

Практична робота (самостійне опрацювання)

Тема 16. Об'єктно-орієнтоване програмування

Дисципліна: Основи програмної інженерії

Студент: Вівчар Вадим Вікторович

Група: АЛК-43

Тема 16. Питання для самостійного опрацювання

Об'єктно-орієнтоване програмування

1. Сутність об'єктно-орієнтованого програмування

Об'єктно-орієнтоване програмування (ООП) — це парадигма програмування, у межах якої програмна система розглядається як сукупність взаємодіючих об'єктів. Кожен об'єкт поєднує в собі дані (властивості) та методи (поведінку), що дозволяє ефективно моделювати реальні сутності та процеси предметної області.

На відміну від процедурного підходу, де акцент робиться на алгоритмах і послідовності дій, ООП концентрується на структурі системи та взаємодії її елементів. Це підвищує зрозумілість коду, полегшує супровід, сприяє повторному використанню компонентів і спрощує масштабування проєктів.

2. Основні поняття ООП

Клас — це шаблон (опис) майбутніх об'єктів, який визначає набір атрибутів і методів. **Об'єкт** — це конкретний екземпляр класу, що має власний стан (значення атрибутів) та може виконувати визначені методи. **Атрибути** відображають стан об'єкта, а **методи** — операції, що змінюють або використовують цей стан.

3. Основні принципи ООП

Інкапсуляція. Інкапсуляція означає об'єднання даних і методів у межах одного класу та приховування внутрішньої реалізації. Завдяки цьому забезпечується захист даних, зменшується кількість помилок, а також спрощується модернізація системи: змінювати реалізацію можна без впливу на зовнішній інтерфейс.

Наслідування. Наслідування — механізм, що дозволяє створювати нові класи на основі існуючих. Похідний клас може успадковувати властивості й методи базового класу, а також доповнювати або перевизначати їх. Це зменшує дублювання коду та формує логічну ієрархію.

Поліморфізм. Поліморфізм — здатність об'єктів різних класів по-різному реагувати на однакові виклики методів через спільний інтерфейс. Він дає змогу будувати гнучкі системи, де поведінка об'єкта визначається його фактичним типом під час виконання програми.

Абстракція. Абстракція полягає у виділенні ключових характеристик сутності та ігноруванні другорядних деталей. У програмуванні абстракція реалізується за допомогою абстрактних класів та інтерфейсів, які задають контракт поведінки для похідних класів.

4. Переваги об'єктно-орієнтованого програмування

- **Модульність** — програма складається з незалежних частин.
- **Повторне використання коду** — компоненти можна застосовувати повторно.
- **Масштабованість** — систему легко розширювати.
- **Підтримуваність** — код простіше читати та супроводжувати.

5. Недоліки ООП

- складніший поріг входу для початківців;
- ризик надмірного ускладнення системи;
- можливі більші витрати ресурсів.

6. Об'єктно-орієнтовані мови програмування

До мов, що підтримують ООП, належать C++, Java, C#, Python, PHP, Swift та інші. Вони широко застосовуються у створенні сучасних програмних продуктів.

7. Об'єктно-орієнтоване проектування

Об'єктно-орієнтоване проектування (OOD) — це процес побудови структури системи у вигляді класів, їхніх зв'язків та сценаріїв взаємодії. Для візуалізації використовують UML-діаграми, а для підвищення якості рішень — шаблони проектування та принципи SOLID.

8. Значення ООП у сучасній програмній інженерії

ООП є основою багатьох сучасних технологій і фреймворків та ефективно поєднується з Agile-методологіями і практиками DevOps.

Висновок

Об'єктно-орієнтоване програмування є потужним підходом до створення складних програмних систем. Воно забезпечує структурованість, гнучкість, повторне використання коду та зручність підтримки.