# Feladat

A turisták látogatása bevételt hoz egy városnak, de kis mértékben rontja is a város állapotát. Egy város, ami jó állapotban van, vonzza a turistákat. Egy rossz állapotú város taszítja az odalátogatni készülőket. Egy turista látogatása átlagosan 100.000 Ft bevételt hoz a városnak. Ha a város bevétele egy évben meghaladja az egy milliárd forintot, az egy milliárdon felüli részt a város javítására és szépítésére fordítják, hogy több látogató érkezzen a következő évben. A város állapota 1 és 100 pont között mozog (1 alá és 100 fölé sose megy, mert az állam elkölti a fölösleget és besegít, ha már nagyon vészes a helyzet). 1 és 33 közt számít lepusztultnak, 34 és 67 között átlagosnak és 67 fölött jó állapotúnak. Minden évben egy milliárd forint bevétel fölött minden húszmillió forint hoz egy pont állapotjavulást a városnak.

A turisták 3 fajtába sorolhatók: a japánok rendet raknak maguk után, így ők nem rontják a város állapotát. A modern országokból érkező turisták kevésbé ügyelnek a környezetükre: 100-asával rontanak egy-egy pontot a város állapotán. A harmadik csoportba sorolható turisták azon országok képviselői, ahol a szemetelés kulturális szokásnak tekinthető, ők 50-esével rontanak egy-egy pontot a város állapotán.

Ha a város jó állapotban van, abban az évben 20%-kal több japánt és 30%-kal több modernt vonz, mint ahány tervezte, hogy ellátogat oda. Átlagos állapotban 10%-kal több modernt és 10%-kal több harmadik típusú turistát vonz. Lepusztult állapot esetén a japánok egyáltalán nem jönnek, a többiek pedig annyian, amennyien tervezték.

Adjuk meg, hogy a fájlban jelölt évek letelte után milyen a város állapota! Körönként mutassuk meg az érkezett turisták számát (hány tervezett és hány jött) kategóriák szerint, az éves bevételt és a város felújítás előtti állapotát (szám és kategória)!

A program egy szövegfájlból olvassa be az adatokat! Az első sorban a város kezdeti állapota szerepel. A második sor jelöli a szimulált évek számát. A következő sorok tartalmazzák, hogy az egyes években hány turista tervezte, hogy eljön a városba: minden sor 3 darabszámot tartalmaz (japánok, modernek, többiek). A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

