто навичок роботи з domain-об'єктами та об'єктно-орієнтованою декомпозицією у середовищі Eclipse IDE.