Науменко Д

КІТ 101.8а

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.

Прикладна задача

Готель. Замовлення номера: паспортні дані; дати поселення та виселення; номер (клас, число місць); причина поселення (кількість не обмежена).

Код програми

Info.java

**package** lab7;

**import** java.time.LocalDate;

**import** java.util.Calendar;

**public** **class** Info {

**private** LocalDate dateOfexit;//дата выселения

**private** LocalDate dateOfent;//дата поселения

**private** Guest guest;

**public** Guest getOffender() {

**return** guest;

}

**public** **void** setOffender(Guest guest) {

**this**.guest = guest;

}

**public** Info(LocalDate dateOfCrime, LocalDate dateOfLastduresse, LocalDate dateOfLastexemption, LocalDate dateOfent, Guest guest)

{

**this**.dateOfexit=dateOfCrime;

**this**.dateOfent= dateOfent;

**this**.guest=guest;

}

**public** LocalDate getdateOfexit() {

**return** dateOfexit;

}

**public** **void** setDateOfCrime(LocalDate dateOfCrime) {

**this**.dateOfexit = dateOfexit;

}

**public** LocalDate getDateOfBirth() {

**return** dateOfent;

}

**public** **void** setDateOfBirth(LocalDate dateOfBirth) {

**this**.dateOfent = dateOfent;

}

**public** **static** **void** Show(Info hotel)

{

System.***out***.println("Фамилия: " + hotel.guest.getLast\_name());

System.***out***.println("Имя: " + hotel.guest.getName());

System.***out***.println("Отчество: " + hotel.guest.getMiddle\_name());

System.***out***.println("дата поселения: " + hotel.dateOfent);

System.***out***.println("Дата выселения: " + hotel.dateOfexit);

System.***out***.println(("Номер: "+ "люкс" ));

System.***out***.println(("Индекс: "+ "20" ));

System.***out***.println(("Причина поселения: "+ "Коммандировка" ));

}

}

Guest.java

**package** lab7;

**public** **class** Guest {

**private** String name;

**private** String last\_name;

**private** String middle\_name;

**public** Guest(String last\_name, String name, String middle\_name)

{

**this**.name=name;

**this**.last\_name=last\_name;

**this**.middle\_name=middle\_name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getLast\_name() {

**return** last\_name;

}

**public** **void** setLast\_name(String last\_name) {

**this**.last\_name = last\_name;

}

**public** String getMiddle\_name() {

**return** middle\_name;

}

**public** **void** setMiddle\_name(String middle\_name) {

**this**.middle\_name = middle\_name;

}

}

Data.java

package lab7;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.LinkedList;

import java.util.List;

public class data {

public static void main(String[] args)

{

Guest off = new Guest("Белянцев", "Татьяна", "Игоревна");

LocalDate date = LocalDate.of(2019, 11, 13);

LocalDate date1 = LocalDate.of(2010, 12, 24);

LocalDate date2 = LocalDate.of(2025, 12, 24);

LocalDate date3 = LocalDate.of(2019, 11, 11);

Info guest = new Info(date, date1, date2, date3, off);

LinkedList<Info> free = new LinkedList<>();

free.add(guest);

for(Object o: free)

{

Info.Show((Info) o);

System.out.println();

}

}

Робота програми :

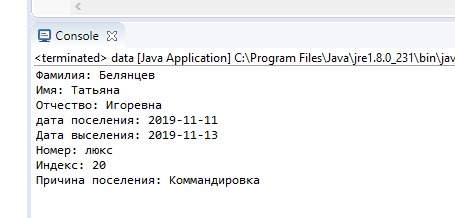


Рис. 1 – Робота програми

Ієрархія класів :

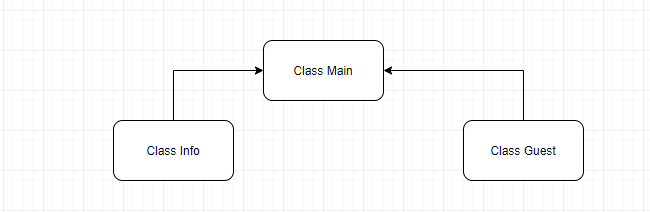


Рис. 2 – Структура класів

1. Висновок : на лабораторній роботі ми використовали об’єктно-орієнтований аналіз, реалізували класи для представлення сутностей відповідно [прикладної задачі](https://oop-khpi.github.io/#task_07_app) - domain-об’єктів.