**Основи введення/виведення Java SE**

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

**Вимоги**

1. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання [лабораторної роботи №7](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/).
2. Забороняється використання [стандартного протокола серіалізації](https://docs.oracle.com/javase/8/docs/platform/serialization/spec/serialTOC.html).
3. Продемонструвати використання моделі [Long Term Persistence](https://docs.oracle.com/javase/tutorial/javabeans/advanced/longpersistence.html).
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.

**Особисте завдання**

Адресна книга

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

**Розробник**: Рябов Олексій Володимирович КІТ119а №18.

**Опис програми**

**Засоби ООП**: клас, метод класу.

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

**Важливі фрагменти програми:**

**public** **class** RecruitmentAgency {

**private** String company\_name;

**private** String specialty;

**private** String working\_сonditions;

**private** **int** payment;

**private** LinkedHashMap<String, String> requirements\_for\_a\_specialist = **new** LinkedHashMap<String, String>();

**public** **void** AddRequirements\_for\_a\_specialist(String key, String value) {

**this**.requirements\_for\_a\_specialist.put(key, value);

}

**public** String toString\_requirements\_for\_a\_specialist() {

**int** size = requirements\_for\_a\_specialist.size();

String string = requirements\_for\_a\_specialist.entrySet().stream()

.map(e -> e.getKey()+ "- " + e.getValue()+ "\n")

.collect(Collectors.*joining*());

string+= " ";

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

string += requirements\_for\_a\_specialist.entrySet().stream()

.map(e -> e.getKey()+ "- " + e.getValue()+ "\n")

.collect(Collectors.*joining*());

string+= " ";

}

**return** string;

}

**public** RecruitmentAgency(String company\_name, String specialty, String working\_сonditions, **int** payment,

String specialty\_of\_the\_worker,String education, String experience) {

**super**();

**this**.company\_name = company\_name;

**this**.specialty = specialty;

**this**.working\_сonditions = working\_сonditions;

**this**.payment = payment;

**this**.requirements\_for\_a\_specialist.put("Специальность ", specialty\_of\_the\_worker);

**this**.requirements\_for\_a\_specialist.put("Образование ", education);

**this**.requirements\_for\_a\_specialist.put("Стаж ", experience);

}

**public** RecruitmentAgency() {

**super**();

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

}

**public** **void** Show()

{

System.***out***.println("Название компании - "+ company\_name +"\r\n" +

"Специализация компании "+ specialty + "\r\n" +

"Условия работы " + working\_сonditions +

"\nЗарплата "+ payment + "\r\n" +

"Условия к работнику "+ toString\_requirements\_for\_a\_specialist() + "\r\n" );

}

}

**public** **class** Container {

RecruitmentAgency array[] = **new** RecruitmentAgency[3];

**public** **void** add()

{

**int** size = array.length;

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

System.***out***.println("Введите название компании");

String company\_name = scan.nextLine();

System.***out***.println("Введите специализацию данной компании");

String specialty = scan.nextLine();

System.***out***.println("Условия приема на работу");

String working\_сonditions = scan.nextLine();

System.***out***.println("Введите оплату за работу");

**int** payment = scan.nextInt();

System.***out***.println("Специальность работника");

String specialty\_of\_the\_worker = scan.nextLine();

System.***out***.println("Стаж работника");

String education = scan.nextLine();

System.***out***.println("Образование работника");

String experience = scan.nextLine();

RecruitmentAgency new\_el = **new** RecruitmentAgency(company\_name,specialty,working\_сonditions,payment,specialty\_of\_the\_worker,education,experience);

RecruitmentAgency new\_arr[] = **new** RecruitmentAgency[++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)

{

RecruitmentAgency new\_arr[] = **new** RecruitmentAgency[--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()

