Лабораторна робота №1 (максимально - 10 балів)

Тема: Принципи програмування. DRY, KISS, SOLID, YAGNI та ін.

Мета роботи: навчитися дотримуватися принципів програмування та обґрунтовувати їх.

Завдання на лабораторну роботу

Завдання 0: Підготовка до виконання завдання

1. Створити окремий **публічний** репозиторій на GitHub. В цьому репозиторії будуть міститися всі виконані Вами завдання курсу “Конструювання програмного забезпечення”. Кожне завдання буде міститися в окремій директорії або гілці `lab-1`, `lab-2 ... lab-n`
2. Впевнитися, що Ви уважно прочитали пункт 1 і Ваш репозиторій точно **публічний** ;)
3. Склонувати створений репозиторій
4. Обрати **один з трьох** варіантів Завдання 1 🙂

Завдання 1 Виконати завдання з дотриманням відомих Вам принципів програмування. **Один** на вибір з 3 варіантів.

Варіант 1: Склад товарів

1. Запрограмуйте клас ***Money*** (об'єкт класу оперує однією валютою) для роботи з грошима. У класі мають бути передбачені: поле для зберігання цілої частини грошей (долари, євро, гривні тощо) і поле для зберігання копійок (центи, євроценти, копійки тощо). Реалізувати методи виведення суми на екран, задання значень частин.
2. Створити клас ***Product*** для роботи з продуктом або товаром. Реалізувати метод, який дозволяє зменшити ціну на задане число.
3. Реалізувати клас ***Warehouse***, який описує товари, що зберігаються на складі: найменування, одиниця виміру, ціна одиниці, кількість, дата останнього завозу, тощо.
4. Реалізувати клас ***Reporting*** для роботи зі звітністю. Реєстрація надходження товару (формування прибуткової накладної) і відвантаження (видаткова накладна). Звіт по інвентаризації (залишки на складі).

5. Для кожного з класів реалізувати необхідні методи і поля. Для класів передбачити реалізацію конструкторів та методів для встановлення та читання значень.
6. Ви також можете додавати власний функціонал для унаочнення принципів програмування. Приклади додаткового функціоналу:
    a. категорії для продуктів;
    b. конкретні дочірні класи валюти
    c. корзина для замовлень.

Варіант 2: Зоопарк

1. Створіть систему класів для обліку зоопарку. Ви можете створювати класи для різних видів і підвидів тварин; для вольєрів різних розмірів і типів; корму для тварин; працівників зоопарку.
2. Створіть класи інвентаризації, для виведення на екран інформації про наявних тварин, кількості співробітників тощо.

Варіант 3: Дія

1. Створіть систему класів для нової покращеної Дії. Ви можете створити класи для різних типів документів: паспорту, водійських прав, військового квитка.
2. Окремі документи можуть мати як спільну поведінку і риси (ім’я, фотографія, приховати інфу, показати інфу, згенерувати QR) так і індивідуальну (скопіювати ідентифікаційний код, зареєструвати авто).
3. Створіть також загальний клас-контейнер для документів, де можна буде міняти порядок документів, задавати тип показу і т.д.

Завдання 2: Написати код для тестування отриманої функціональності.

1. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
2. Достатньо буде просто вивести певну інформацію, щоб показати, що класи комунікують певним чином між собою.

Завдання 3: Опишіть особливості дотримання принципів програмування в Вашому коді

1. Додайте файл README.md в кореневу директорію цієї лабораторної роботи. В файлі README.md опишіть дотримання окремо кожного принципу програмування, який Вам відомо, і який можна продемонструвати Вашим кодом.
2. Опис можна залишати українською або (бажано) англійською мовами.
3. Ваш опис повинен містити посилання на відповідні файли і рядки коду.
4. Як залишати посилання на свої рядки коду можна глянути тутечки (для посилання на директорію) або тут (для посилання на окремі рядки).
5. Синтаксис .md файлів документації можна знайти туть або туть.
6. Для отримання максимальної оцінки Ви повинні продемонструвати мінімум **7** принципів. SOLID принципи рахуються окремо. Повний список принципів, які було розглянуто на лекції:

    a. DRY,

    b. KISS,

    c. SOLID (5 окремих принципів)

    d. YAGNI

    e. Composition Over Inheritance

    f. Program to Interfaces not Implementations

    g. Fail Fast

Завдання 4: UML діаграма

1. Підготувати діаграму створених у програмі класів та інтерфейсів за допомогою https://app.diagrams.net/. **Звертайте особливу увагу на стрілки між сутностями.** На лекції було розглянуто особливості створення UML діаграм для нашого предмету.

2. Експортувати створену діаграму у вигляді картинки або PDF та запушити експортований файл у кореневу директорію цієї лабораторної.