Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 3

з дисципліни: «Кросплатформні засоби програмування»

на тему: «Класи та пакети»

**Виконав:**

студент групи КІ-35

Скіпар І.В.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2022

**Мета роботи:** ознайомитися з процесом розробки класів та пакетів мовою Java.

**Завдання (варіант № 10)**

1. Написати та налагодити програму на мові Java, що реалізує у вигляді класу предметну область згідно варіанту. Програма має задовольняти наступним вимогам:

* програма має розміщуватися в пакеті Група.Прізвище.Lab3;
* клас має містити мінімум 3 поля, що є об’єктами класів, які описують складові частини предметної області;
* клас має містити кілька конструкторів та мінімум 10 методів;
* для тестування і демонстрації роботи розробленого класу розробити клас-драйвер;
* методи класу мають вести протокол своєї діяльності, що записується у файл;
* розробити механізм коректного завершення роботи з файлом (не надіятися на метод finalize());
* програма має володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.

1. Автоматично згенерувати документацію до розробленого пакету.
2. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.
3. Дати відповідь на контрольні запитання:
4. Синтаксис визначення класу.
5. Синтаксис визначення методу.
6. Синтаксис оголошення поля.
7. Як оголосити та ініціалізувати константне поле?
8. Які є способи ініціалізації полів?
9. Синтаксис визначення конструктора.
10. Синтаксис оголошення пакету.
11. Як підключити до програми класи, що визначені в зовнішніх пакетах?
12. В чому суть статичного імпорту пакетів?
13. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

**Текст програми**

House.java

*/\*\*  
 \* Skipar package  
 \*/*package KI35.Skipar.Lab3;  
  
import java.io.FileNotFoundException;  
import java.io.PrintWriter;  
  
*/\*\*  
 \* Class <code>House</code> implement house  
 \** ***@author*** *Ілля  
 \** ***@version*** *1.0  
 \*/*public class House {  
  
 */\*\*  
 \* Default constructor  
 \** ***@throws*** *FileNotFoundException Not found log-file  
 \*/* public House() throws FileNotFoundException {  
 fout = new PrintWriter("Log.txt");  
 citizens = 0;  
 }  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *citizens count of citizens in the house  
 \** ***@throws*** *FileNotFoundException Not found log-file  
 \*/* public House(int citizens, int l, int w, int h, int br\_count, int dr\_count, int wd\_count) throws FileNotFoundException {  
 this.citizens = citizens;  
 set\_brick(l, w, h, br\_count);  
 \_door.setCount(dr\_count);  
 \_window.setCount(wd\_count);  
 fout = new PrintWriter("Log.txt");  
 }  
  
 */\*\*  
 \* Method set or change bricks in the house  
 \** ***@param*** *c count of bricks  
 \** ***@param*** *h brick height  
 \** ***@param*** *l brick length  
 \** ***@param*** *w brick width  
 \*/* public void set\_brick(int l, int w, int h, int c) {  
 \_brick = new Brick(l,w,h,c);  
 }  
  
 */\*\*  
 \* Method which return count of brick in the house  
 \** ***@return*** *count of window  
 \*/* public int get\_brick\_count() {  
 return \_brick.getCount();  
 }  
  
 */\*\*  
 \* Method which return count of door  
 \** ***@return*** *count of doors  
 \*/* public int get\_door() {  
 return \_door.getCount();  
 }  
  
 */\*\*  
 \* Method set or change count of doors in the house  
 \** ***@param*** *count doors count  
 \*/* public void set\_door(int count) {  
 \_door = new Door(count);  
 }  
  
 */\*\*  
 \* Method which return count of window  
 \** ***@return*** *count of window  
 \*/* public int get\_window() {  
 return \_window.getCount();  
 }  
  
 */\*\*  
 \* Method set or change count of windows in the house  
 \** ***@param*** *count windows count  
 \*/* public void set\_window(int count) {  
 \_window = new Window(count);  
 }  
  
 */\*\*  
 \* Method return count of citizens in the house  
 \** ***@return*** *count of citizens in the house  
 \*/* public int getCitizens() {  
 return citizens;  
 }  
  
 */\*\*  
 \* Method set or change count of citizens in the house  
 \** ***@param*** *citizens count of citizens in the house  
 \*/* public void setCitizens(int citizens) {  
 if (this.citizens<citizens)  
 fout.print(citizens-this.citizens + " new citizens in the house\n");  
 else fout.print(this.citizens-citizens + " citizens leave the house\n");  
 this.citizens = citizens;  
 fout.flush();  
 }  
  
