1) Enum + state machine: rendelés állapotok (Order)

Cél: PHP 8.1 enum, állapotgép-szerű logika, szabályozott átmenetek, perzisztencia JSON-ba.

Feladat:

1. Hozzon létre egy OrderStatus enum-ot rendelésállapotokkal (pl. új, fizetve, csomagolva, kiszállítva, törölve).
2. Készítsen Order osztályt:
   - tárolja az állapotot enumként,
   - legyen „következő állapot” léptetés metódus (csak engedett átmenetekkel),
   - legyen „törlés” metódus (pl. kiszállítottat már ne engedje törölni).
3. Mentse a rendeléseket JSON fájlba (repository réteggel).
4. index.php:
   - új rendelés felvétele űrlappal,
   - listázás táblázatban,
   - gombok: „következő állapot”, „törlés”,
   - PRG használata.

---

2) Strategy minta: szállítási díj számítás több algoritmussal

Cél: Strategy interface + több implementáció, kontextus/service osztály, DI.

Feladat:

1. Definiáljon ShippingStrategy interface-t, ami kiszámolja a szállítási díjat.
2. Készítsen legalább két stratégiát:
   - fix díj,
   - távolság (és pl. súly) alapú díj.
3. Készítsen CheckoutService osztályt, ami a stratégiát konstruktorban kapja.
4. index.php:
   - űrlap: stratégia választás, súly, távolság, részösszeg,
   - eredmény: szállítás + végösszeg táblázatban.

---

3) Observer / Event dispatcher: NoteCreated esemény

Cél: események és listener-ek, lazán csatolt mellékhatások (log, statisztika), perzisztencia.

Feladat:

1. Készítsen egy minimális EventDispatcher-t, amire listener-ek feliratkozhatnak esemény típus alapján.
2. Készítsen NoteCreated eseményt és Note domain osztályt.
3. NoteService mentse a jegyzeteket JSON-ba, majd dobjon NoteCreated eseményt.
4. Regisztráljon legalább 2 listenert:
    - az egyik logoljon fájlba,
    - a másik vezessen számlálót (hányszor jött létre jegyzet).
5. index.php:
    - jegyzet felvétel,
    - jegyzetek listázása,
    - utolsó log sorok megjelenítése,
    - számláló megjelenítése.

---

4) Decorator minta: árkalkuláció kedvezménnyel + opcionális cache-sel

Cél: dekorátorok láncolása egy közös interface körül (kedvezmény, cache).

Feladat:

1. Definiáljon PriceProvider interface-t (termék id → ár).
2. Készítsen alap implementációt, ami JSON katalógusból olvas.
3. Készítsen DiscountDecorator dekorátort, ami százalékos kedvezményt alkalmaz.
4. Készítsen CacheDecorator dekorátort, ami fájlban cache-eli az árakat (hit/miss statisztikával).
5. index.php:
    - termék választás, kedvezmény megadása,
    - checkbox: cache be/ki,
    - eredmény táblázatban, cache stat megjelenítése.

5) Template Method + Factory: riport generálás CSV/HTML formátumban

Cél: Template Method algoritmusváz + Factory a megfelelő generátor kiválasztására + letöltés.

Feladat:

1. Készítsen ReportGenerator absztrakt osztályt Template Method mintával:
    - generate() fix folyamat,
    - a konkrét részek (header/row/footer) alosztályban.
2. Készítsen két generátort:
    - CSV,
    - HTML.
3. Készítsen ReportFactory-t, ami formátum alapján visszaadja a megfelelő generátort.
4. A hallgatók adatai JSON-ból jöjjenek (repository).
5. index.php:
    - hallgató hozzáadás űrlappal,
    - lista táblázatban,
    - linkek riport letöltéshez (download.php?format=csv/html).