**Лабораторна робота №4**

Застосування патернів програмування

Завдання

1. Аналізуючи діаграми класів (Лабораторна робота №3), обґрунтувати

застосування підібраних для реалізації патернів.

2. Використовуючи лабораторні роботи із дисциплін професійної

підготовки, створити репозиторій, розмістити файли проектів. Надати

доступ викладачам на github / gitlab ресурсі.

Для створення діаграми класів молочної фабрики я визначили ключові об'єкти та їх функціональність, щоб забезпечити повне розуміння структури та взаємодії між різними частинами системи. Кожен клас і його функціональність вибрані на основі логічного поділу обов'язків та основних функцій, які необхідні для роботи молочної фабрики.

### **Опис класів та їх функціонал**

1. **МолочнаФабрика (DairyFactory)**
   * **Чому використаний:** Основний клас, що представляє всю фабрику і відповідає за загальне управління.
   * **Функціонал:**
     + **ПочатиВиробництво()** – запуск процесу виробництва.
     + **ЗавершитиВиробництво()** – завершення процесу виробництва.
     + **НабратиПрацівників(int кількість)** – збільшення кількості працівників.
     + **ВідправитиПродукцію()** – відправка продукції до клієнтів.
     + **ПровестиІнвентаризацію()** – перевірка наявності продукції та ресурсів.
2. **Продукція (Product)**
   * **Чому використаний:** Представляє продукцію, яку виробляє фабрика.
   * **Функціонал:**
     + **ВстановитиЦіну(float ціна)** – встановлення ціни на продукцію.
     + **ВстановитиТермінПридатності(DateTime термін)** – встановлення терміну придатності.
     + **ОтриматиІнформацію()** – отримання інформації про продукцію.
     + **ОбчислитиВартістьВиробництва()** – розрахунок вартості виробництва.
     + **ПеревіритиЯкість()** – перевірка якості продукції.
3. **Молоко (Milk)**
   * **Чому використаний:** Конкретний тип продукції, який виробляє фабрика.
   * **Функціонал:**
     + **ВимірятиЖирність()** – вимірювання жирності молока.
     + **ВстановитиОб'єм(float об'єм)** – встановлення об'єму молока.
     + **ПеревіритиСвіжість()** – перевірка свіжості молока.
     + **Пастеризувати()** – пастеризація молока.
     + **Запакувати()** – пакування молока.
4. **Сир (Cheese)**
   * **Чому використаний:** Інший тип продукції, яку виробляє фабрика.
   * **Функціонал:**
     + **ВизначитиТипСиру(string тип)** – встановлення типу сиру.
     + **ВстановитиВагу(float вага)** – встановлення ваги сиру.
     + **ПеревіритиЗрілість()** – перевірка зрілості сиру.
     + **Розфасувати()** – фасування сиру.
     + **Пакувати()** – пакування сиру.
5. **Йогурт (Yogurt)**
   * **Чому використаний:** Ще один тип продукції, яку виробляє фабрика.
   * **Функціонал:**
     + **ВстановитиСмак(string смак)** – встановлення смаку йогурту.
     + **ВстановитиВагу(float вага)** – встановлення ваги йогурту.
     + **ПеревіритиЯкість()** – перевірка якості йогурту.
     + **Охолодити()** – охолодження йогурту.
     + **Запакувати()** – пакування йогурту.
6. **Працівник (Employee)**
   * **Чому використаний:** Представляє працівників фабрики, які виконують різні завдання.
   * **Функціонал:**
     + **ВиконатиЗавдання(string завдання)** – виконання конкретного завдання.
     + **ОтриматиЗарплату()** – отримання зарплати.
     + **ПідвищитиКваліфікацію()** – підвищення кваліфікації працівника.
     + **НабратиНовихПрацівників(int кількість)** – набір нових працівників.
     + **ПеревіритиСтанОбладнання()** – перевірка стану обладнання.
7. **Клієнт (Client)**
   * **Чому використаний:** Представляє клієнтів, які замовляють продукцію.
   * **Функціонал:**
     + **ЗробитиЗамовлення(string продукція, int кількість)** – зробити замовлення.
     + **ОтриматиРахунок()** – отримання рахунку за замовлення.
     + **ОплатитиЗамовлення(float сума)** – оплата замовлення.
     + **ПеревіритиСтатусЗамовлення()** – перевірка статусу замовлення.
     + **Відгукнутись()** – залишити відгук.
8. **Замовлення (Order)**
   * **Чому використаний:** Представляє замовлення, які роблять клієнти.
   * **Функціонал:**
     + **ВстановитиСтанЗамовлення(string стан)** – встановлення стану замовлення.
     + **РозрахуватиЗагальнуВартість()** – розрахунок загальної вартості замовлення.
     + **ДодатиПродукцію(string продукція, int кількість)** – додати продукцію до замовлення.
     + **ВидалитиПродукцію(string продукція)** – видалити продукцію з замовлення.
     + **ОтриматиІнформаціюПроЗамовлення()** – отримання інформації про замовлення.
9. **Постачальник (Supplier)**
   * **Чому використаний:** Представляє постачальників сировини та інших матеріалів.
   * **Функціонал:**
     + **НадатиСировину(string сировина, int кількість)** – надання сировини.
     + **ОтриматиЗамовлення()** – отримання замовлення від фабрики.
     + **ВідправитиРахунок()** – відправка рахунку за сировину.
     + **ПеревіритиСтатусДоставки()** – перевірка статусу доставки.
     + **ОновитиКонтактнуІнформацію()** – оновлення контактної інформації.
10. **Транспорт (Transport)**
    * **Чому використаний:** Представляє транспортні засоби для доставки продукції.
    * **Функціонал:**
      + **ЗавантажитиПродукцію()** – завантаження продукції.
      + **РозвантажитиПродукцію()** – розвантаження продукції.
      + **ПеревіритиСтан()** – перевірка стану транспортного засобу.
      + **Заправити()** – заправка транспортного засобу.
      + **ВизначитиМаршрут()** – визначення маршруту для доставки.

### **Резюме**

Класи вибрані на основі їхньої функціональності та ролі в молочній фабриці. Кожен клас виконує конкретні завдання, що дозволяє структурувати систему ефективно та логічно. Взаємодія між класами забезпечує комплексне управління процесами виробництва, замовлення, постачання та доставки продукції.