Tóth Márton	3. beadandó	2020. május 02.
FF66GJ	*6. feladat*	
ff66gj@inf.elte.hu		
21. csoport		

3. beadandó

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 10 év 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.)*

Elemzés

A feladat önálló objektumai a különböző „típusú” turisták (*Tourist*), akik három csoportba sorolhatók: Japánok (*Japanese*), Modernek (*Modern*) és Szemetelők (*Waster*).

A turisták egy vektorban vannak tárolva, így objektumonként az évente menni tervezett és ténylegesen elment turisták száma van tárolva. Megjelenésükkel 100 000 Ft bevételt garantálnak a városnak, illetve látogatásuk az alábbiak alapján befolyásolja a város állapotát:

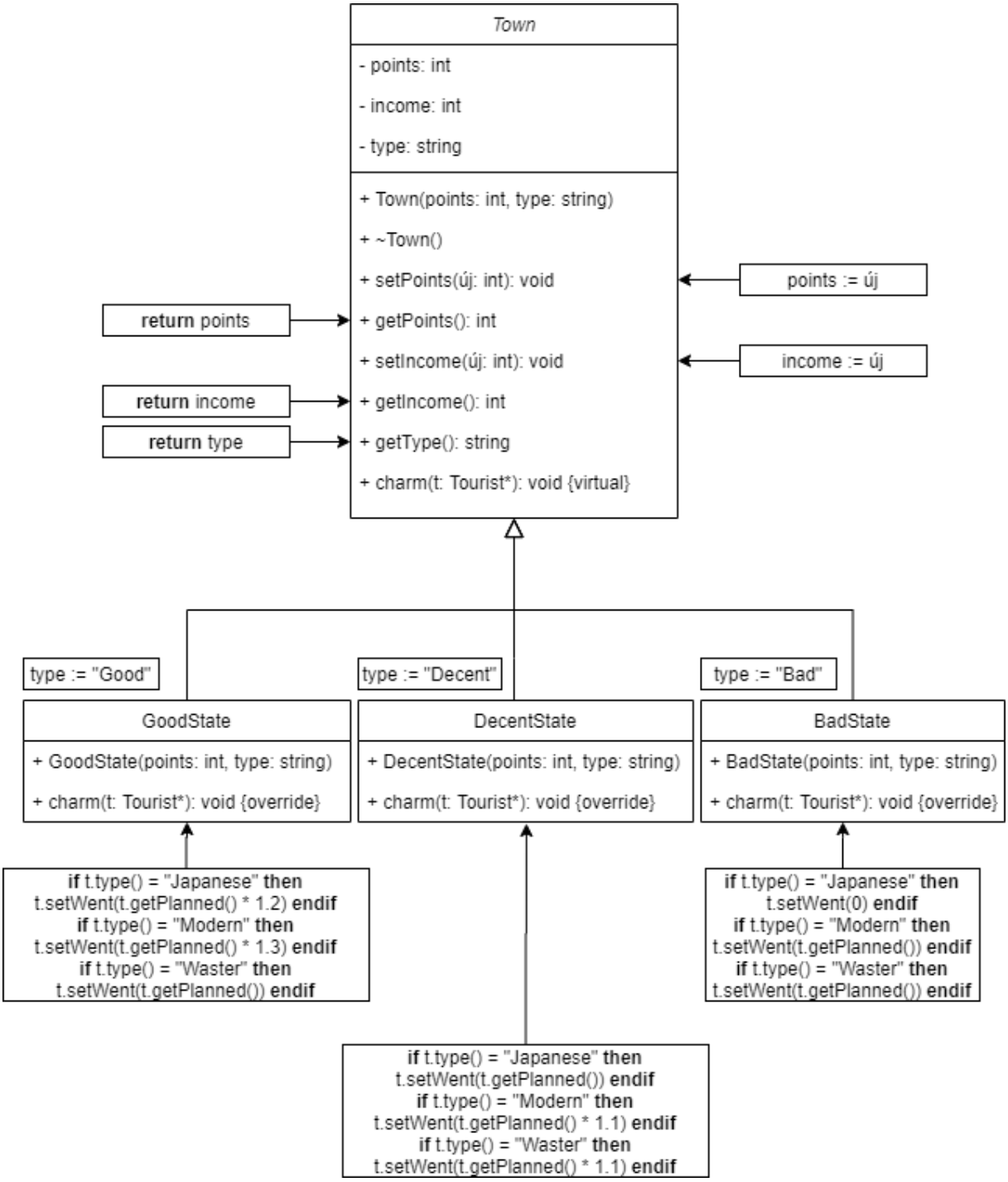
Japanese	-
Modern	Minden 100. turista után -1 pont
Waster	Minden 50. turista után -1 pont