 */\*\*  
 \* Method close log-file  
 \*/* public void closeFile () {  
 fout.close();  
 }  
  
 private Brick \_brick= new Brick();  
 private Door \_door = new Door();  
 private Window \_window = new Window();  
 private int citizens;  
 private PrintWriter fout;  
  
}

HouseApp.java

*/\*\*  
 \* Skipar package  
 \*/*package KI35.Skipar.Lab3;  
  
import java.io.IOException;  
import static java.lang.System.*out*;  
  
*/\*\*  
 \* House Application class implements main method for House  
 class possibilities demonstration  
 \** ***@author*** *Ілля  
 \** ***@version*** *1.0  
 \*/*public class HouseApp {  
  
 */\*\*  
 \** ***@param*** *args args  
 \** ***@throws*** *IOException i/o exception  
 \*/* public static void main(String[] args) throws IOException {  
 House my\_house = new House(15, 230, 120, 65,  
 10000, 4, 8);  
 *out*.print("Door: " + my\_house.get\_door());  
 *out*.print("\nWindows: " + my\_house.get\_window());  
 *out*.print("\nCitizens: " + my\_house.getCitizens());  
 *out*.print("\nBricks: " + my\_house.get\_brick\_count());  
 my\_house.set\_door(5);  
 my\_house.set\_window(10);  
 my\_house.setCitizens(15);  
 my\_house.set\_brick(230,120,80, 9500);  
 *out*.print("\n\n\nDoor: " + my\_house.get\_door());  
 *out*.print("\nWindows: " + my\_house.get\_window());  
 *out*.print("\nCitizens: " + my\_house.getCitizens());  
 *out*.print("\nBricks: " + my\_house.get\_brick\_count());  
  
 my\_house.closeFile();  
 }  
  
}

Brick.java

package KI35.Skipar.Lab3;  
  
public class Brick {  
  
 */\*\*  
 \* Default constructor  
 \*/* public Brick() {  
 height = 65;  
 width = 120;  
 length = 250;  
 count = 0;  
 }  
  
 */\*\*  
 \* Constructor  
 \*  
 \** ***@param*** *l brick length  
 \** ***@param*** *w brick width  
 \** ***@param*** *h brick height  
 \** ***@param*** *c count of bricks  
 \*/* public Brick(int l, int w, int h, int c) {  
 height = h;  
 width = w;  
 length = l;  
 count = c;  
 }  
  
 */\*\*  
 \* Method return count of bricks  
 \** ***@return*** *count of bricks  
 \*/* public int getCount() {  
 return count;  
 }  
 private int height;  
 private int width;  
 private int length;  
 private int count;  
  
}

Window.java

package KI35.Skipar.Lab3;  
  
public class Window {  
  
 */\*\*  
 \* Default constructor  
 \*/* public Window(){  
 count = 0;  
 }  
  
 */\*\*  
 \* A method which returns the count of windows  
 \** ***@return*** *count of windows  
 \*/* public int getCount() {  
 return count;  
 }  
  
 */\*\*  
 \* A method which set the count of windows  
 \** ***@param*** *count count of windows  
 \*/* public void setCount(int count) {  
 this.count = count;  
 }  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *count count of windows  
 \*/* public Window(int count) {  
 this.count = count;  
 }  
  
 private int count;  
  
}

Door.java

package KI35.Skipar.Lab3;  
  
public class Door {  
  
 */\*\*  
 \* Default constructor  
 \*/* public Door(){  
 count = 0;  
 }  
  
 */\*\*  
 \* A method which returns the count of doors  
 \** ***@return*** *count of doors  
 \*/* public int getCount() {  
 return count;  
 }  
  
 */\*\*  
 \* A method which set the count of doors  
 \** ***@param*** *count count of doors  
 \*/* public void setCount(int count) {  
 this.count = count;  
 }  
  
 */\*\*  
 \* Constructor  
 \** ***@param*** *count count of doors  
 \*/* public Door(int count) {  
 this.count = count;  
 }  
  
 private int count;  
  
