Звіт

Автор: Лунгол О., КІТ101.8б

Дата: 12.12.2019

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](https://oop-khpi.github.io/#task_07_app) - domain-об’єктів.
2. Забезпечити та продемонструвати коректне введення та відображення кирилиці.
3. Продемонструвати можливість управління масивом domain-об’єктів.

ОПИС ПРОГРАМИ

**Опис змінних**

**private** **long** NumberPassport;

**private** String specialty;

**private** String workingConditions;

**private** **int** marks;

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

**private** String data;

**private** String position;

**private** String education;

**private** String department;

**Ієрархія та структура класів**

**class** Container- клас в якому описуються та знаходяться контейнеры

**class** Lab07- головний клас

**class** HDR – клас в якому данні про відділ кадрів

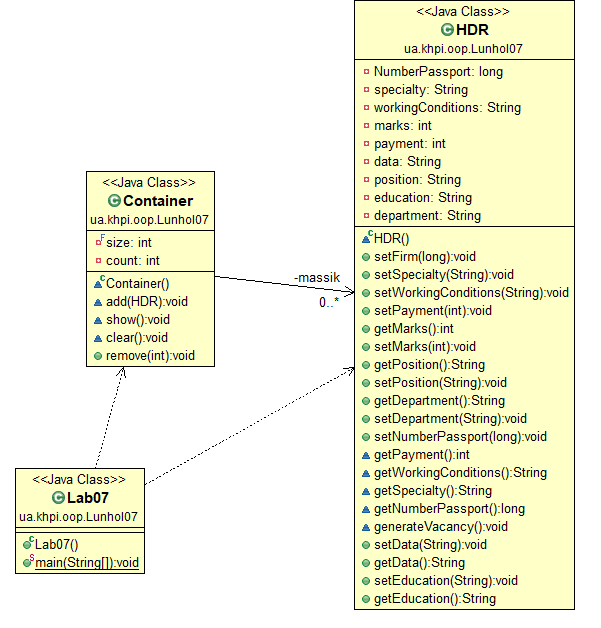


Рисунок 1 – Ієрархія класів

ТЕКСТ ПРОГРАМИ

Lab07.java

**package** ua.khpi.oop.Lunhol07;

**public** **class** Lab07 {

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

Container contains = **new** Container();

HDR rec1 = **new** HDR();

HDR rec2 = **new** HDR();

rec1.generateVacancy();

contains.add(rec1);

System.***out***.println("\n");

contains.add(rec1);

System.***out***.println("\n");

contains.add(rec1);

System.***out***.println("\n");

rec2.generateVacancy();

contains.add(rec2);

contains.show();

}

}

HDR.java

**package** ua.khpi.oop.Lunhol07;

**import** java.util.Scanner;

**class** HDR {

**private** **long** NumberPassport;

**private** String specialty;

**private** String workingConditions;

**private** **int** marks;

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

**private** String data;

**private** String position;

**private** String education;

**private** String department;

**public** **void** setFirm(**long** NumberPassport) {

**this**.NumberPassport = NumberPassport;

}

**public** **void** setSpecialty(String specialty) {

**this**.specialty = specialty;

}

**public** **void** setWorkingConditions(String workingConditions) {

**this**.workingConditions = workingConditions;

}

**public** **void** setPayment(**int** payment) {

**this**.payment = payment;

}

**public** **int** getMarks() {

**return** marks;

}

**public** **void** setMarks(**int** marks) {

**this**.marks = marks;

}

**public** String getPosition() {

**return** position;

}

**public** **void** setPosition(String position) {

**this**.position = position;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** **void** setNumberPassport(**long** numberPassport) {

NumberPassport = numberPassport;

}

**int** getPayment() {

**return** payment;

}

String getWorkingConditions() {

**return** workingConditions;

}

String getSpecialty() {

**return** specialty;

}

**long** getNumberPassport() {

**return** NumberPassport;

}

**void** generateVacancy() {

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

Scanner scan2 = **new** Scanner(System.***in***);

**int** choose = 0;

System.***out***.print("\nВведите номер паспорта: ");

NumberPassport = scan.nextLong();

System.***out***.print("\nВведите образование: ");

education = scan2.nextLine();

System.***out***.print("\nОплата: ");

payment = scan.nextInt();

System.***out***.print("\nДата взятия на должность: ");

data = scan2.nextLine();

System.***out***.print("\nВведите должность: ");

position = scan2.nextLine();

System.***out***.print("\nВведите отдел: ");

department = scan2.nextLine();

System.***out***.print("\nНабор свойств: ");

workingConditions = scan2.nextLine();

System.***out***.println("\nОценка кандидата: ");

marks = scan.nextInt();

}

**public** **void** setData(String data) {

**this**.data = data;

}

**public** String getData() {

**return** data;

}

**public** **void** setEducation(String education) {

**this**.education = education;

}

**public** String getEducation() {

**return** education;

}

}

Container.java

**package** ua.khpi.oop.Lunhol07;

**import** java.util.Arrays;

**class** Container {

/\*\*

\*/

**private** **final** **int** size = 10;

/\*\*

\* Array contains all data.

\*/

**private** **int** count = 0;

**private** HDR[] massik = **new** HDR[size];

**void** add(**final** HDR 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! Массив пустой!");

} **else** {

System.***out***.println();

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

System.***out***.println("Кандидат №" + (i+1));

System.***out***.println("Паспорт: " + massik[i].getNumberPassport());

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

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

System.***out***.println("Дата трудоустройства: " + massik[i].getData());

System.***out***.println("Должность: " + massik[i].getPosition());

System.***out***.println("Отдел: " + massik[i].getDepartment());

System.***out***.println("Набор качеств: " + massik[i].getWorkingConditions());

System.***out***.println("Оценка кандидата: " + 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--;

}

}

\* First size for array.

РОБОТА ПРОГРАМИ

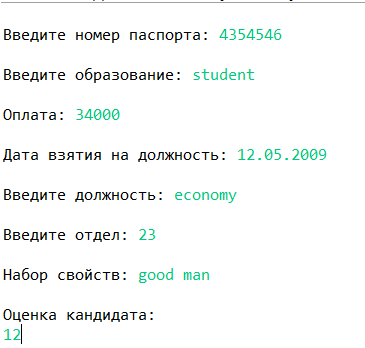


Рисунок 2- меню робочої програми

ВИСНОВОК

В даній лабораторній роботі ми навчились використанню об’єктно-орієнтованого підходу для розробки об’єкта предметної (прикладної) галузі.