

Об'єктно-орієнтована декомпозиція

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

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

Особисте завдання

Адресна книга

Запис: П.І.Б .; дата народження; телефони (кількість не обмежена); адреса; дата і час редагування.

Розробник: Білий Вадим Іванович КІТ-119а №3.

Опис програми

Засоби ООП: клас, метод класу.

Структура класів: Публічний клас Main, клас даних Person, утилітарний клас AddressBook.

Важливі фрагменти програми:

```
public class Person {
    private String name;
    private ArrayList<String> telephone;
    private Calendar dateOfBirthday;
    private String address;
    private Date dateOfLastRedaction;
    public void setName(String name) {
        this.name = name;
        dateOfLastRedaction = new Date();
    }
    public String getName() {
        return name;
    }
    public void setTelephone(ArrayList<String> telephone) {
        this.telephone = telephone;
        dateOfLastRedaction = new Date();
    }
    public ArrayList<String> getTelephone() {
        return telephone;
    }
    public void setDateOfBirthday(Calendar dateOfBirthday) {
        this.dateOfBirthday = dateOfBirthday;
    }
}
```

```

        dateOfLastRedaction = new Date();
    }
    public Calendar getDateOfBirthday() {
        return dateOfBirthday;
    }
    public void setAddress(String address) {
        this.address = address;
        dateOfLastRedaction = new Date();
    }
    public String getAddress() {
        return address;
    }
    public Date getDateOfLastRedaction() {
        return dateOfLastRedaction;
    }
    public void addTelephone(String telephone) {
        this.telephone.add(telephone);
        dateOfLastRedaction = new Date();
    }
    Person(){
        name = null;
        telephone = null;
        dateOfBirthday = null;
        address = null;
        dateOfLastRedaction = new Date();
    }
    Person(String name,ArrayList<String> telephone,Calendar dateOfBirthday,String
address){
        this.setName(name);
        this.addTelephone(telephone);
        this.setDateOfBirthday(dateOfBirthday);
        this.setAddress(address);
    }
    Person(String name,String telephone,Calendar dateOfBirthday,String address){
        this.setName(name);
        ArrayList<String> mass= new ArrayList<>();
        mass.add(telephone);
        this.addTelephone(mass);
        this.setDateOfBirthday(dateOfBirthday);
        this.setAddress(address);
    }
}

public class AddressBook {
    ArrayList<Person> mass = new ArrayList<Person>();
    public void add(Person person) {
        mass.add(person);
    }
    public void print() {
        for(var person : mass) {
            System.out.println("\nName: "+person.getName()+"\nTelephone: "+
person.getTelephone()+"\nDate of birthday: "+person.getDateOfBirthday().
"\nAddress: "+person.getAddress()+"\nDate of last redaction: "+
person.getDateOfLastRedaction().toString());
        }
    }
}

```

Висновки

Оволодів навичками розробки об'єкту за допомогою об'єктно-орієнтованого підходу.