**Лабораторна робота №8**

**Основи введення/виведення Java SE**

**Мета:** Оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

**1 ВИМОГИ**

1. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
2. Забороняється використання стандартного протокола серіалізації.
3. Продемонструвати використання моделі Long Term Persistence.
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

**1.1Розробник** Завадський Дмитро Богданович КІТ-119а №5.

**1.3 Задача**

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 String getName\_point\_of\_sale() {

return name\_point\_of\_sale;

}

public void setName\_point\_of\_sale(String name\_point\_of\_sale) {

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

}

public ArrayList<Integer> getPhone\_number() {

return phone\_number;

}

public void setPhone\_number(ArrayList<Integer> phone\_number) {

this.phone\_number = phone\_number;

}

public String getSpecialization() {

return specialization;

}

public void setSpecialization(String specialization) {

this.specialization = specialization;

}

public LinkedHashMap<String, String> getWork\_time() {

return work\_time;

}

public void setWork\_time(LinkedHashMap<String, String> work\_time) {

this.work\_time = work\_time;

}

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" +

"Номера телефонов \n"+ toString\_phone\_number() + "\r" +

"Специализация " + specialization +

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

}

}

**Клас Container**

public class Container {

PointOfSale[] array = new PointOfSale[3];

public void add()

{

int size = array.length;

Scanner scan = new Scanner(System.***in***);

System.***out***.println("Введите название торговой точки");

String name\_point\_of\_sale = scan.nextLine();

System.***out***.println("Введите номер телефону");

int number = scan.nextInt();

System.***out***.println("Введите специализацию торговой точки");

String specialty = scan.nextLine();

System.***out***.println("Введите день недели в который точка работает");

String day = scan.nextLine();

System.***out***.println("Введите в какой интервал времени работает точка");

String time = scan.nextLine();

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

work\_time.put(day,time);

PointOfSale new\_el = new PointOfSale(name\_point\_of\_sale,new ArrayList<Integer>(number),specialty,work\_time);

PointOfSale new\_arr[] = new PointOfSale[++size];

for(int i=0;i<size-1;i++)

{

new\_arr[i]=array[i];

}

new\_arr[size-1] = new\_el;

array = new\_arr;

}

public void remove()

{

int size = array.length;

if(size!=0)

{

PointOfSale new\_arr[] = new PointOfSale[--size];

for(int i=0;i<size;i++)

{

new\_arr[i]=array[i];

}

array = new\_arr;

}

else

{

System.***out***.println("Ваш масив пуст, удалять нечего");

}

}

public void Show()

{

for (PointOfSale carsShop : array) {

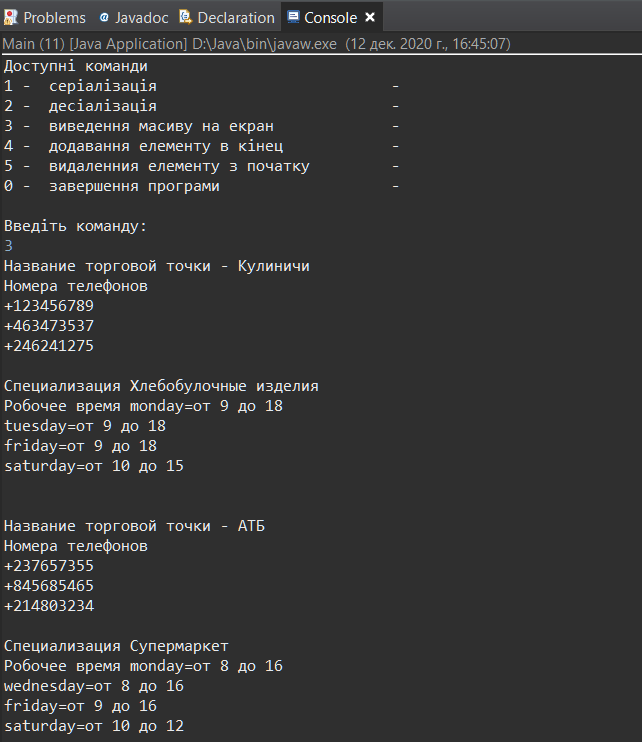
carsShop.Show();

