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 **№ 7**

**«Покриття програми юніт тестами»**

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:** Покриття програми юніт тестами.

**Мета:** Покрити програму юніт тестами (coverage >= 90%).

**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**

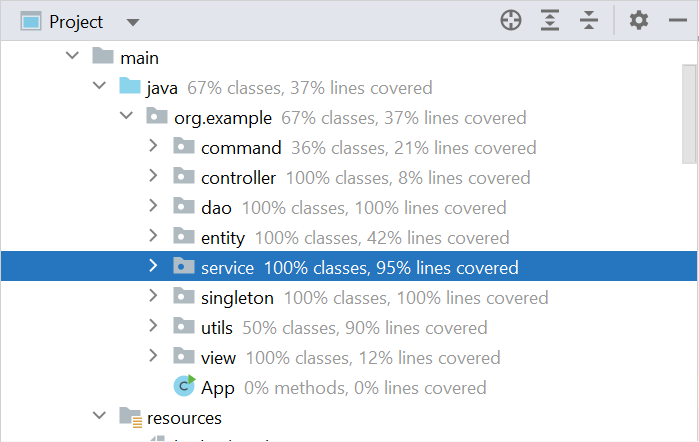
Покрийте вашу програму юніт тестами (coverage >= 90%).

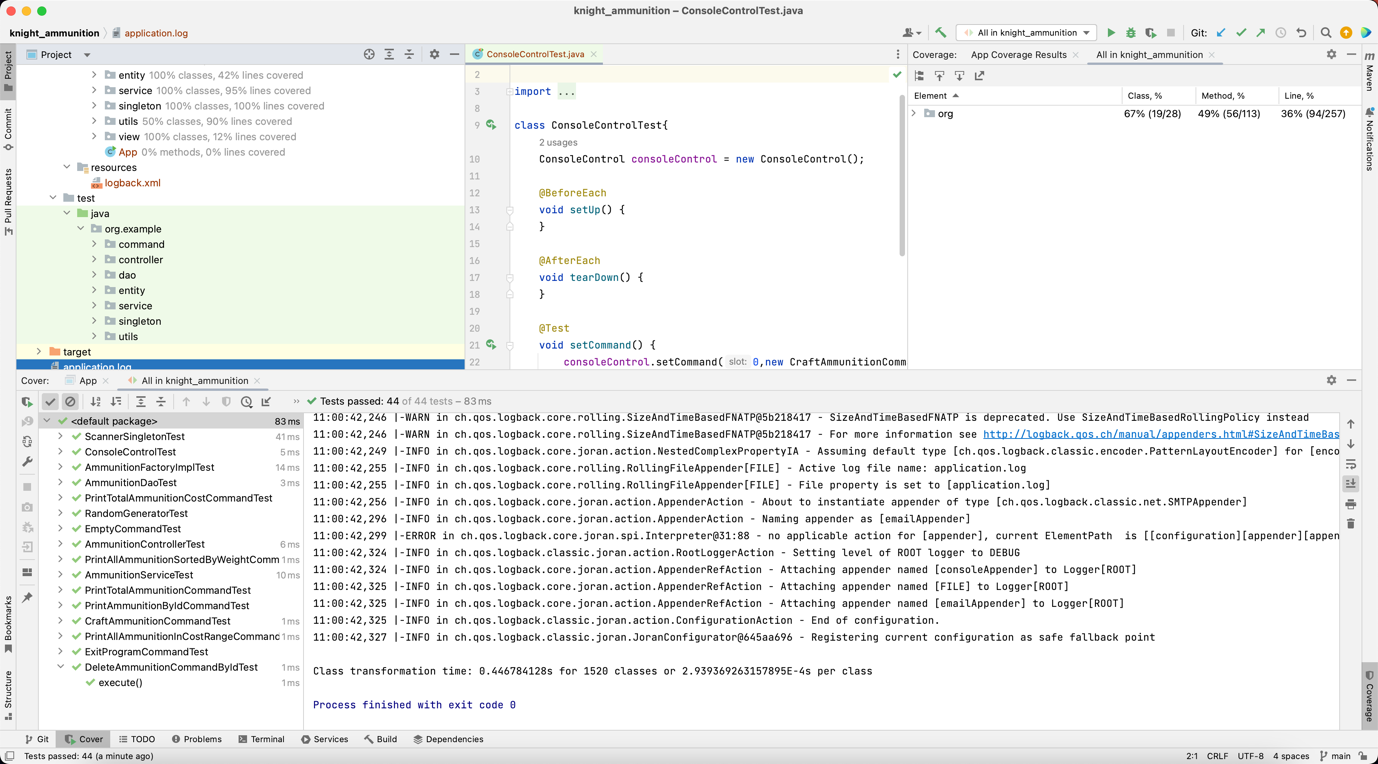
**Хід виконання лабораторної роботи**

Додано в pom.xml Maven-dependency для Junit5:

