Звіт

Автор: Васильченко С., 1.КІТ101.8б

Дата: 07.11.2019

Лабораторна робота №7

ОБ’ЄКТНО-ОРІЄНТОВАНА ДЕКОМПОЗИЦІЯ

***Мета:***

* 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 - domain-об’єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об’єктів.

ЗАВДАННЯ ДО РОБОТИ

Використовуючи об’єктно-орієнтований аналіз, реалізувати класи для представлення сутностей відповідно прикладної задачі.

**Прикладна задача.** Кадрове агентство. Дані про вакансії: фірма; спеціальність; умови праці; оплата; вимоги до фахівця - набір необов’язкових властивостей у вигляді “спеціальність, стаж, освіта”.

ОПИС ПРОГРАМИ

***2.1 Опис змінних:***

Kontainer kontain = new Kontainer(); // об’єкт контейнера

Recruitment rec1 = new Recruitment(); // об’єкт класа кадрового агенства

Recruitment rec2 = new Recruitment(); // об’єкт класа кадрового агенства

***2.2 Ієрархія та структура класів.***

final class Lab07 – головний клас. Містить метод main(точку входу у програму).

class Functions - допоміжний клас в якому реалізуються функції

class Recruitment - клас прикладної задачі кадрового агенства

**ТЕКСТ ПРОГРАМИ**

File Lab07.java:

package sergey.Lab07;

public class Lab07 {

public static void main(String[] args) {

Kontainer contains = new Kontainer();

Recruitment rec1 = new Recruitment();

Recruitment rec2 = new Recruitment();

rec1.generateVacancy();

contains.add(rec1);

contains.add(rec1);

contains.add(rec1);

rec2.generateVacancy();

contains.add(rec2);

contains.show();

}

}

Kontainer.java:

package sergey.Lab07;

import java.util.Arrays;

class Kontainer {

/\*\*

\* First size for array.

\*/

private final int size = 10;

/\*\*

\* Array contains all data.

\*/

private int count = 0;

private Recruitment[] massik = new Recruitment[size];

void add(final Recruitment temp){

if (count == massik.length) {

massik = Arrays.copyOf(massik, massik.length \* 2);

massik[count++] = temp;

} else {

massik[count++] = temp;

}

}

void show() {

if (count == 0) {

System.out.println("Ошибка! Пустой массив!");

} else {

System.out.println();

for (int i = 0; i < count; i++) {

System.out.println("Вакансия #" + (i+1));

System.out.println("Фирма: " + massik[i].getFirm());

System.out.println("Специальность: " + massik[i].getSpecialty());

System.out.println("Условия работы: " + massik[i].getWorkingConditions());

System.out.println("Оплата: " + massik[i].getPayment());

if (massik[i].getConfirms()) {

System.out.println("Необходимая специальность: " + massik[i].getNeedsSpeciality());

System.out.println("Опыт: " + massik[i].getExperience());

System.out.println("Образование: " + massik[i].getEducation());

}

}

}

}

void clear() {

for (int i = 0; i < count; i++) {

massik[i] = null;

}

count = 0;

}

public void remove(int index) {

for (int i = index; i < count; i++) {

massik[index++] = massik[i + 1];

}

count--;

}

}

Recruitment.java :

package sergey.Lab07;

import java.util.Scanner;

class Recruitment {

private String firm;

private String specialty;

private String workingConditions;

private int payment;

private String needsSpeciality;

private int experience;

private String education;

private boolean confirm = false;

public void setFirm(String firm) {

this.firm = firm;

}

public void setSpecialty(String specialty) {

this.specialty = specialty;

}

public void setWorkingConditions(String workingConditions) {

this.workingConditions = workingConditions;

}

public void setPayment(int payment) {

this.payment = payment;

}

int getPayment() {

return payment;

}

String getWorkingConditions() {

return workingConditions;

}

String getSpecialty() {

return specialty;

}

String getFirm() {

return firm;

}

void generateVacancy() {

Scanner scan = new Scanner(System.in);

Scanner scan2 = new Scanner(System.in);

int choose = 0;

System.out.print("\nВведите фирму: ");

firm = scan.nextLine();

System.out.print("\nВведите специальность: ");

specialty = scan.nextLine();

