Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**ЗВІТ**

Про виконання лабораторної роботи **№ 8**

**«Логгер»**

з дисципліни **«Прикладне програмування»**

**Лектор:**

доц. кафедри АСУ

Скорохода О.В.

**Виконав:**

студ. групи КН-203

Репей А.В.

**Прийняв:**

асис. кафедри АСУ

Мельник Р. В.

«\_\_\_» \_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2022

**Тема:** Логгер.

**Мета:** Додайте логгер з логуванням основних подій та виняткових ситуацій. При цьому дані логу мають зберігатися у файлі, а критичні помилки додатково надсилатися на e-mail.

**Посилання на репозиторій Github**: https://github.com/temerep/labs\_ap/tree/main/knight\_ammunition

**Хід виконання лабораторної роботи:**

Створити консольну програму на мові програмування Java “Лицарі”.

Визначити ієрархію амуніції лицаря. Екіпірувати лицаря.

Підрахувати вартість. Здійснити сортування амуніції на основі ваги. Знайти елементи амуніції, що відповідають заданому діапазону параметрів ціни.

Окрім цього, програма повинна відповідати таким додатковим вимогам:  
• Використовувати можливості ООП: класи, успадкування, поліморфізм, інкапсуляцію.  
• Кожний клас повинен мати назву, яка повністю описує його суть, і інформативний склад.  
• Успадкування потрібно використовувати тільки тоді, коли воно має сенс.  
• При записі програми потрібно використовувати домовленості щодо оформлення коду java code convention.  
• Класи потрібно грамотно розкласти по пакетах.  
• Для збереження параметрів ініціалізації потрібно використовувати файли.  
• Повинне бути консольне меню для вибору різних дій, наприклад: «завантажити з файлу», «довідка», «вихід» тощо.

**Додаткове завдання**

Додайте логгер з логуванням основних подій та виняткових ситуацій. При цьому дані логу мають зберігатися у файлі, а критичні помилки додатково надсилатися на e-mail.

**Хід виконання лабораторної роботи**

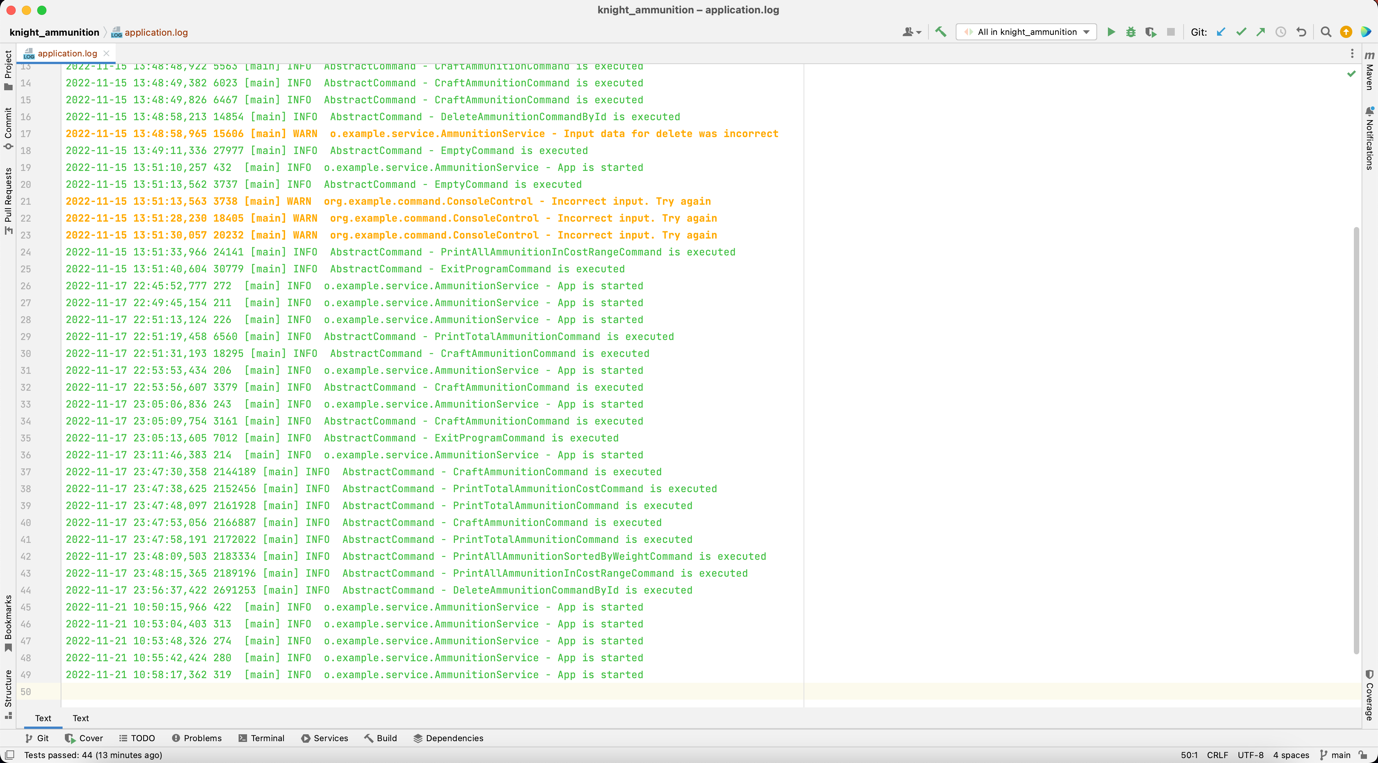
Для підключення логера logback і можливості надсилання логів на e-mail додано наступні залежності в pom.xml:

<**dependency**>  
 <**groupId**>ch.qos.logback</**groupId**>  
 <**artifactId**>logback-classic</**artifactId**>  
 <**version**>1.2.9</**version**>  
</**dependency**>  
  
<**dependency**>  
 <**groupId**>ch.qos.logback</**groupId**>  
 <**artifactId**>logback-core</**artifactId**>  
 <**version**>1.2.9</**version**>  
</**dependency**>  
  
<**dependency**>  
 <**groupId**>org.slf4j</**groupId**>  
 <**artifactId**>slf4j-api</**artifactId**>  
 <**version**>1.7.30</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>  
  
*<!-- https://mvnrepository.com/artifact/com.sun.mail/javax.mail -->*<**dependency**>  
 <**groupId**>com.sun.mail</**groupId**>  
 <**artifactId**>javax.mail</**artifactId**>  
 <**version**>1.6.2</**version**>  
</**dependency**>  
  
  
*<!-- Log4j -->  
<!-- https://mvnrepository.com/artifact/javax.activation/activation -->*<**dependency**>  
 <**groupId**>javax.activation</**groupId**>  
 <**artifactId**>activation</**artifactId**>  
 <**version**>1.1.1</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.slf4j</**groupId**>  
 <**artifactId**>slf4j-api</**artifactId**>  
 <**version**>1.7.36</**version**>  
</**dependency**>

