## Feladat:

Egy szöveges állományban neptunkód-osztályzat párokat tartalmazó sorokat helyeztünk el. (Az neptunkód 6 karakter hosszú, utána egy szóköz jön, azt követően pedig egy 0 és 5 közötti egész szám.) Az állomány neptunkód szerint növekedően rendezett (ugyanolyan neptunkódot tartalmazó sorból egymás után több is lehet). Igaz-e, hogy minden hallgatónak az átlaga legalább négyes? A választ a konzolablakba írjuk! (A kiíratáson kívül csak egyetlenegy üres „else” ágú elágazást használjon!)

input:

AAAAAA 3

AAAAAA 5

BBBBBB 2

BBBBBB 5

BBBBBB 5

output:

Eredmény: Igaz

## Programterv:

-  A feladat megoldása több lépésből áll: először olvasunk a fileból, majd azt több lépésben feldolgozzuk.

-  Az első részfeladathoz szükségünk van egy szekvenciális fájl felsoroló típusra, amely rendelkezésünkre áll (SeInFileEnumerator). A szekvenciális fájlból stringeket és számokat olvasunk be.

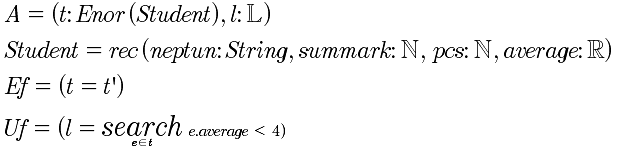
-  Létrehozunk egy új felsorolót ami a már kiátlagolt neptunkód- és számpárokat sorolja fel(Student). Ebben a felsorolóban használunk egy a Summation-ből örököltetett összegzést(StudentSummation).

-  A főprogramban a saját felsoroló elemire alkalmazunk egy saját, a LinSearch-ből   
örököltetett StudentDecide osztályt, amely lényegében meghatározza, hogy van-e négyesnél kisebb átlag.

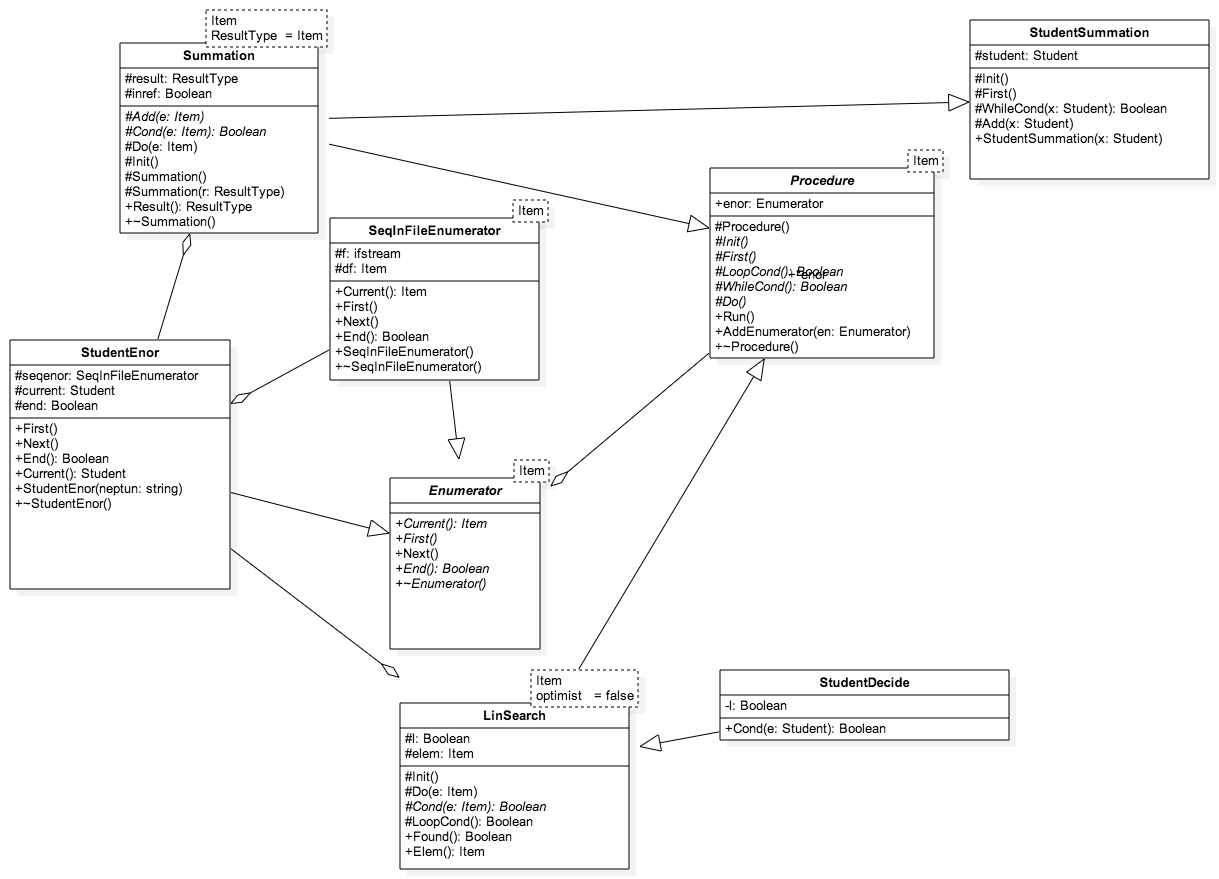
-  Az általános felsoroló/programozási tétel osztálygyűjteményből felhasználjuk a következő osztályokat: általános tevékenység (Procedure), összegzés (Summation), lineáris keresés (LinSearch), általános felsoroló (Enumerator), szekvenciális fájl felsoroló (SeqInFileEnumerator)

-  Az Enumerator osztályból származó StudentEnor is egy felsoroló amely a neveket és már a hozzájuk tartozó átlagokat sorolja fel. A First, Next, End, Current műveletek felül vannak definiálva

-  A Summation-ből származó StudentSummation elemtípusa a Student, eredménytípusa a Student. Meghatározza az adott neptunkódhoz tartozó átlagot.

-  A LinSearch-ből származó StudentDecide az StudentEnum által felsorolt elemek közül keresi meg hogy van-e négyesnél rosszabb átlagú Student. A Cond-ot felül kell definiálni (,ha talál 4-nél kisebb átlagot akkor térjen vissza true-val.

Feladat specifikáció:

**A program osztálydiagramja a mellékelt oaf4diag.****png.**

*Preview:*

**Megvalósítás:**

A megvalósításnál kódújrafelhasználást alkalmazunk.

Újrafelhasznált fejállományok: enumerator.hpp, seqinfileenumerator.hpp, procedure.hpp, summation.hpp, linsearch.hpp

Létrehozott állományok: student.h, student.cpp, studentenor.h, studentenor.cpp, studentdecide.h, studentsummation.h, studentsummation.cpp

**Tesztelési terv:**

*Feltételezzük hogy a bemeneti fájl helyesen van kitöltve.*

Fekete doboz tesztesetek:

1. Üres fájl. (üres fájl esetén a program igazat ad vissza)
2. Minden neptun kódhoz egyetlen jegy tartozik:
   1. Csak négyesnél rosszabb jegy szerepel.
   2. Minden jegy legalább négyes.
3. Többször is szerepelnek azonos neptun kódok:
   1. Van négyesnél rosszabb átlag.
   2. Minden átlag rosszabb négyesnél.
   3. Minden átlag legalább négyes.

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.