

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

- Дем'яненко Дмитро Андрійович
- Група: КІТ-119д
- 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

```
package ua.khpi.oop.demianenko07;

public class Client
{
    private String gender;
    private int id;
    private String registrationDate;
    private InfoAboutYourself information;
    private PartnerRequirements requirements;

    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;
    }

    public Client()
    {
    }

    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 ClientList

```

package ua.khpi.oop.demianenko07;

public class ClientList
{
    Client array[] = new Client[2];

    public void print()
    {
        for(int i = 0; i < array.length; i++)
        {
            System.out.println("ID - " + array[i].getId() + "\nRegistration
Date - " + array[i].getDate() + "\nGender - " + array[i].getClientGender() + "\n");
            System.out.println("Personal information:\nName - " +
array[i].getInformation().getName() + "\nAge - " + array[i].getInformation().getAge()
+
"\nHeight - " +
array[i].getInformation().getHeight() + "\nEye color - " +
array[i].getInformation().getEyeColour() +
"\nHobby - " +
array[i].getInformation().getClientHobby() + "\n");
            System.out.println("Partner requirements:\nGender - " +
array[i].getRequirements().getPartnerGender() +
"\nMinimum age - " +
array[i].getRequirements().getMinAge() + "\nMaximum age - " +
array[i].getRequirements().getMaxAge());
            System.out.println("-----");
        }
    }
}

```

Клас PartnerRequirements

```

package ua.khpi.oop.demianenko07;

public class PartnerRequirements
{
    String gender;
    private int minAge;
    private int maxAge;

    public PartnerRequirements(String gender, int minAge, int maxAge)
    {

```

```
        this.gender = gender;
        this.minAge = minAge;
        this.maxAge = maxAge;
    }

    public PartnerRequirements()
    {

    }

    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;
    }
}
```

3 Результати роботи програми

```
ID - 1
Registration Date - 21.04.2016
Gender - Man

Personal information:
Name - Dmitriy
Age - 19
Height - 182
Eye color - Brown
Hobby - Football

Partner requirements:
Gender - Woman
Minimum age - 18
Maximum age - 25
-----
ID - 2
Registration Date - 17.09.2018
Gender - Woman

Personal information:
Name - Alina
Age - 19
Height - 167
Eye color - Brown
Hobby - Drawing

Partner requirements:
Gender - Man
Minimum age - 18
Maximum age - 25
-----
```

Рисунок 7.1 – Результат роботи програми 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.