Основи програмування

Лабораторна робота №7

Тема: Шаблон «Стратегія»

Хід роботи:
1. Повторити теоретичні відомості
2. Провести рефакторинг свого коду з лабораторної роботи №6
   - використати шаблон «Стратегія»
   - дотримуватись принципів SOLID
3. Додати реалізацію ще одного алгоритму сортування на свій вибір
   - крім Array.sort()
4. Відповісти на контрольні запитання

Контрольні питання:

1. Що таке патерни (шаблони) проектування? Для чого вони потрібні?
   a. A design pattern provides **a general reusable solution for the common problems that occur in software design**. The pattern typically shows relationships and interactions between classes or objects. The idea is to speed up the development process by providing well-tested, proven development/design paradigms
2. Для чого потрібен шаблон «Стратегія»?
   a. Strategy is **a behavioral design pattern that turns a set of behaviors into objects and makes them interchangeable inside original context object**. The original object, called context, holds a reference to a strategy object and delegates it executing the behavior.
3. Пояснити смисл кожного з принципів проектування SOLID.
   a. SOLID is an acronym that stands for five key design principles: **single responsibility principle, open-closed principle, Liskov substitution principle, interface segregation principle, and dependency inversion principle**.
4. Для кожного з принципів SOLID навести фрагмент коду, який його порушує, пояснити сутність проблеми та запропонувати спосіб її вирішення.
   a. SOLID is an acronym for the five software design principles by Robert C. Martin. I highly recommend reading his book “Clean Architecture.” So here’s the list of principles:
   Single-responsibility principle (SRP)
   Open-closed principle (OCP)
   Liskov substitution principle (LSP)
   Interface-segregation Principle (ISP)
   Dependency-inversion principle (DIP)
5. Що означають терміни «компонент» та «залежність» (component, dependency).
   a. A dependence on a habit-forming substance such as a **drug or alcohol**; addiction. Dependency is defined as a state of needing something or someone. When you rely on coffee to get you through the day, this is an example of a caffeine dependency. ... A state of dependence; a refusal to exercise initiative.In UML, a dependency relationship is **a relationship in which one element, the client, uses or depends on another element, the supplier**. ... As the following figure illustrates, a dependency is displayed in the diagram editor as a dashed line with an open arrow that points from the client to the supplier.

6. В чому полягає принцип явного використання залежностей (Explicit Dependencies Principle)?

   The Explicit Dependencies Principle states: **Methods and classes should explicitly require (typically through method parameters or constructor parameters) any collaborating objects they need in order to function correctly**.

7. В чому полягають переваги і недоліки явного (explicit) і неявного (implicit) використання залежностей?
   a. **Explicit and implicit methods** are approaches used in numerical analysis for obtaining numerical approximations to the solutions of time-dependent ordinary and partial differential equations, as is required in computer simulations of physical processes. Explicit methods calculate the state of a system at a later time from the state of the system at the current time, while implicit methods find a solution by solving an equation involving both the current state of the system and the later one.
8. Пояснити смисл термінів «Зв'язність» (Coupling) та «Пов'язаність» (Cohesion).
   a. **" Coupling " describes the relationships between modules**, and " cohesion " describes the relationships within them. ... This means that in a good design, the elements within a module (or class) should have internal cohesion.