A város (*Town*) aktuális (pontszám alapján eldöntött) állapota pedig hatással van a turisták tervezés utáni tényleges megjelenésére. 3 állapotra osztható a város: jó (*good*), átlagos (*decent*) és rossz (*bad*). Az állapottól függően a ténylegesen megjelent turisták száma így a következőképp alakul:

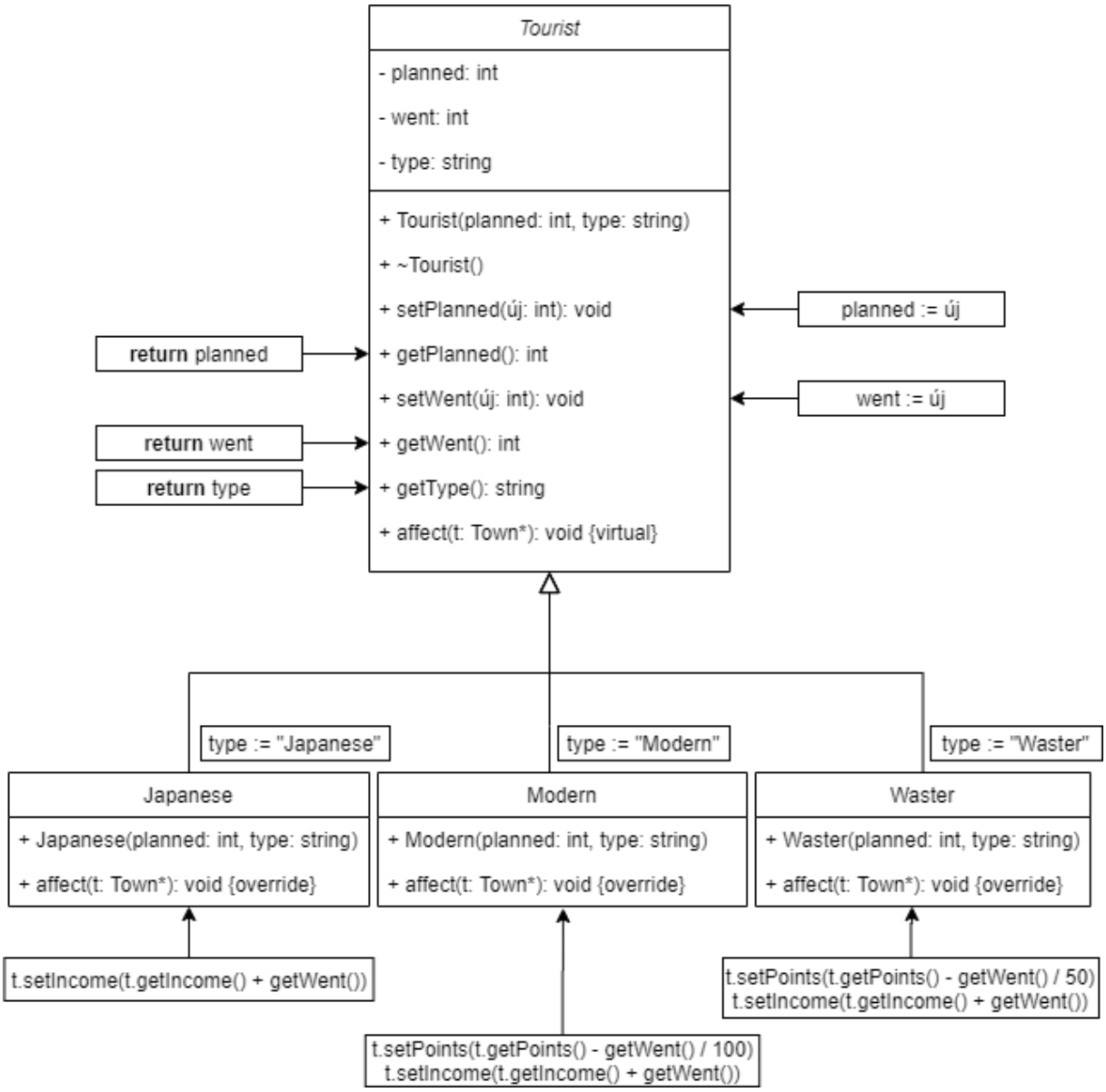
Town stance	Japanese	Modern	Waster
Good	tervezett * 1.2	tervezett * 1.3	tervezett
Decent	tervezett	tervezett * 1.1	tervezett * 1.1
Bad	0	tervezett	tervezett

Terv

Város (Town) objektum



Turista (Tourist) objektum



A kisebb memóriaigény érdekében a Town objektum nincs évenkénti vektortagokra bontva, hanem a következő évre iterálva felülírja a korábbi önmagát az új értékekkel.