<**dependency**>  
 <**groupId**>org.junit.jupiter</**groupId**>  
 <**artifactId**>junit-jupiter-api</**artifactId**>  
 <**version**>5.8.2</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>

Покриття тестів:



****

***AmmunitionDaoTest:***

**package** org.example.dao;  
  
**import** org.example.entity.Ammunition;  
**import** org.example.entity.Helmet;  
**import** org.example.entity.Shield;  
**import** org.example.entity.VisionLevel;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.List;  
  
**import static** org.junit.jupiter.api.Assertions.\*;  
**import static** org.junit.jupiter.api.Assertions.*assertEquals*;  
  
**class** AmmunitionDaoTest {  
 **private static** AmmunitionDao<Ammunition> *ammunitionDao*;  
 Helmet **helmet** = **new** Helmet(**"h1"**, 1, 1, 1, VisionLevel.***HALF\_FACE***);  
  
 @org.junit.jupiter.api.BeforeEach  
 **void** setUp() {  
 *ammunitionDao* = **new** AmmunitionDao<>();  
 *ammunitionDao*.**storage**.add(**helmet**);  
 }  
  
 @org.junit.jupiter.api.AfterEach  
 **void** tearDown() {  
 *ammunitionDao*.**storage**.clear();  
 *ammunitionDao*.**storage** = **null**;  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** testFindAll() **throws** Exception {  
 *assertEquals*(List.*of*(**helmet**), *ammunitionDao*.findAll());  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** testFindEntityById() **throws** Exception {  
 *assertEquals*(**helmet**, *ammunitionDao*.findEntityById(0));  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** testDeleteById() **throws** Exception {  
 *ammunitionDao*.delete(0);  
 *assertEquals*(0, *ammunitionDao*.**storage**.size());  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** testDeleteByEntity() **throws** Exception {  
 *ammunitionDao*.delete(**helmet**);  
 *assertEquals*(0, *ammunitionDao*.**storage**.size());  
 }  
  
 @org.junit.jupiter.api.Test  
 **void** testCreate() **throws** Exception {  
 *ammunitionDao*.create(**new** Shield(**"sh"**, 1, 1, 1, 1));  
 *assertEquals*(2,*ammunitionDao*.**storage**.size());  
 }  
}

***Тест роботи фабрики:***

**package** org.example.entity;  
  
**import** org.junit.jupiter.api.Test;  
  
**import static** org.junit.jupiter.api.Assertions.\*;  
  
**class** AmmunitionFactoryImplTest {  
 AmmunitionFactory **ammunitionFactory** = **new** AmmunitionFactoryImpl();  
 @Test  
 **void** createShield() {  
 *assertEquals*(**"Shield"**,**ammunitionFactory**.createShield().getClass().getSimpleName());  
 }  
  
 @Test  
 **void** createSword() {  
 *assertEquals*(**"Sword"**,**ammunitionFactory**.createSword().getClass().getSimpleName());  
 }  
  
 @Test  
 **void** createChestPlate() {  
 *assertEquals*(**"ChestPlate"**,**ammunitionFactory**.createChestPlate().getClass().getSimpleName());  
 }  
  
 @Test  
 **void** createHelmet() {  
 *assertEquals*(**"Helmet"**,**ammunitionFactory**.createHelmet().getClass().getSimpleName());  
 }  
  
 @Test  
 **void** createRandomAmmunition() {  
 *assertTrue*(**ammunitionFactory**.createHelmet() **instanceof** Ammunition);  
 }  
}

