

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра автоматизованих систем управління



Звіт
до лабораторної роботи № 5
з дисципліни
Прикладне програмування
на тему:
“Консольне меню”

Виконав: студент КН-105
Руденко Руслан

Прийняв:
Мельник Р.В.

Лабораторна робота № 5

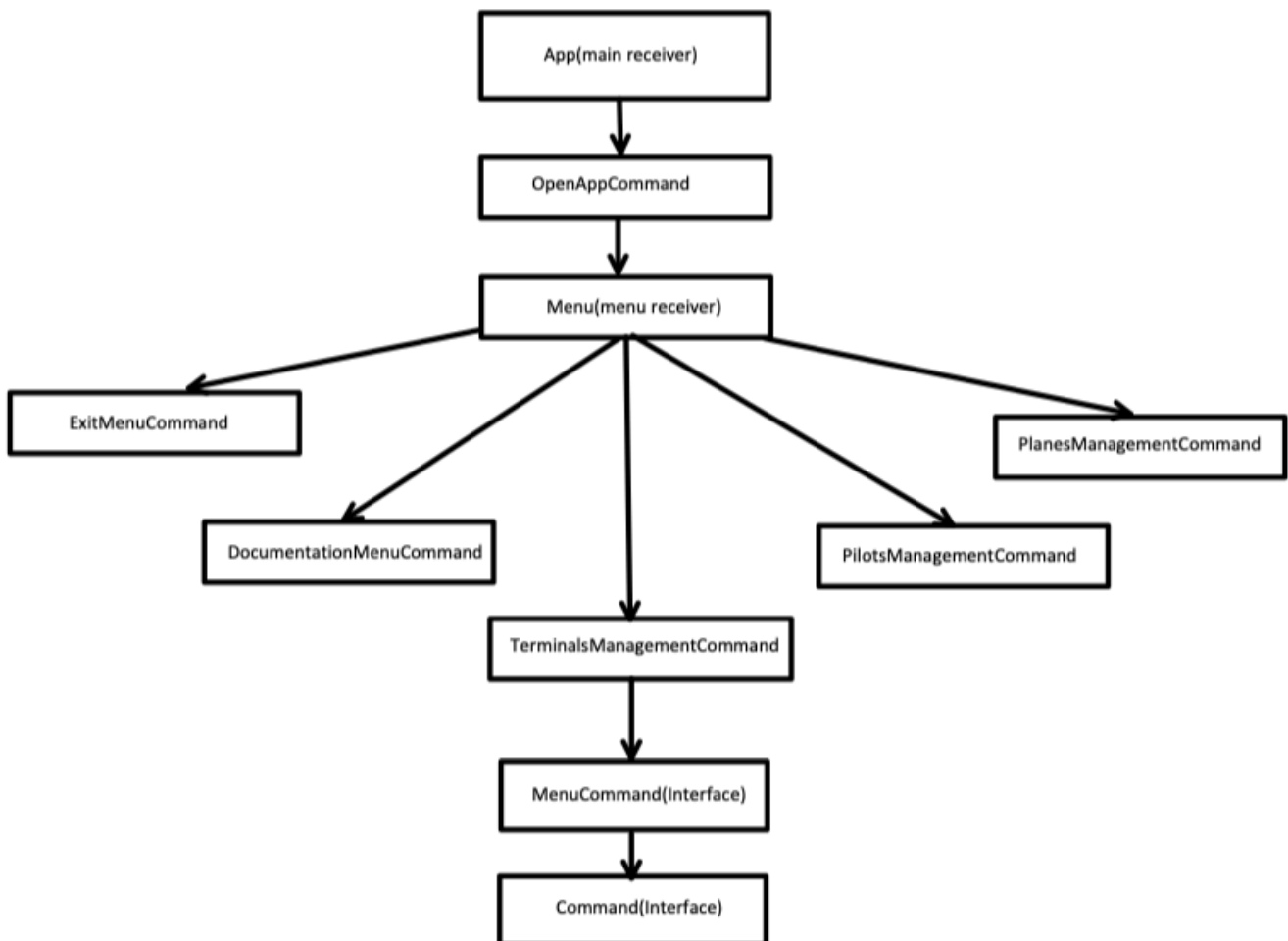
Тема роботи: “Консольне меню”

Мета роботи: здобути застосування паттерну команда для розпрбки програм мовою Java

Завдання лабораторної роботи(9 варіант):

Реалізуйте консольне меню до вашого завдання з використанням паттерну «Команда». На цьому етапі не потрібно реалізовувати функціонал конкретних пунктів меню – тільки створити «кістяк» вашої програми.