}

**Результат виконання програми**

Консоль

Door: 4

Windows: 8

Citizens: 15

Bricks: 10000

Door: 5

Windows: 10

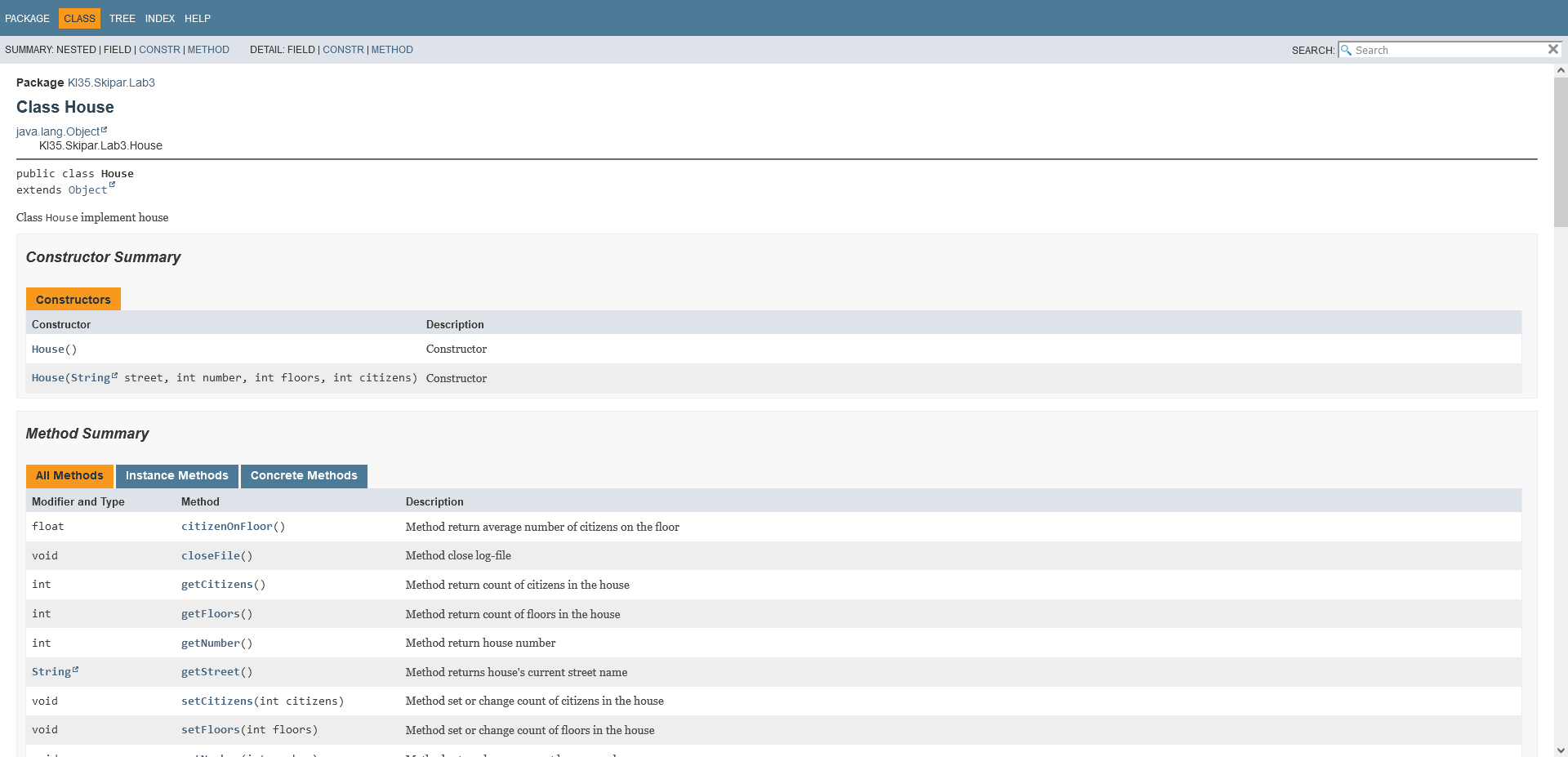
Citizens: 15

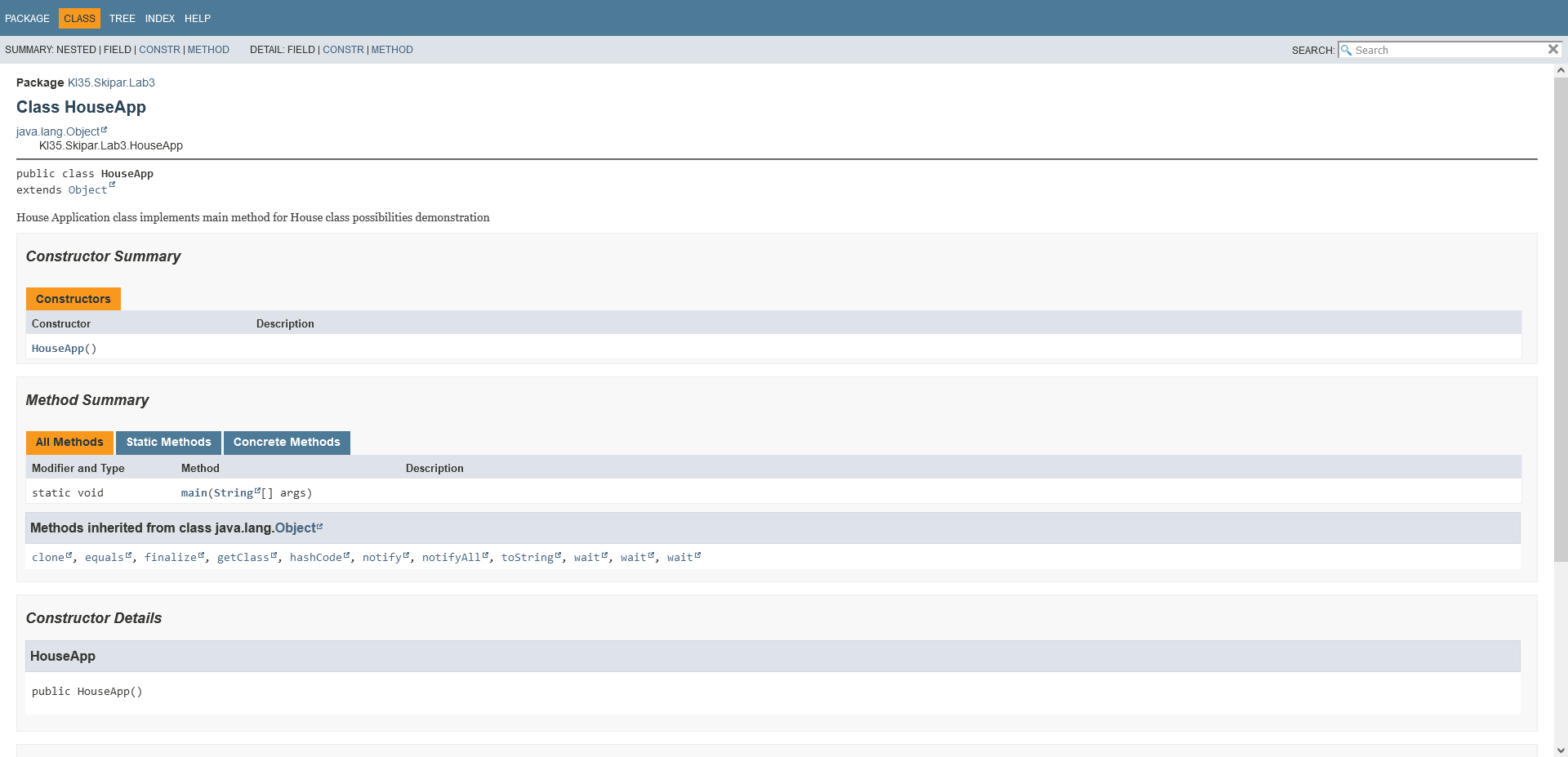
Bricks: 9500

Log.txt

0 citizens leave the house

**Фрагмент згенерованої документації**

[](doc/KI35/Skipar/Lab3/House.html)

[](doc/KI35/Skipar/Lab3/HouseApp.html)

**Відповіді на контрольні запитання**

1. Спершу пишемо специфікатор доступу (public, private, protected), потім ключове слово «class», назву класу і у фігурних дужках його реалізацію.
2. Метод визначається як функція. Специфікатор доступу, тип, назва методу, в дужках параметри які приймає метод, можливі exceptions, реалізація методу.
3. Специфікатор доступу, тип, назва поля.
4. Константне поле можна ініціалізувати за допомогою ключового слова «static».
5. Є три способи ініціалізувати поле: у конструкторі, явно при оголошенні, у блоці ініціалізації
6. Конструктор визначаємо в такому порядку: специфікатор доступу, назва класу, в круглих дужках параметри які приймає цей конструктор, у фігурних дужках тіло конструктора.
7. Пакети оголошуються за допомогою ключового слова package, тобто «package назва пакету{.назва підпакету»}.
8. Підключити до програми класи, що визначені в зовнішніх пакетах можна за допомогою ключового слова import.
9. Суть статичного імпорту пакетів полягає в тому що він дозволяє не вживати явно назву класу при звертанні до статичного поля або методу класу.
10. Використання пакетів вимагає, щоб файли і каталоги проекту та їх ієрархія були строго структурованими. Так назви пакету і його підпакетів мають співпадати з назвами каталогів, де вони розміщуються.

**Висновок**

В результаті виконання лабораторної роботи я навчився працювати з класами та потоками в Java. Частина коду автоматично згенерована середовищем IntelliJ IDEA. Пакет створений в лабораторній роботі може використовуватись в подальшому для обліку будинків в містах та кількості жителів в них.