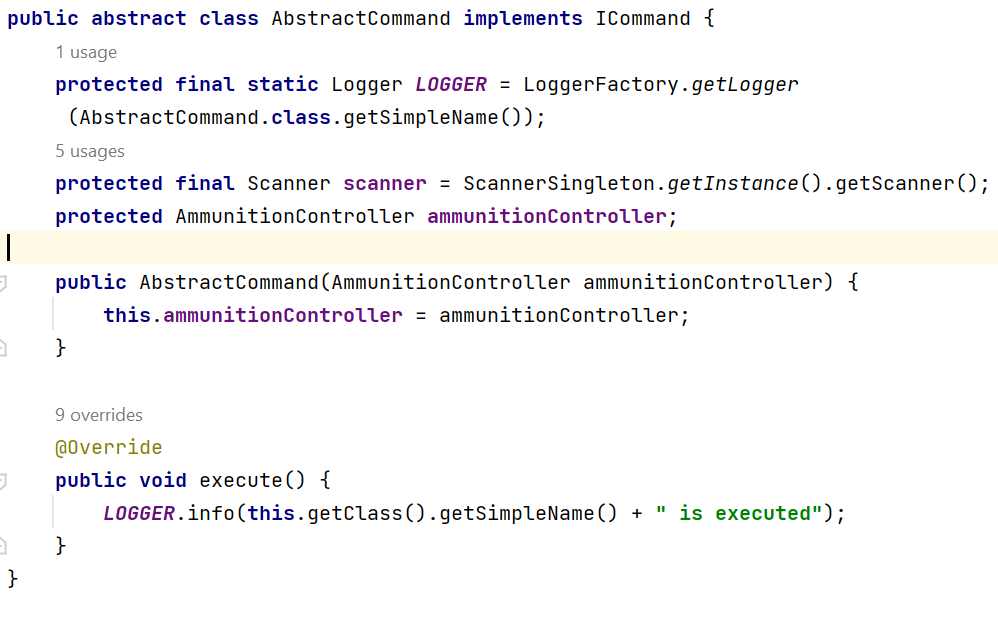
Створено файл конфігурацій src/main/resources/logback.xml із наступним наповненням:

*<?***xml version="1.0" encoding="UTF-8"***?>*<**configuration scan="true"**>  
 <**appender name="consoleAppender" class="ch.qos.logback.core.ConsoleAppender"**>  
 <**encoder**>  
 <**charset**>UTF-8</**charset**>  
 <**Pattern**>%msg%n</**Pattern**>  
 </**encoder**>  
 </**appender**>  
  
 <**appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender"**>  
 <**file**>application.log</**file**>  
 <**rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy"**>  
<**fileNamePattern**>/srv/logs/application\_%d{yyyy-MM-dd}.%i.log</**fileNamePattern**>  
  
 <**timeBasedFileNamingAndTriggeringPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP"**>  
 <**maxFileSize**>5MB</**maxFileSize**>  
 </**timeBasedFileNamingAndTriggeringPolicy**>  
 *<!-- keep 30 days' worth of history -->* <**maxHistory**>30</**maxHistory**>  
 </**rollingPolicy**>  
  
 <**encoder**>  
 <**charset**>UTF-8</**charset**>  
 <**pattern**>%d %-4relative [%thread] %-5level %logger{35} - %msg%n</**pattern**>  
 </**encoder**>  
 </**appender**>  
  
 <**appender name="emailAppender" class="ch.qos.logback.classic.net.SMTPAppender"**>  
 <**appender name="emailAppender" class="ch.qos.logback.classic.net.SMTPAppender"**>  
 <**smtpHost**>smtp.gmail.com</**smtpHost**>  
 <**smtpPort**>587</**smtpPort**>  
 <**STARTTLS**>true</**STARTTLS**>  
 <**asynchronousSending**>false</**asynchronousSending**>  
 <**username**>example1234@gmail.com</**username**>  
 <**password**>example1234</**password**>  
 <**to**>random1234@gmail.com</**to**>  
 <**from**>example1234@gmail.com</**from**>  
 <**subject**>BAELDUNG: %logger{20} - %msg</**subject**>  
 <**layout class="ch.qos.logback.classic.html.HTMLLayout"**/>  
 </**appender**>  
 <**cyclicBufferTracker class="ch.qos.logback.core.spi.CyclicBufferTracker"**>  
 <**bufferSize**>1</**bufferSize**>  
 </**cyclicBufferTracker**>  
 </**appender**>  
  
 <**root level="DEBUG"**>  
 <**appender-ref ref="consoleAppender"** />  
 <**appender-ref ref="FILE"**/>  
 <**appender-ref ref="emailAppender"**/>  
 </**root**>  
</**configuration**>

Створені апендери для консолі, файлу application.log і smtp-апендер.

