

SZOFTVERTERVEZÉS ÉS -FEJLESZTÉS 2. – FÉLÉVES FELADAT

Készítsen egy órarend generáló rendszert, ami különféle tantárgyak kurzusait tudja kezelni. Lényeges, hogy a rendszer működése független legyen a konkrét tantárgytól; minden *IKurzus* interfészt megvalósító osztályt tudjon kezelni, ami az alábbiakat írja elő:

- *Kód*, *Típus* (*Online*, *Offline*), *Nap*, *ÓraKezdet* (óra:perc), *ÓraVég* (óra:perc), *Tantárgy* tulajdonságokat,
- egy *Ütközik*(*IKurzus-ok*) metódust, ami igazat ad vissza, amennyiben az átadott kurzusok közül valamelyikkel ütközik az adott kurzus (egy kurzus akkor ütközik egy másikkal, ha azonos napon vannak és az egyik kezdési és végzési időpontja között kezdődik vagy végződik a másik kurzus).

Készítsen egy absztrakt *Tantárgy* osztályt, ami egy láncolt listában tárolja az *IKurzus* példányokat. Az osztálynak legyen egy *Név*, egy *Kredit*, egy *Félév* csak olvasható és egy *Kreditek* tulajdonsága. (Például a Szoftvertervezés- és fejlesztés I. tantárgyhoz tartozik egy SF1_EA kódú előadás és több SF1_LA_0* kódú labor.)

Készítsen néhány osztályt (ehhez építsen ki egy ésszerű osztályhierarchiát), amelyek vagy az absztrakt osztályból származnak vagy megvalósítják az interfészt, és ezekből hozzon létre mintapéldányokat. Például:

- Gyakorlat (a típusa csak *Offline* lehet, az óra nem végződhet később, mint 22:00)
- Labor (gyakorlat leszármazottja, időtartama csak 90 vagy 180 perc lehet)
- Előadás (rendelkezik még egy *VizsgaTípus*(*Szóbeli*, *Írásbeli*) tulajdonsággal is, csak délelőtt kezdődhet)
- Vizsgakurzus (nem szerepelhet az órarendben)
- SzTF1 (név: Szoftvertervezés- és fejlesztés I, kredit: 6, félév: 1, kurzusok: egy előadás és hat labor kurzust)
- Analízis (név: Analízis I, kredit: 6, félév: 1, kurzusok: egy előadás és három gyakorlat kurzust)

Ezt követően készítse el a rendszert, amely az alábbi funkciókkal rendelkezik:

1. Az *IKurzus*-okat egy rendezett láncolt listában tárolja el, ahol a rendezés szempontja a kurzus kezdési időpontja (ebbe beleszámít a nap is, vagyis a Kedd 9:50-es kezdésű kurzust megelőzi a Hétfő 10:10-es). Ezt a listát az alábbi lehetőségeken keresztül lehessen feltölteni:
 - egy tantárgy listával (minden tantárgy minden kurzusa bekerül a belső listába),
 - egy tantárggyal (a tantárgy minden kurzusa bekerül a belső listába),
 - egy kurzus listával,
 - kurzusokkal egyesével.
2. A kurzus listában nem szerepelhet két azonos kódú és típusú kurzus, ilyen esetben egy kivétellel jelezze a problémát, ami tartalmazza az újonnan felveendő kurzust és azt amelyikkel egyezik.
3. Lehessen törölni a kurzus listából kód, típus, nap és az óra időtartama alapján (ha például nap szerint szeretnénk törölni, akkor az összes azon a napon lévő kurzus törlődik).
4. Legyen egy optimális órarend összeállító funkciója, amely visszalépéses kereséssel elkészíti az órarendet. Az órarend készítés során (alkosson saját életszerű szabályokat az osztályai alapján), például:
 - egyetlen vizsgakurzus sem szerepelhet az órarendben,
 - az előadásoknak mindenféleképpen szerepelnie kell az órarendben,
 - ha egy tantárgyhoz tartozik gyakorlat vagy labor akkor azokból legalább egynek szerepelnie kell az órarendben.

Amennyiben több megoldás is lehetséges, akkor azt a megoldást válassza, amelyben a legkevesebb napon lesznek az órák. Természetesen egyetlen kurzus sem ütközhet az órarendben.

Az órarendet egy *Órarend* példányban tárolja el, ami tárolja a kurzusokat és lehetőséget biztosít az órarend táblázatos és felsorolás szerű megjelenítésére, melyből, ha a táblázatos megjelenítést választjuk akkor az a konzolra és a fájlba is kiírásra kerül.

5. A rendszer az órarend összeállítás során egy eseményen keresztül már a keresés közben is jelezze egy órarend példány átadásával, hogyha talált egy, az addig legjobbnak gondoltnál optimálisabb összeállítást.
6. A kurzusok rendelkezzenek egy *IdőpontVáltozás* nevű eseménnyel. Amennyiben ez aktiválódik, akkor erről a generáló rendszer is értesüljön, és ennek megfelelően hajtsa újra végre az órarend generálását.

A feladat megoldása során tartsa be a tanult OOP alapelveket (egységbezárás, láthatóság, öröklés), kivételkezelés segítségével kezelje a felmerülő problémás eseteket. Ahol szükségesnek érzi, egészítse ki a megadott osztályhierarchiát, funkcionalitást, illetve használja a tanult technikákat (pl. eseménykezelés, generikus típusok, stb.). A tananyaghoz kapcsolódó adatszerkezeteket (lista, bináris keresőfa, stb.) ön valószínűsíti meg és működésükhöz implementálja a szükséges metódusokat (pl. Hozzáadás, Törlés, Keresés, Beszúrás).