

## OBJEKTUM ELVŰ ALKALMAZÁSOK FEJLESZTÉSE

### Dokumentáció a 4. házi feladathoz

**Név:** Soós Bálint  
**Neptun kód:** HDX9MU  
**Elérhetőség:** soba95@inf.elte.hu

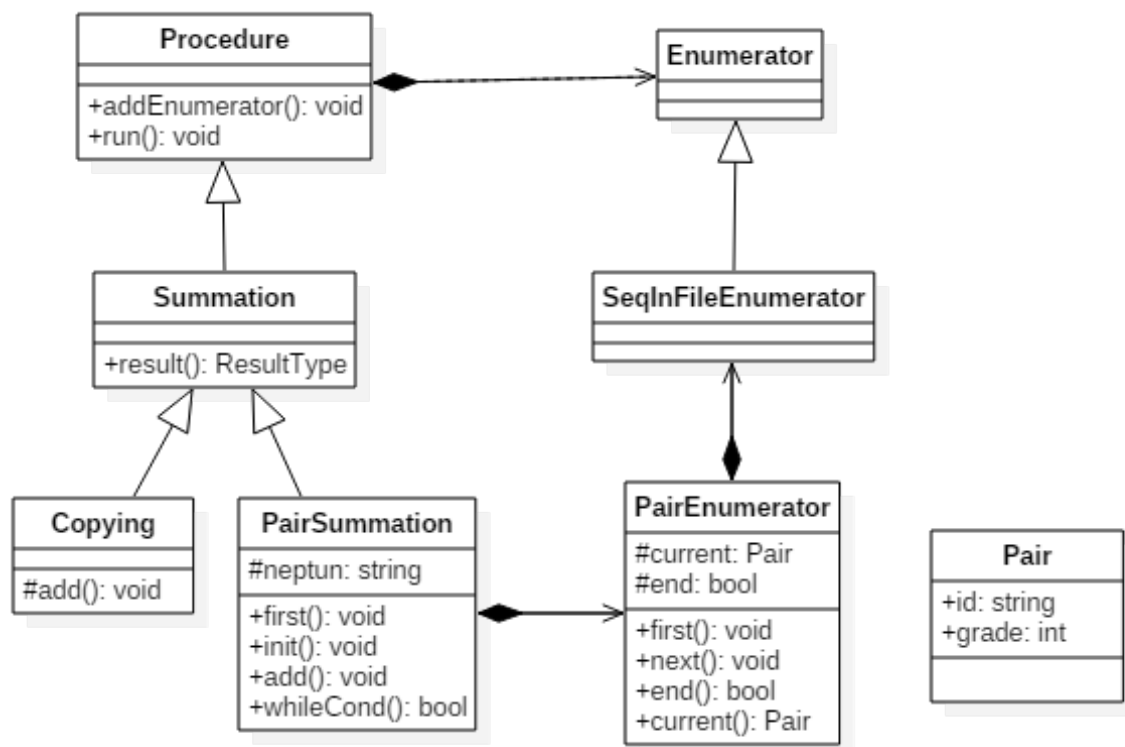
**Csoport:** 8.  
**Feladatszám:** 7.  
2015. december 9.

#### Feladatileírás

Egy szöveges állomány egy csoport hallgatóinak megajánlott gyakorlati jegyeit tartalmazza neptunkód-osztályzat alakú sorok formájában (az neptunkód 6 karakter hosszú, utána egy szóköz jön, azt követően pedig egy 0 és 5 közötti egész szám). Azok a hallgatók, akik sikerrel szerepeltek tehetséggondozási programozási versenyen, a megajánlott jegyüknél eggyel jobbat kapnak, pontosabban azoknak a hallgatóknak nő eggyel a gyakorlati jegye, akik a programozási versenyen 40 pontnál jobb eredményt értek el, de a megajánlott jegyük nem elégtelen vagy jeles. A verseny eredményeit egy másik szöveges állomány tartalmazza neptunkódpontszám alakú sorok formájában (az neptunkód 6 karakter hosszú, utána egy szóköz jön, azt követően pedig egy 0 és 100 közötti egész szám). Készítse el a csoport hallgatóinak végleges gyakorlati jegyeit tartalmazó szöveges állományt neptunkód-osztályzat alakú sorok formájában! Mindkét állomány neptunkód szerint növekedően rendezett: az gyakorlati jegyeket tartalmazó állomány neptunkód szerint szigorúan növekedően rendezett, de a második állomány a programozási verseny részeredményeit tartalmazza, nem az elért összpontszámot. Így ebben az állományban egy neptunkód, többször is szerepelhet egymás után, ezért egy-egy hallgató összpontszámát ki kell számolni. (Kizárólag a saját felsorolók Next() metódusaiban használhat elágazásokat!)

#### Programterv

A feladat megoldásához elképzelhetünk egy olyan adatabsztrakciót, ahol elég csak a Neptun kód, jegy párokon végigiterálnunk, és kiírni őket a kimeneti fájlba. Ehhez kell egy egyedi felsoroló, amely elemenként megadja az adott neptun kódhoz tartozó diák aktuális jegyét, és a versenyen elért pontszámát. Ha ez meghaladja a 40-et, és a jegye legalább 2-es, és legfeljebb 4-es, akkor eggyel növeljük a jegyét, így megkapjuk a végleges érdemjegyeket.



Az adatokat az in1.txt és in2.txt fájlokból olvassuk be, az egyikben a neptun-jegy, a másikban a neptun-pont párok vannak. A megoldáshoz származtattam egy PairEnumerator osztályt az Enumerator osztályból, amely minden neptun kódra meghívja a Summation osztályból származtatott PairSummation osztály run metódusát (átadva az aktuális neptun kódot). A PairEnumerator next metódusában ha a fenti feltétel teljesül, akkor megnöveljük az érdemjegyet. A fájl kiírásáért a Copying osztály felel, amit szintén a Summation osztályból származtattam.

### Tesztelési terv:

I) A feladat specifikációjára épülő (fekete doboz) tesztesetek:

- 1) pont < 40
- 2) pont = 40
- 3) pont > 40 és jegy = 1
- 4) pont > 40 és jegy > 1 és jegy < 4
- 5) pont > 40 és jegy > 1 és jegy = 4
- 6) pont > 40 és jegy > 1 és jegy > 4

II) A megoldó programra épülő (fehér doboz) tesztesetek:

A saját kódrészben egyetlen elágazás található az eredmény kiírásánál (ennek mindkét ágát befutották a fekete doboz tesztesetek), ezen kívül csak szekvencia szerkezetet látunk, ezért nincs szükség külön fehér doboz tesztre.