{

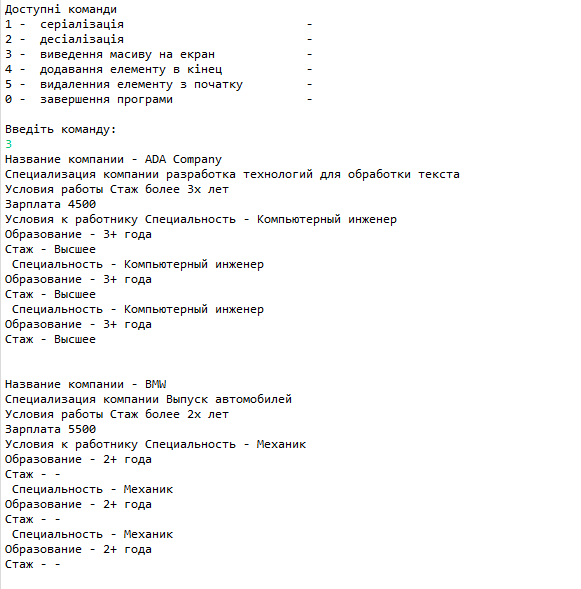
**for** (RecruitmentAgency carsShop : array) {

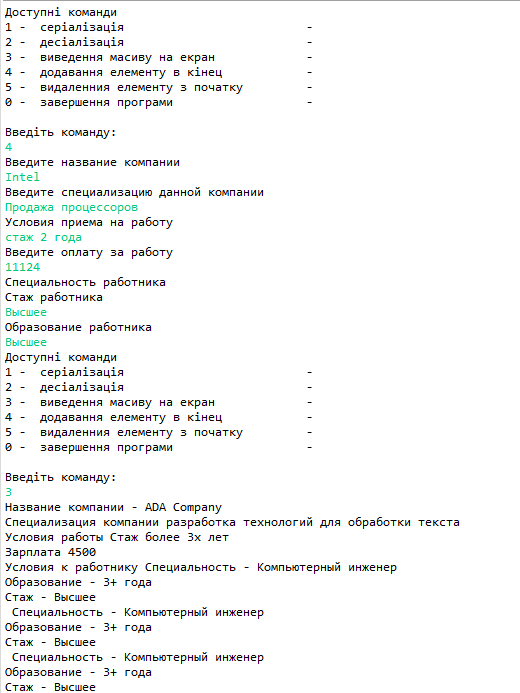
carsShop.Show();

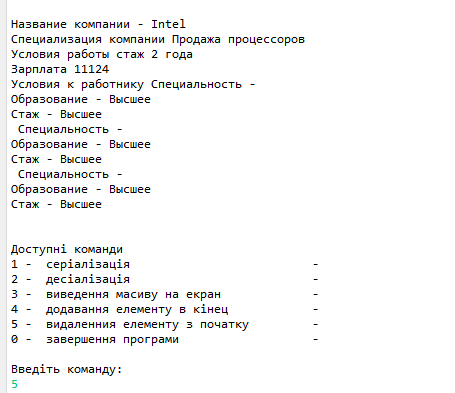
}

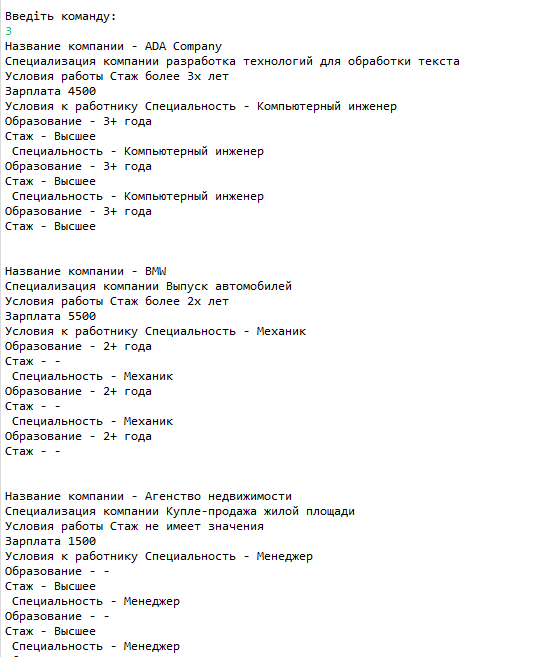
}

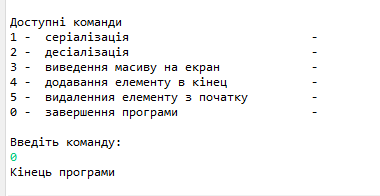
**Результати роботи**

****

****

****

****

****

**Висновки**

Оволодів навичками серіалізації об’єкту за допомогою використання моделі Long Term Persistence.