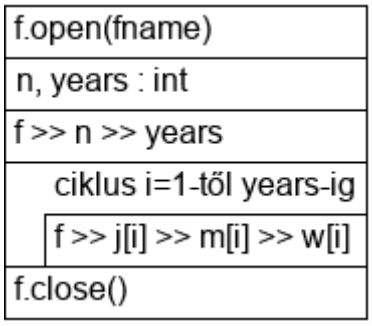
A: (city: Town, tourists: Tourist^n, tentype: string, tenpoints: int)

Ef: (tourists = tourists$_0$)

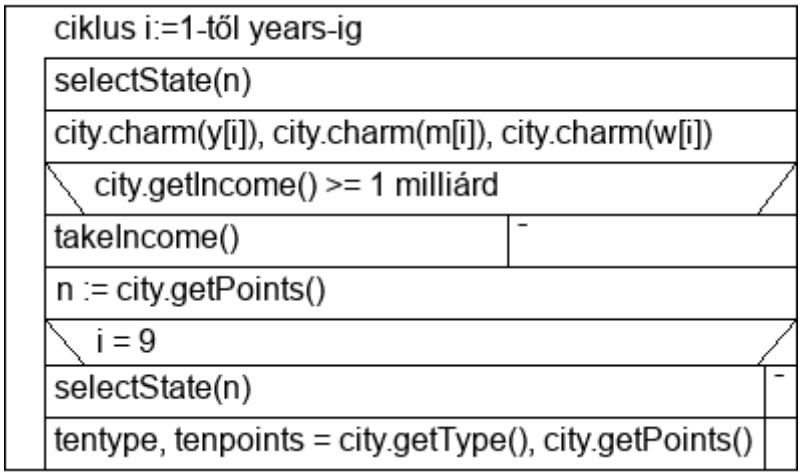
Uf: (years = |tourists| $\wedge$ **if** years<10 **then** tentype, tenpoints = city$_{years}$.type(), city$_{years}$.points() **else** tentype, tenpoints = city$_{10}$.type(), city$_{10}$.points()

Az objektumokra 6 különféle függvény hivatkozik, a *SelectState(), addPoints(), readFile(f,fname), process(), tenYears(),* és a *remove()*.

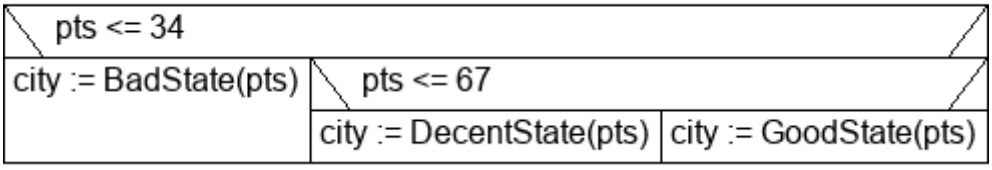
ReadFile(ifstream, string) függvény



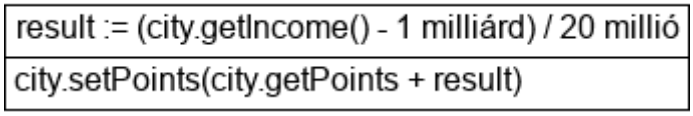
Process() függvény



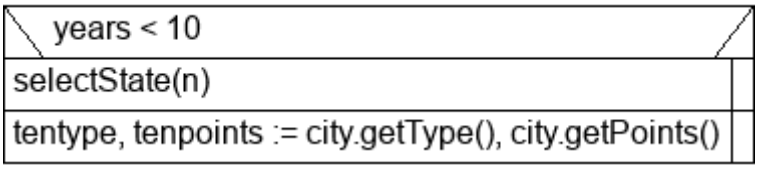
SelectState(pts) függvény



AddPoints() függvény



TenYears() függvény



Remove() függvény

Eltávolítja a city-hez rendelt objektumot, illetve a 3 turistavektorba tett objektumokat, majd kiüríti a vektorokat.

Tesztelési terv

1. Üres fájl beolvasása esetén a beolvasásra használt változók értéke megfelelő marad
2. A beolvasott pontszám és évszám megfelelő
3. Az objektumokon lefutott műveletek után kapott pontszám és évszám megfelelő
4. A „BadState” *charm* függvénye megfelelően működik a különböző turistatípusokra hivatkozva
5. A „DecentState” *charm* függvénye megfelelően működik a különböző turistatípusokra hivatkozva
6. A „GoodState” *charm* függvénye megfelelően működik a különböző turistatípusokra hivatkozva
7. A végleges (10. évi) eredmény (tenpoints, tentype) értékei megfelelőek
8. Jól választja ki a program a város kezdőállapotát a kezdőpontból
9. A későbbi állapotok a megváltozott pontszámokhoz megfelelően változnak
10. 1 és 100 közé sorolja az ezen az értékhatáron kívül megadott kezdőpontokat is a program
11. Beolvassa és működőképesen feldolgozza azokat a bemeneti fájlokat is, amelyek több mint 10 évet tartalmaznak
12. Több mint 10 évet tartalmazó bemeneti fájl esetén is rendes eredményt ír ki a tenpoints, tentype változó esetén