***AmmunitionServiceTest:***

**package** org.example.service;  
  
**import** org.example.dao.AmmunitionDao;  
**import** org.example.entity.Ammunition;  
**import** org.example.entity.ChestPlate;  
**import** org.example.entity.Helmet;  
**import** org.example.entity.VisionLevel;  
**import** org.junit.jupiter.api.AfterEach;  
**import** org.junit.jupiter.api.BeforeEach;  
**import** org.junit.jupiter.api.Test;  
  
**import** java.util.List;  
  
**import static** org.junit.jupiter.api.Assertions.\*;  
  
**class** AmmunitionServiceTest {  
 AmmunitionDao<Ammunition> **ammunitionDao**;  
 Helmet **helmet** = **new** Helmet(**"h1"**, 1, 1, 1, VisionLevel.***HALF\_FACE***);  
 ChestPlate **chestPlate** = **new** ChestPlate(**"cp2"**, 2, 2, 2, 2);  
 AmmunitionService **ammunitionService**;  
 @BeforeEach  
 **void** setUp() **throws** Exception {  
 AmmunitionDao<Ammunition> dao = **new** AmmunitionDao<>();  
 dao.create(**helmet**);  
 dao.create(**chestPlate**);  
 **ammunitionService** = **new** AmmunitionService(dao);  
 }  
  
 @AfterEach  
 **void** tearDown() {  
 **ammunitionService** = **null**;  
 **ammunitionDao** = **null**;  
 }  
  
 @Test  
 **void** findAll() **throws** Exception {  
 *assertEquals*(List.*of*(**helmet**, **chestPlate**), **ammunitionService**.findAll());  
 }  
  
 @Test  
 **void** findEntityById() **throws** Exception {  
 *assertEquals*(**chestPlate**,**ammunitionService**.findEntityById(1));  
 }  
  
 @Test  
 **void** deleteById() **throws** Exception {  
 **ammunitionService**.delete(0);  
 *assertEquals*(1, **ammunitionService**.findAll().size());  
 }  
  
 @Test  
 **void** deleteByEntity() **throws** Exception {  
 **ammunitionService**.delete(0);  
 *assertEquals*(1, **ammunitionService**.findAll().size());  
 }  
  
 @Test  
 **void** create() **throws** Exception {  
 **ammunitionService**.create(**new** Helmet(**"h2"**,1,1,1,VisionLevel.***HALF\_FACE***));  
 *assertEquals*(3, **ammunitionService**.findAll().size());  
 }  
  
 @Test  
 **void** computeTotalEquippedAmmunitionCost() **throws** Exception {  
 *assertEquals*(3, **ammunitionService**.computeTotalEquippedAmmunitionCost());  
 }  
  
 @Test  
 **void** getAmmunitionListSortedByWeight() **throws** Exception {  
 List<Ammunition> ammunitionListSortedByWeight = **ammunitionService**.getAmmunitionListSortedByWeight();  
 *assertEquals*(List.*of*(**helmet**, **chestPlate**),ammunitionListSortedByWeight);  
 }  
  
 @Test  
 **void** getAmmunitionListInCostRange() **throws** Exception {  
 List<Ammunition> ammunitionListInCostRange = **ammunitionService**.getAmmunitionListInCostRange(0, 1);  
 *assertEquals*(List.*of*(**helmet**),ammunitionListInCostRange);  
 }  
}

***Тест роботи executor-а команд ConsoleControl:***

**package** org.example.command;  
  
**import** org.junit.jupiter.api.AfterEach;  
**import** org.junit.jupiter.api.BeforeEach;  
**import** org.junit.jupiter.api.Test;  
  
**import static** org.junit.jupiter.api.Assertions.\*;  
  
**class** ConsoleControlTest{  
 ConsoleControl **consoleControl** = **new** ConsoleControl();  
  
 @BeforeEach  
 **void** setUp() {  
 }  
  
 @AfterEach  
 **void** tearDown() {  
 }  
  
 @Test  
 **void** setCommand() {  
 **consoleControl**.setCommand(0,**new** CraftAmmunitionCommand(**null**, **null**));  
 *assertTrue*(**consoleControl**.**commands**.get(0) **instanceof** CraftAmmunitionCommand);  
 }  
  
 @Test  
 **void** buttonWasPushed() {  
 }  
}

**Висновок**

На лабораторній роботі я покрив програму юніт тестами з coverage ≈ 67%.