

Практична робота 4

Принципи побудови та тестування об'єктно-орієнтованого коду

Система оцінювання

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Завдання на практичну роботу	3,8*
3.	<i>Здача звіту</i>	0,2
	Всього	5

* – діє бонусна система

- 0,3 бала **Визначення сутностей у проекті.** У межах даного завдання спроєктуйте схему для предметної області «Магазин з продажу техніки». У процесі роботи магазину потрібна буде інформація щодо продавців, адміністрації, товару, що продається, відвідувачів та ін. Кожну сутність на схемі представте у вигляді прямокутника, в якому описані її характеристики та можливі дії. Стрілками опишіть, як ці сутності пов'язані між собою.
- 0,3 бала **Створення сутностей у коді.** Перенесіть сутності зі схеми у програмний код. Розбийте їх за пакетами відносно напрямків роботи. Наприклад, пакет client, який міститиме всі класи, що характеризують відвідувачів магазину. Методи можна залишити пустими, проте називати їх відповідно до смислу дії, яку вони виконують. Пам'ятайте про принцип однієї відповідальності.
- 0,5 бала **Сценарії роботи магазину.** Опишіть сценарії взаємодії сутностей підчас роботи магазину, наприклад, виклик консультанта, виконання консультації щодо купівлі товару, оформлення покупки, оплату покупки тощо. Пам'ятайте про принцип підстановки Барбери Лісков – встановіть необхідні для здійснення кожного сценарію умови контракту. Перелічіть тестові випадки для кожного сценарію у вигляді таблиці:

Сценарій	Тестовий випадок	Необхідні вхідні дані	Очікуваний результат

- 0,6 бала **Створення архітектури додатку.** Спроєктуйте UML-діаграму, яка опише зв'язки між створеними класами. Спільні методи та поля винесіть в батьківські класи, а в породжених класах можна додавати власні методи або змінювати існуючі. Використайте в реалізації асоціативний зв'язок (агрегацію/композицію). Наприклад, коли список покупок включає окремі товари. Можливе вкладення одних класів у інші. Не забувайте поліморфний принцип відкритості/закритості та принцип розділення інтерфейсу.
- 0,5 бала **Реалізація відношень між класами / інтерфейсами.** Доповніть програмний код із завдання 2 таким чином, щоб він відповідав архітектурі завдання 4. Забезпечте поліморфізм підтипів за потреби. Пам'ятайте принцип інверсії залежностей.

6. 0,6 бала **Створення та взаємодія об'єктів.** Створіть торговий зал, додайте співробітників, товари та відвідувачів. Передбачте роботу адміністратора – пошук та надання консультанта. Виконайте в програмі консультацію та покупку товару. Згенеруйте javadoc-документацію для задачі, додайте анотації, що описують класи та методи, зокрема @param, @return, @author, @version. При використанні колекції зробіть посилання на інформацію (@see). Рекомендується додати обробники виключень та відповідну анотацію @throws.
7. 0,6 бала **Реалізація тестових випадків.** Створіть тестовий код, який виконує модульне тестування розробленого в завданні 5 коду відповідно до тестових випадків, представлених у завданні 3.
8. 0,4 бала (Параметризоване unit-тестування) Створіть клас Prime, об'єкти якого дозволяють виконувати знаходження простих чисел відповідно до лістингу:
- ```
package prime;
public class Prime {
 public boolean definePrime(int n) {
 prime = n;
 for(int i=2; i<prime/2; ++i) {
 if(prime%i == 0)
 return false; //число не є простим
 }
 return true; //число є простим
 }
 private int prime;
}
}
```
- Напишіть тестовий проект, що складається з колекції параметризованих тестів (@RunWith(Parameterized.class)) та дозволяє відповідно до переданого числа, визначити, просте воно чи ні (і відповідно, пройдено тест чи ні).