Міністерство освіти і науки України

Національний університет «Львівська Політехніка»

Інститут комп’ютерних наук та інформаційних технологій

**ЛАБОРАТОРНА РОБОТА №4**

З дисципліни **«**Прикладне програмування**»**

Тема:

**«Комплексний проект»**

|  |
| --- |
| **Виконав:** |
| Гуменюк М.І. |
| студент групи  КН-204 |

|  |
| --- |
| **Прийняв:** |
| Вергун В.Р. |
|  |

Львів 2020

**Індивідуальне завдання**

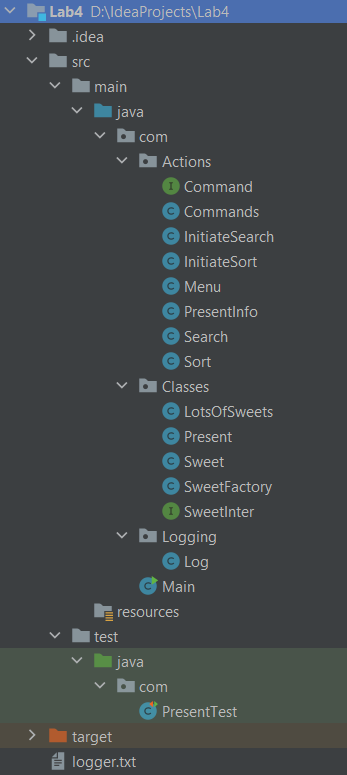
Новорічний подарунок. Визначити ієрархію цукерок та інших солодощів. Створити

кілька об’єктів-цукерок. Зібрати дитячий подарунок з визначенням його ваги.

Здійснити сортування цукерок в подарунку на основі одного з параметрів. Знайти

цукерку в подарунку, що відповідає заданому діапазону вмісту цукру.

**Класи та пакети**

****

**Тестування класу Present**

package com;  
  
import com.Classes.Present;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class PresentTest {  
  
 @Test  
 void estimateWeight() {  
 Present test = new Present(5);  
 *assertEquals*(21, test.estimateWeight());  
 }  
}

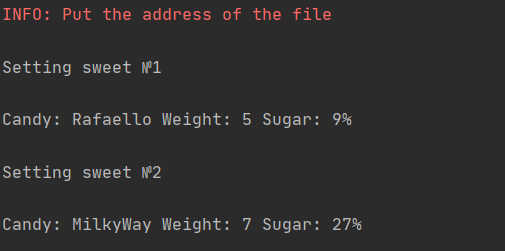
**Логгер**

package com.Logging;  
import java.io.File;  
import java.io.IOException;  
import java.util.logging.FileHandler;  
import java.util.logging.Logger;  
import java.util.logging.SimpleFormatter;  
  
  
public class Log {  
 private Logger logger;  
 private FileHandler handler;  
  
 public Log(String filename) throws SecurityException,IOException {  
 File file = new File(filename);  
 if(!file.exists()) {  
 file.createNewFile();  
 logger.setUseParentHandlers(false);  
 }  
 handler = new FileHandler(filename, true);  
 logger = Logger.*getLogger*("Logger");  
 logger.addHandler(handler);  
 handler.setFormatter(new SimpleFormatter());  
 }  
  
 public Logger getLogger(){ return logger; }  
}

**Головне меню**

package com.Actions;  
import com.Classes.Present;  
import com.Logging.Log;  
  
  
import java.io.IOException;  
import java.util.Scanner;  
  
public class Menu {  
 private Log log;  
 {  
 try{  
 log = new Log("logger.txt");  
 }catch (IOException a){  
 a.printStackTrace();  
 }  
 }  
  
 public void presentOperation() {  
 log.getLogger().info("Put the address of the file");  
 Present obj = new Present();  
 Command Info = new PresentInfo(obj);  
 Command Sort = new InitiateSort(obj);  
 Command Find = new InitiateSearch(obj);  
 Commands start = new Commands(Info,Find,Sort);  
 Scanner enter = new Scanner(System.*in*);  
 int choice;  
 System.*out*.println("\nOur present is completed! Let's do something with it");  
 System.*out*.println("\nPress: \n1 - Watch present info \n2 - Search sweets with parameters " +  
 "\n3 - Sort present \n4 - Exit");  
 do {  
 log.getLogger().info("run 'PrintMenuInfo' function");  
 choice = enter.nextInt();  
 switch (choice) {  
 case 1 -> {  
 log.getLogger().info("run 'PresentInfo' function");  
 start.PresentInfo();  
 }  
 case 2 -> {  
 log.getLogger().info("run 'Search' function");  
 start.Search();  
 }  
 case 3 -> {  
 log.getLogger().info("run 'Sort' function");  
 start.Sort();  
 }  
 case 4 -> {  
 log.getLogger().info("exit program");  
 System.*exit*(0);  
 }  
 default -> {  
 log.getLogger().info("invalid request in menu");  
 System.*out*.println("invalid request");  
 System.*exit*(1);  
 }  
 }  
 }while(true);  
 }  
  
 public void toBegin() {  
 presentOperation();  
 }

**Результат виконання програми**

****

