**Лабораторна робота №7**

[**Об'єктно-орієнтована декомпозиція**](https://oop-khpi.gitlab.io/#tasks)

**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/#_4) - domain-об'єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об'єктів.

**1.1Розробник** Завадський Дмитро Богданович КІТ119а №5.

**1.3 Задача**

Довідник покупця

Торгівельна точка: назва; адреса; телефони (кількість не обмежена); спеціалізація; час роботи (з зазначенням днів тижня).

**2 ОПИС ПРОГРАМИ**

**2.2 Ієрархія та структура класів**

Було створено 3 класи:

* Public class PointOfSale – клас, що містить необхідні поля та методи прикладної області “Торгівельна точка”.
* public class Main – містить лише метод main.
* Public class Container – клас що містить в собі масив об’єктів класу PointOfSale

**Важливі фрагменти програми**

**Клас** PointOfSale

public class PointOfSale {

private String name\_point\_of\_sale;

private ArrayList<Integer> phone\_number ;

private String specialization;

private LinkedHashMap<String, String> work\_time = new LinkedHashMap<String, String>();

public void Addphone\_number(int value) {

this.phone\_number.add(value);

}

public void Addwork\_time(String key, String value) {

this.work\_time.put(key, value);

}

public String toString\_phone\_number() {

String string = "";

for (Integer integer : phone\_number) {

string +="+" + integer.toString();

string+="\n";

}

return string;

}

public String toString\_work\_time() {

String string ="";

Set set = work\_time.entrySet();

for (Object el: set) {

string+=el.toString();

string+="\n";

}

return string;

}

public PointOfSale(String name\_point\_of\_sale, ArrayList<Integer> phone\_number, String specialization,

LinkedHashMap<String, String> work\_time) {

super();

this.name\_point\_of\_sale = name\_point\_of\_sale;

this.phone\_number = phone\_number;

this.specialization = specialization;

this.work\_time = work\_time;

}

public PointOfSale() {

super();

// **TODO** Auto-generated constructor stub

}

public void Show()

{

System.***out***.println("Название торговой точки - "+ name\_point\_of\_sale +"\r\n" +

"Контактный номер телефона \n"+ toString\_phone\_number() + "\r" +

"Специализация - " + specialization +

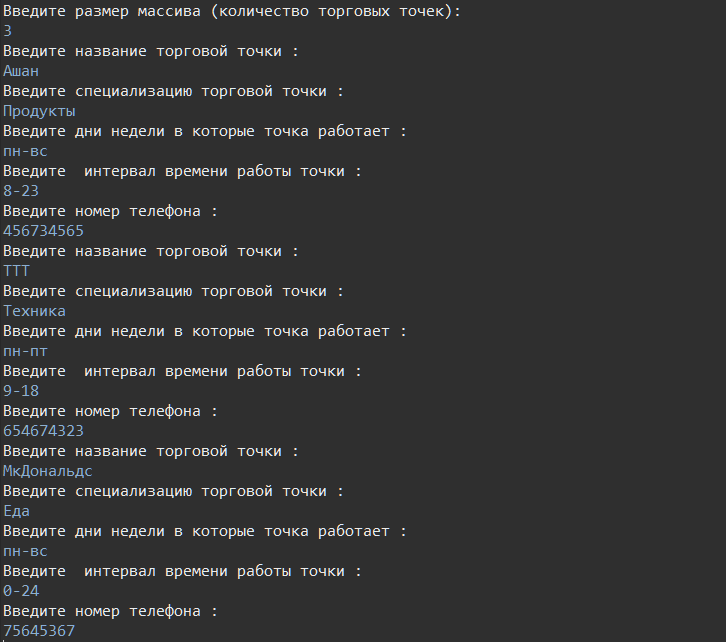
"\nРобочее время : "+ toString\_work\_time() + "\r\n");

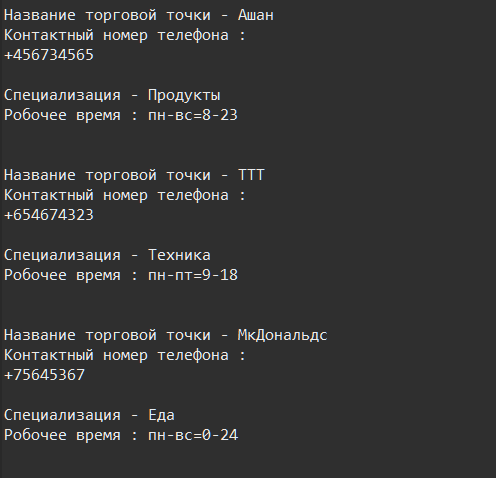
}

}

**3 ВИСНОВКИ**

Результат роботи програми:





**ВИСНОВКИ**

У результаті виконання лабораторної роботи було набуто навичок роботи з domain-об‘єктами; було набуто навичок з об‘єктно-орієнтованою декомпозицією у середовищі JavaEclipse.