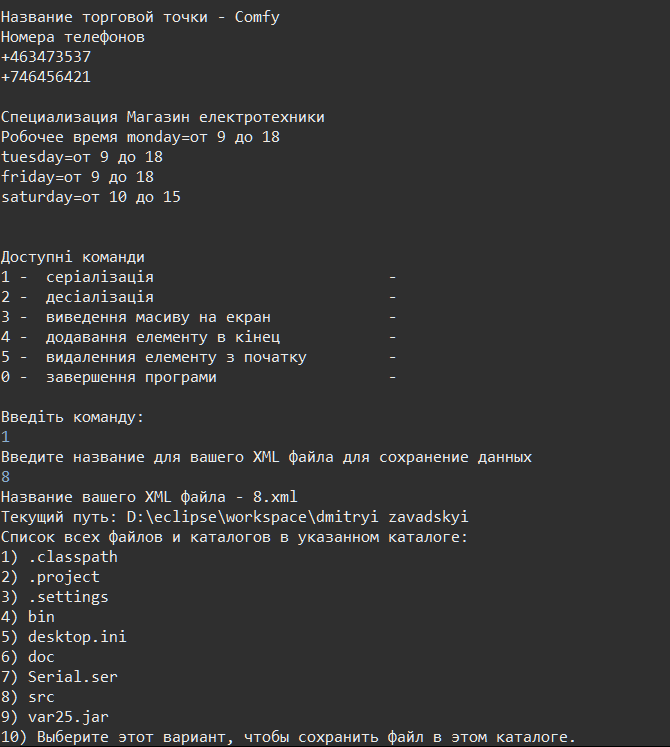
}

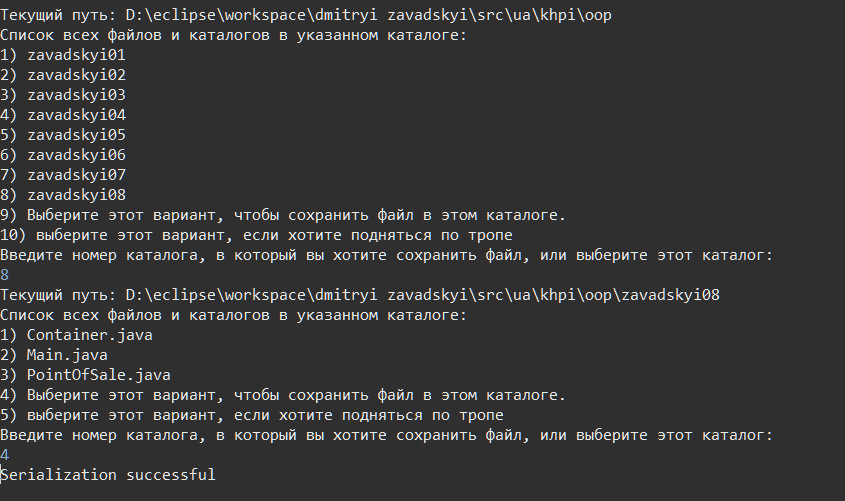
}

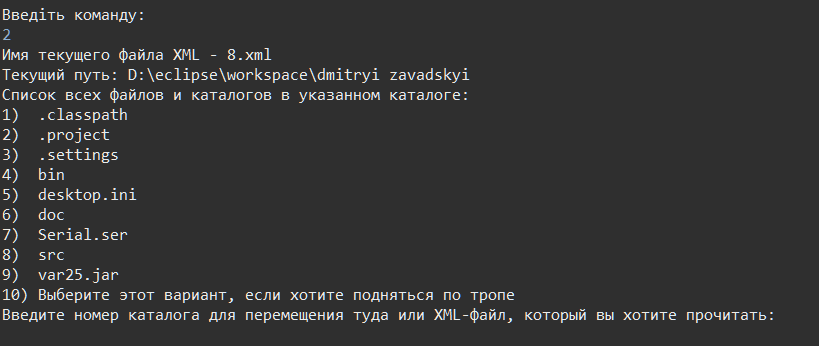
**3 ВИСНОВКИ**

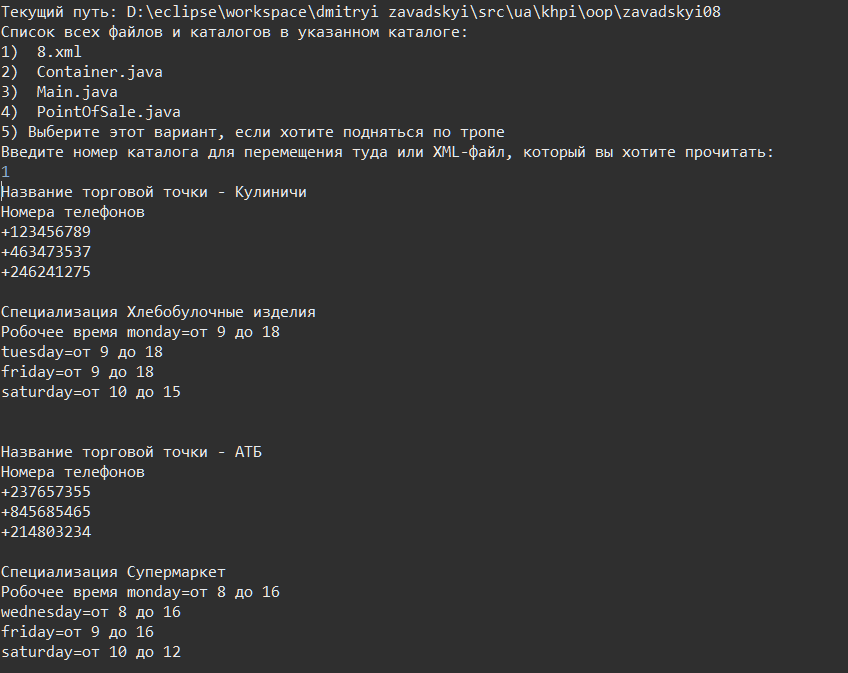
Результат роботи програми:











**ВИСНОВКИ**

У результаті виконання лабораторної роботи було набуто навичок роботи з основами введення/виведення у середовищі Java Eclipse.