**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](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/" \l "_4) - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

**Прикладна галузь**

Адресна книга

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

**1.1 Розробник**

***Виконавець:*** Сиромятников Марк

***Група:*** КІТ-119а

**Варіант:** 19

**2 ОПИС ПРОГРАМИ**

Було використано наступні засоби:

* *StringBuffer* – конкатенація у циклі
* *Об’єктно орієнтована декомпозиція* – керування масивом domain об’єктів.

**2.1 Ієрархія та структура класів**

Було створено 1 клас MainClas, у якому створюється об’єкт класу AdressBook, який містить масив на 3 елементи класу Person, у якому знаходяться полі з даними, гетери та сетери.

**2.2 Важливі фрагменти програми**

public static void main(String[] args) {

AdressBook ab = new AdressBook();

ab.array[0] = new Person("Марк","11.08.2001",new long[] {1,2,3,4,5,6,7,8},"Morozova.St", "13.03.2020");

ab.array[1] = new Person("Деус","12.12.1965",new long[] {10,20,30,40},"Khreshatic.St", "12.04.2020");

ab.array[2] = new Person("Сыромятников","23.03.2015",new long[] {10000,2111,33},"", "12.02.2020");

ab.printAll();

}

public class Person {

private String fullName;

private String dateOfBirth;

private long telephoneNumbers[];

private String adress;

private String editingDateAndTime;

public long[] getTelephoneNumbers() {

return telephoneNumbers;

}

public void setTelephoneNumbers(long[] telephoneNumbers) {

this.telephoneNumbers = telephoneNumbers;

}

public String showNumbers() {

StringBuilder sb = new StringBuilder();

if (telephoneNumbers.length > 1) {

for (int i = 0; i < telephoneNumbers.length; i++) {

sb.append(telephoneNumbers[i] + " ");

}

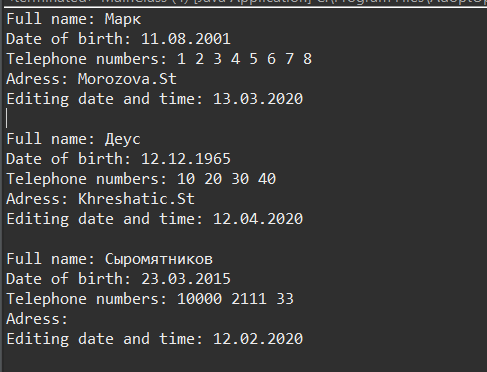
return sb.toString();

}

return sb.append(telephoneNumbers[0]).toString();

}

***Результат роботи програми***



**ВИСНОВКИ**

У результаті виконання лабораторної роботи було створено об’єктно орієнтовану декомпозицію задля керування масиву з domain об’єктів, заповнено клас прикладної галузі.