***Вміст application.log:***

Логери додано на рівні AbstractCommand#execute



Також в блоках catch при перехопленні виключень в AmmunitionController:

**package** org.example.controller;  
  
**import** org.slf4j.LoggerFactory;  
**import** org.slf4j.Logger;  
**import** org.example.entity.Ammunition;  
**import** org.example.service.AmmunitionService;  
**import** org.example.utils.ReflectionUtils;  
**import** org.example.view.AmmunitionView;  
  
**import** java.util.List;  
  
**public class** AmmunitionController {  
 **private static final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(AmmunitionService.**class**.getName());  
 AmmunitionService **ammunitionService**;  
 AmmunitionView **ammunitionView**;  
  
 **public** AmmunitionController(AmmunitionService ammunitionService) {  
 **this**.**ammunitionService** = ammunitionService;  
 **ammunitionView** = **new** AmmunitionView(**this**, ammunitionService);  
 }  
  
 **public void** initView() {  
 ***LOGGER***.info(**"App is started"**);  
 **ammunitionView**.initView();  
 }  
  
 **public void** updateView() {  
 **ammunitionView**.updateView();  
 }  
  
  
 **public void** findAll() {  
 **try** {  
 **ammunitionView**.printAmmunition(**ammunitionService**.findAll());  
 } **catch** (Exception e) {  
  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** findEntityById(**int** id) {  
 **try** {  
 Ammunition ammunitionById = **ammunitionService**.findEntityById(id);  
 **ammunitionView**.printAmmunitionById(id, ammunitionById);  
 } **catch** (java.lang.Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** delete(Ammunition ammunition){  
 **try** {  
 **ammunitionService**.delete(ammunition);  
 **ammunitionView**.printDeleted(ammunition);  
 } **catch** (Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** delete(**int** id) {  
 **try** {  
 Ammunition entityById = **ammunitionService**.findEntityById(id);  
 **ammunitionService**.delete(id);  
 **ammunitionView**.printDeleted(entityById);  
 } **catch** (java.lang.Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** create(Ammunition ammunition) {  
 **try** {  
 **ammunitionService**.create( ammunition);  
 **ammunitionView**.printCreated(ammunition);  
 } **catch** (Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
  
 **public void** computeTotalEquippedAmmunitionCost() {  
 **try** {  
 **double** l = **ammunitionService**.computeTotalEquippedAmmunitionCost();  
 **ammunitionView**.printTotalEquippedAmmunitionCost(l);  
 } **catch** (java.lang.Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** getAmmunitionListSortedByWeight() {  
 **try** {  
 List<Ammunition> ammunitionListSortedByWeight = **ammunitionService**.getAmmunitionListSortedByWeight();  
 **ammunitionView**.printAmmunitionSortedByWeight(ammunitionListSortedByWeight);  
 } **catch** (java.lang.Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
  
 **public void** getAmmunitionListInCostRange(**long** inclusiveStart, **long** inclusiveEnd) {  
 **try** {  
 List<Ammunition> ammunitionListInCostRange = **ammunitionService**.getAmmunitionListInCostRange(inclusiveStart, inclusiveEnd);  
 **ammunitionView**.printAmmunitionInCostRange(ammunitionListInCostRange, inclusiveStart,inclusiveEnd);  
 } **catch** (java.lang.Exception e) {  
 ***LOGGER***.warn(**"Input data for "**+ReflectionUtils.*getCurrentExecutingMethod*() + **" was incorrect"**);  
 *// throw new RuntimeException(e);* }  
 }  
}

**Висновок**

На лабораторній роботі я додав логгер з логуванням основних подій та виняткових ситуацій. При цьому дані логу зберігаються у файлі, а критичні помилки додатково мають можливість надсилання на e-mail.