System.out.print("\nВведите условия работы: ");

workingConditions = scan.nextLine();

System.out.print("\nВведите оплату: ");

payment = scan.nextInt();

System.out.println("Желаете добавить дополнительные условия работы? 1 - Да. 0 - Нет");

choose = scan.nextInt();

while (true){

if (choose == 1) {

System.out.print("\nВведите необходимую специальность: ");

needsSpeciality = scan2.nextLine();

System.out.print("\nНеобходимое образование: ");

education = scan2.nextLine();

System.out.print("\nНеобходимый опыт работы: ");

experience = scan2.nextInt();

confirm = true;

break;

} else if (choose == 0) {

needsSpeciality = null;

experience = 0;

education = null;

break;

} else {

System.out.println("Ошибка! Повторите ввод: ");

}

}

}

public void setExperience(int experience) {

this.experience = experience;

}

public int getExperience() {

return experience;

}

public void setNeedsSpeciality(String needsSpeciality) {

this.needsSpeciality = needsSpeciality;

}

public String getNeedsSpeciality() {

return needsSpeciality;

}

public void setEducation(String education) {

this.education = education;

}

public String getEducation() {

return education;

}

public boolean getConfirms() { return confirm; }}

**ВАРІАНТИ ВИКОРИСТАННЯ**

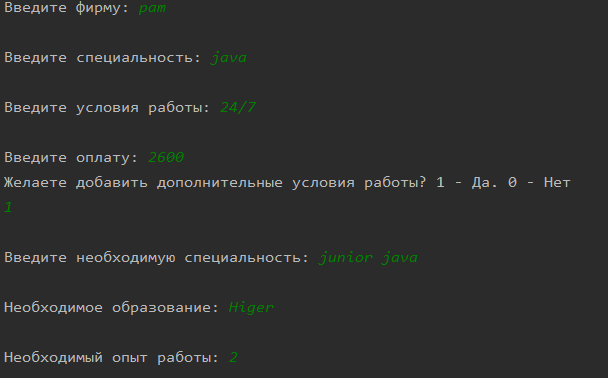
****

Рис. 7.1 – Результат роботи програми

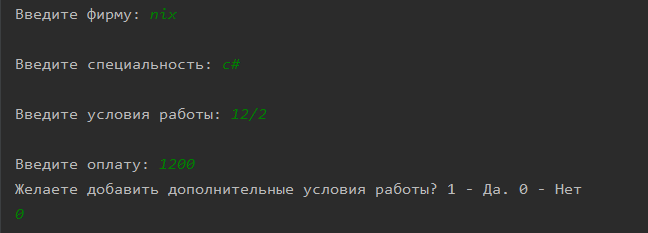


Рис. 7.2 – Результат роботи програми

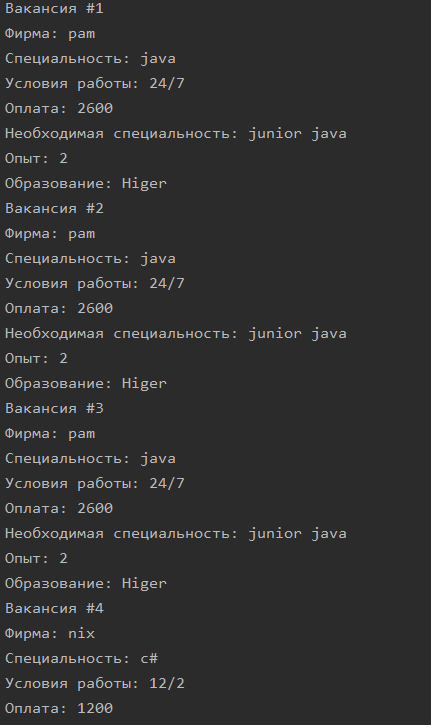


Рис. 7.3 – Результат роботи програми

Програму можна використовувати задля створення бази даних. А саме кадрове агенство, в якому представляються різноманітні вакансії. Також для вибору доступно багато інших можливостей.

**ВИСНОВКИ**

При виконанні лабораторної роботи набуто практичних навичок щодо розробки класів для заданих прикладних областей. Завдання виконано. Програма працює успішно!