50

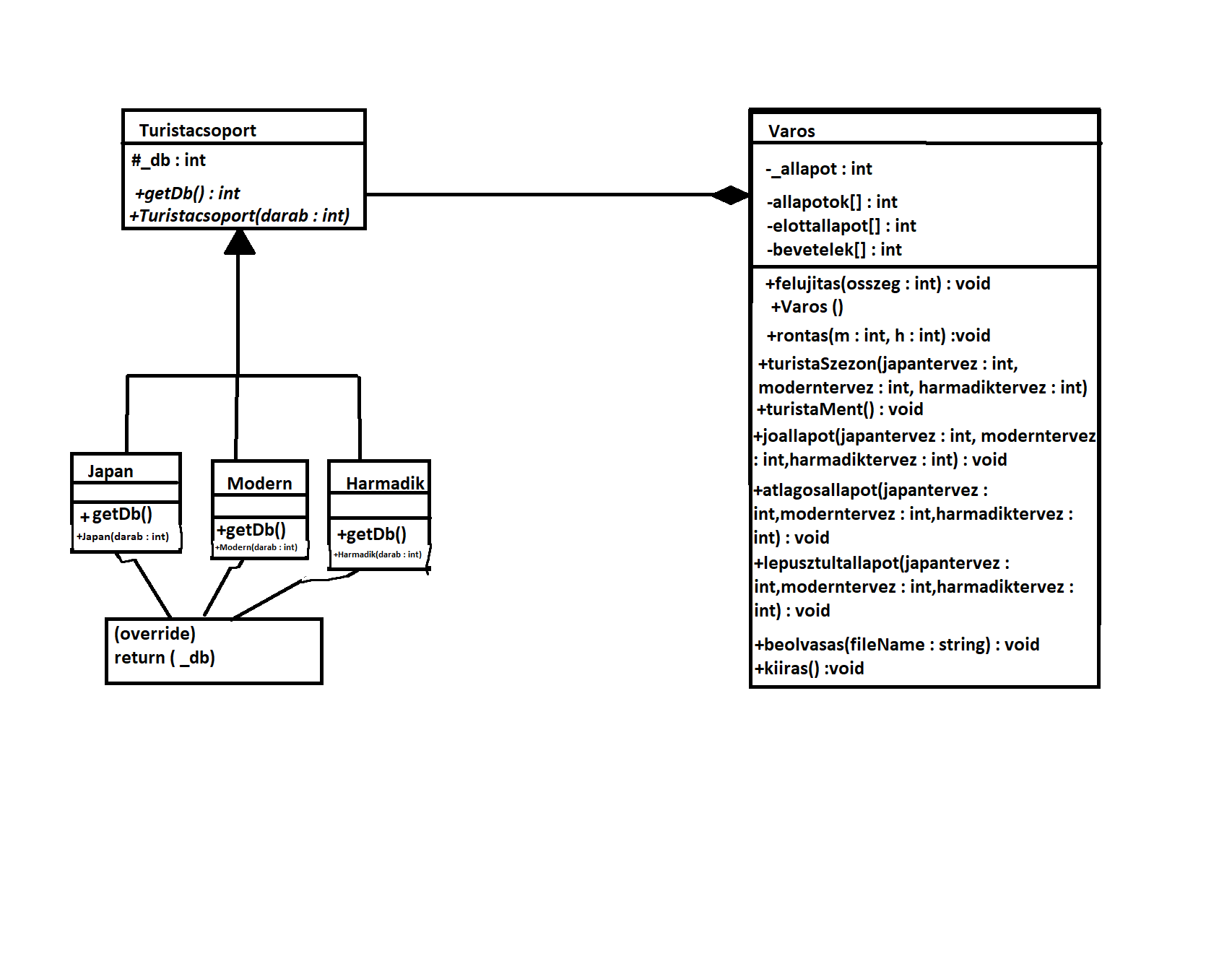
3

1000 4000 6000

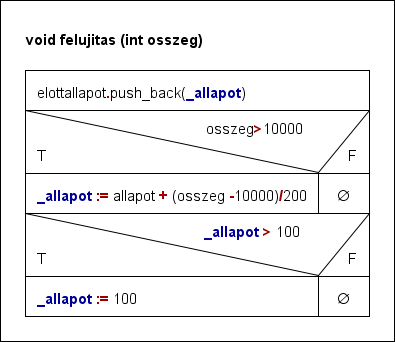
2000 3000 8000

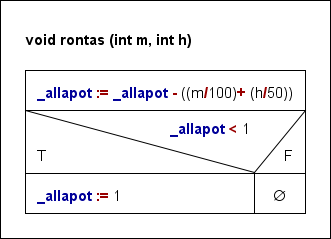
6500 5000 3000

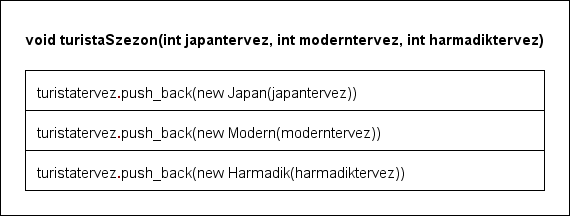
# UML Diagram (terv)

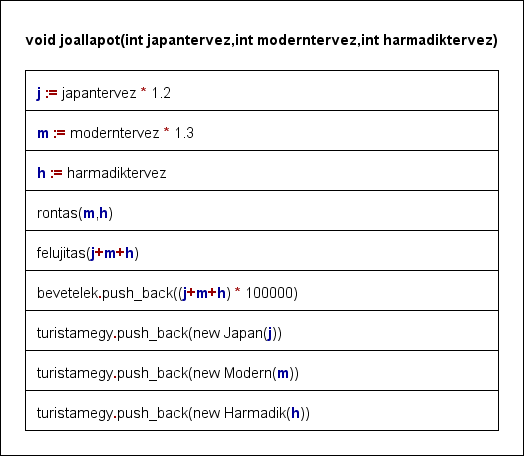


A diagram során használt függvények stuktogrammjai:



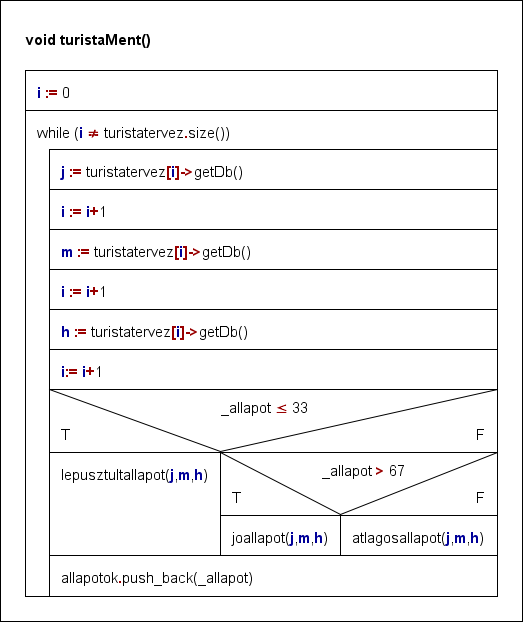


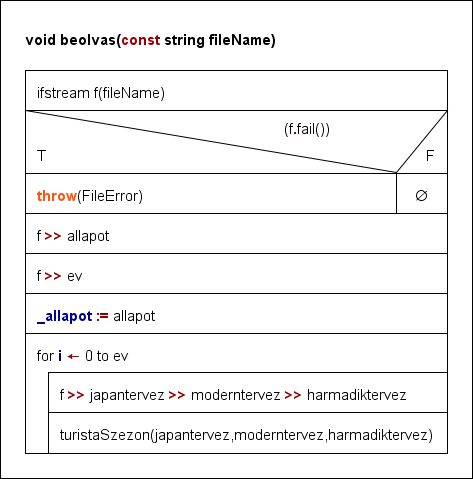












Ezeken kívül pedig még van egy void kiiras() függvényem, ami csak a megfelelő formában kiírja az adatainkat a képernyőre.

# Elemzés

A program tartalmaz egy Turistacsoport osztályt, aminek egy darab protected adattagja van, egy konstruktora, illetve egy virtuális metódusa. Ebből az osztályból származik le a Japan, Modern és Harmadik osztály, amelyeknek a származtatásból kifolyólag egy \_db adattagjuk van, ennek a lekérdező metódusa, illetve mindegyiknek egy konstruktor, amelyek az eredeti (ősosztálybeli) konstruktort implementálják.

A program következő osztálya, a Varos osztály, amely privát adattagként tartalmaz egy állapot egész számot, két Turistacsoport\* gyűjteményt (egyet a tervezett számra, egyet pedig a valós számra), illetve három számokat tároló gyűjteményt, amelyekben az egyes évek alatt kiszámolt adatokat tárolhatom el.

A felujitas metódus vár egy összeg egész számot paraméterül, amely valójában (egyszerűsítések okán) csak a turisták száma. Kezdetben elmenti az eredeti állapotot egy gyűjteménybe, hogy tudjuk később, a felújítás előtt milyen volt az állapot. Ezután a feladatban meghatározott számítási forma alapján javítjuk a város állapotát, ha erre alkalmas. Ha túllépjük a maximális értéket a város állapotával, akkor ez utóbbit a maximális értékre visszaállítjuk.

A rontas metódus a feladatban meghatározott számolási formával csökkenti az állapotot, illetve ha a minimumnál kevesebb lenne az állapot, akkor a minimumra állítja azt.

A turistaSzezon metódus elmenti a három paraméterként kapott számot abba a gyűjteménybe, amely a tervezett számát vezeti a túristáknak.

A turistaMent metódus a már elraktározott tervezett turista gyűjteményünkön halad végig hármasával és a város állapota alapján meghatározza és elmenti egy külön gyűjteménybe, hogy valójában hányan mentek a városba az adott évben, majd elmenti az új állapotát a városnak, amelyet az állapot meghatározás utáni három függvény valamelyike fog megváltoztatni (lepusztultallapot,joallapot,atlagosallapot). Ebben a metódusban egy speciális összegzés tételt használunk, amely 3 feltétel alapján rendszerezi külön az adott elemeket.

A joallapot, atlagosallapot, lepusztultallapot metódusok, a feladatban meghatározott módszerek alapján változtatják meg a tervezett turisták számát az igazi turisták számára, majd mentik el ezeket a megfelelő gyűjteménybe. Ezek a metódusok hívják meg a rontas és a felujitas függvényeket, továbbá a bevetelek gyüjteménybe is ezek a metódusok teszik bele a valóban érkezett turisták száma alapján a bevételét a városnak.

A beolvas metódus a nevéhez híven csak a fájl beolvasásáért felel. Kap paraméterként egy fájlnevet, amelyet megpróbál megnyitni, ha nem sikerül neki hibát dob, ha sikerül akkor pedig feltételezvén, hogy a fájl jól van megadva (ahogyan azt a feladat írta, hogy feltételezhetjük), beolvas egy állapotot, egy évszámot, majd évenként meghívja a turistaSzezon függvényt úgy, hogy minden évre 3 adattagot olvas be az adott év sorából és ezeket adja oda a turistaSzezon függvénynek.

A main-ben ezeknek a függvényeknek a válogatott meghívása történik, először létrehozunk egy Varos mutató város objektumot. Ebbe megpróbálunk beolvasni, ha nem sikerül akkor pedig errort írunk ki a képernyőre. Ezután meghívjuk a turistaMent függvényt a város objektumunkra, majd a kiiras metódust, ezután pedig visszaad a függvény 0-át, jelezvén, hogy a programunk hibamentesen futott le.