**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](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

**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.

**Важливі фрагменти програми:**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

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

array[0] = **new** RecruitmentAgency("ADA Company","разработка технологий для обработки текста","Стаж более 3х лет",4500,"Компьютерный инженер","3+ года","Высшее");

array[1] = **new** RecruitmentAgency("BMW","Выпуск автомобилей","Стаж более 2х лет",5500,"Механик","2+ года","-");

array[2] = **new** RecruitmentAgency("Агенство недвижимости","Купле-продажа жилой площади","Стаж не имеет значения",1500,"Менеджер","-","Высшее");

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

car.Show();

}

}

**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" +

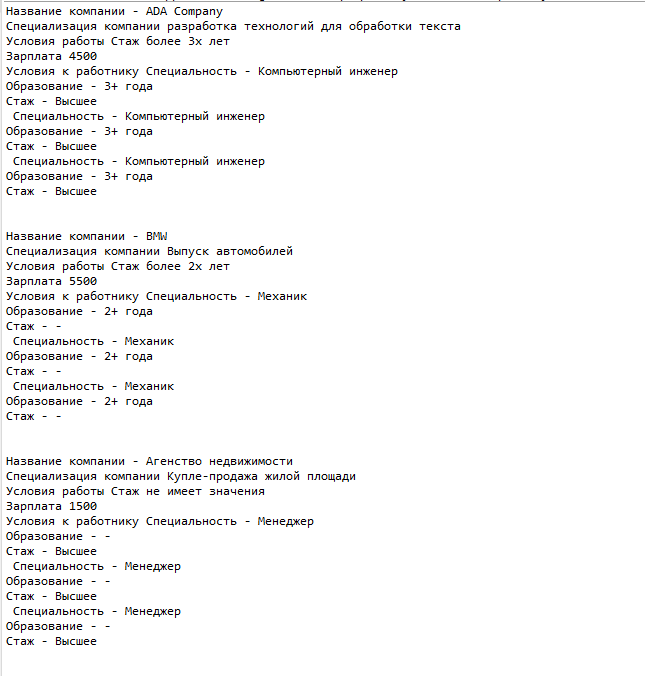
"Условия работы " + working\_сonditions +

"\nЗарплата "+ payment + "\r\n" +

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

}

**Результати роботи**

****

**Висновки**

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