

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Кафедра ЕОМ



Лабораторна робота №3

з дисципліни: «Кросплатформенні засоби програмування»

на тему: «Класи та пакети»

Варіант № 10

Виконав: ст. гр. КІ-34

Прийняв: Іванов Ю.С

Львів – 2022

Мета: Ознайомитися з процесом розробки класів та пакетів мовою Java.

Завдання:

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.

2. Автоматично згенерувати документацію до розробленого пакету.

3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації.

4. Дати відповідь на контрольні запитання.

Код House.java:

```
package main.kzp.lab3;

public class House {
    private String address;
    private Human owner;
    private double squareMeters;

    //Constructors
    public House() {
        LoggerInFile.printLogToFile("House is created.");
    }

    public House(String address, Human owner, double squareMeters) {
        this.address = address;
        this.owner = owner;
        this.squareMeters = squareMeters;
    }
}
```

```

        LoggerInFile.printLogToFile("House is created.");
    }

    //Getters
    public String getAddress() {
        LoggerInFile.printLogToFile("Get address: " + address + '.');
        return address;
    }

    public Human getOwner() {
        LoggerInFile.printLogToFile("Get owner: " + owner + '.');
        return owner;
    }

    public double getSquareMeters() {
        LoggerInFile.printLogToFile("Get square meters: " + squareMeters + '.');
        return squareMeters;
    }

    //Setters
    public void setAddress(String address) {
        LoggerInFile.printLogToFile("Set address: " + address + '.');
        this.address = address;
    }

    public void setOwner(Human owner) {
        LoggerInFile.printLogToFile("Set owner: " + owner + '.');
        this.owner = owner;
    }

    public void setSquareMeters(double squareMeters) {
        LoggerInFile.printLogToFile("Set square meters: " + squareMeters + '.');
        this.squareMeters = squareMeters;
    }

    @Override
    public String toString() {
        LoggerInFile.printLogToFile("Call toString from House class.");
        return "House:\n" +
            "\taddress: " + address + '\n' +
            "\towner: " + owner + '\n' +
            "\tsquareMeters: " + squareMeters + '\n';
    }
}

```

Код Human.java:

```

package main.kzp.lab3;

public class Human {
    private String firstName, lastName;

    //Constructors
    public Human() {
        LoggerInFile.printLogToFile("Human is created.");
    }

    public Human(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
}

```

```

        LoggerInFile.printLogToFile("Human is created.");
    }

    //Getters
    public String getFirstName() {
        LoggerInFile.printLogToFile("Get firstname: " + firstName + '.');
        return firstName;
    }

    public String getLastName() {
        LoggerInFile.printLogToFile("Get lastname: " + lastName + '.');
        return lastName;
    }

    //Setters
    public void setFirstName(String firstName) {
        LoggerInFile.printLogToFile("Set firstname: " + firstName + '.');
        this.firstName = firstName;
    }

    public void setLastName(String lastName) {
        LoggerInFile.printLogToFile("Set lastname: " + lastName + '.');
        this.lastName = lastName;
    }

    @Override
    public String toString() {
        LoggerInFile.printLogToFile("Call toString from Human class.");
        return lastName + " " + firstName;
    }
}

```

Код LoggerInFile.java:

```

package main.kzp.lab3;

import java.io.*;

public class LoggerInFile {

    public static void printLogToFile(String data) {
        try(FileWriter fileWriter = new FileWriter("Lab3Logger.txt", true)){
            fileWriter.write(data + '\n');
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}

```

Код Main.java:

```

package main.kzp.lab3;

public class Main {
    public static void main(String[] args) {
        House house = new House();

        house.setAddress("City: Lviv, St. Stepana Bandera 12");
        house.setOwner(new Human("Yuriy", "Bobaylo"));
        house.setSquareMeters(2000.0);
    }
}

```

```
        System.out.println(house.toString());  
    }  
}
```

Скріншоти програми:

```
House:  
    address: City: Lviv, St. Stepana Bandera 12  
    owner: Bobaylo Yuriy  
    squareMeters: 2000.0  
  
Process finished with exit code 0
```

Рис. 1. Результат роботи програми.

```
1  House is created.  
2  Set address: City: Lviv, St. Stepana Bandera 12.  
3  Human is created.  
4  Call toString from Human class.  
5  Set owner: Bobaylo Yuriy.  
6  Set square meters: 2000.0.  
7  Call toString from House class.  
8  Call toString from Human class.  
9
```

Рис. 2. Логуювання роботи програми.

Висновок: Я ознайомився з процесом розробки класів та пакетів мовою Java.

Контрольні питання:

1. Синтаксис визначення класу.

Відповідь: [public] class НазваКласу

```
{  
    [конструктори]  
    [методи]  
    [поля]  
}
```

2. Синтаксис визначення методу.

Відповідь: [СпецифікаторДоступу] [static] [final] Тип назваМетоду([параметри])
[throws класи]

{

[Тіло методу]

[return [значення]];

}

3. Синтаксис оголошення поля.

Відповідь: [СпецифікаторДоступу] [static] [final] Тип НазваПоля [= ПочатковеЗначення];

4. Як оголосити та ініціалізувати константне поле?

Відповідь: [СпецифікаторДоступу] static final Тип НазваПоля = Значення;

- явно при оголошенні поля класу;

- у статичному блоці ініціалізації.

5. Які є способи ініціалізації полів?

Відповідь:

6. Синтаксис визначення конструктора.

Відповідь: [СпецифікаторДоступу] НазваКласу([параметри])

{

Тіло конструктора

}

7. Синтаксис оголошення пакету.

Відповідь: package НазваПакету{.НазваПідпакету};

8. Як підключити до програми класи, що визначені в зовнішніх пакетах?

Відповідь: вказуючи повне ім'я пакету перед іменем кожного класу або використовуючи оператор import

9. В чому суть статичного імпорту пакетів?

Відповідь: Можливість імпортувати окремі статичні методи або поля класу

- import static

НазваПакету{.НазваПідпакету}.НазваКласу.НазваСтатичногоМетодуАбоПоля;

- import static НазваПакету{.НазваПідпакету}.*;

10. Які вимоги ставляться до файлів і каталогів при використанні пакетів?

Відповідь: Для уникнення конфліктів імен не зловживати імпортом пакетів.