Class діаграма



Текст програми

```
package lab5;
```

```
import lab5.commands.Command;
import lab5.commands.OpenMenuCommand;
import lab5.menu.*;

public class Lab5 {
    public static void main(String[] args) {
        Menu menu = new Menu(new MenuCommand[]{
            new PilotsManagementMenuCommand(),
            new PlanesManagementMenuCommand(),
            new TerminalsManagementMenuCommand(),
            new ScheduleManagementMenuCommand(),
            new DocumentationMenuCommand(),
            new ExitMenuCommand()
        });
        Command openMenuCommand = new OpenMenuCommand(menu);
        new App(openMenuCommand).startApp();
    }
}
```

```
package lab5;
```

```
import lab5.commands.Command;

public class App {
    private final Command openAppCommand;

    public App(Command openAppCommand) {
        this.openAppCommand = openAppCommand;
    }

    public void startApp() {
        openAppCommand.execute();
    }
}
```

```
package lab5.model;
```

```
public class Terminal {
    private int id;
    private String location;
    private int maxCapacity;
}
```

```

package lab5.model;

public enum PlaneType {
    PRIVATE, CARGO, COMMERCIAL
}

package lab5.model;

public class Plane {
    private int id;
    private PlaneType type;
    private double fuelConsumption;
    private int capacity;
    private double maxFlightDistance;
}

package lab5.model;

public enum PilotType {
    CAPTAIN, SECOND_PILOT
}

package lab5.model;

import java.util.Date;

public class Pilot {
    private int id;
    private Date birthday;
    private int salary;
    private double flyingHours;
    private PilotType type;
}

package lab5.commands;

public interface Command {
    void execute();
}

package lab5.commands;

import lab5.menu.Menu;

public class OpenMenuCommand implements Command {
    private final Menu menu;
}

```

```

    public OpenMenuCommand(Menu menu) {
        this.menu = menu;
    }

    @Override
    public void execute() {
        menu.startShowing();
    }
}

package lab5.menu;

public class TerminalsManagementMenuCommand implements
MenuCommand {
    @Override
    public void execute() {
        System.out.println("TODO add handlers");
    }

    @Override
    public String menuOptionString() {
        return "Terminals management";
    }
}

package lab5.menu;

public class ScheduleManagementMenuCommand implements
MenuCommand {
    @Override
    public void execute() {
        System.out.println("TODO add handlers");
    }

    @Override
    public String menuOptionString() {
        return "Schedule management";
    }
}

package lab5.menu;

public class PlanesManagementMenuCommand implements
MenuCommand {
    @Override
    public void execute() {
        System.out.println("TODO add handlers");
    }
}

```

```

        @Override
        public String menuOptionString() {
            return "Planes management";
        }
    }
}

```

```
package lab5.menu;
```

```
public class PilotsManagementMenuCommand implements
MenuCommand {

```

```

    @Override
    public void execute() {
        System.out.println("TODO add handlers");
    }

    @Override
    public String menuOptionString() {
        return "Pilots management";
    }
}

```

```
package lab5.menu;
```

```
import lab5.commands.Command;
```

```
public interface MenuCommand extends Command {
    String menuOptionString();
}

```

```
package lab5.menu;
```

```
import java.util.Scanner;
```

```
public class Menu {
    private final MenuCommand[] menuOptions;

    public Menu(MenuCommand[] menuOptions) {
        this.menuOptions = menuOptions;
    }

    public void startShowing() {
        printMenu();
    }

    private void printMenu() {

```

```

        System.out.println("=====Menu=====");
        for (int i = 0; i < menuOptions.length; i++) {
            System.out.println((i + 1) + ". " +
menuOptions[i].menuOptionString());
        }

        handleMenuInput();
    }

    private void handleMenuInput() {
        Scanner input = new Scanner(System.in);

        String menuItem = input.nextLine();
        try {
            int menuNum = Integer.parseInt(menuItem) - 1;

            menuOptions[menuNum].execute();
        } catch (Exception e) {
            System.out.println("Entered invalid menu number!
Please try again.\n");
        } finally {
            printMenu();
        }
    }
}

```

```
package lab5.menu;
```

```

public class ExitMenuCommand implements MenuCommand {
    @Override
    public void execute() {
        System.exit(0);
    }

    @Override
    public String menuOptionString() {
        return "Exit";
    }
}

```

```
package lab5.menu;
```

```

public class DocumentationMenuCommand implements MenuCommand {

    @Override
    public void execute() {
        System.out.println("TODO add a handler");
    }
}

```

```
@Override  
public String menuOptionString() {  
    return "Documentation";  
}  
}
```

Висновок

В ході лабораторної роботи я вивчив та застосував на практиці паттерн “Команда”. Це доволі потужний інструмент за допомогою якого можна будувати гнучкі програми де інкапсульовано виклик команд та її реалізація, що дає змогу швидко замінити реалізації без додаткових зайвих змін.