HÁZI FELADAT

Programozás alapjai 2.
Tóth Gábor
F041OM
2022. április 2.

1.1. Feladat

Tervezze meg a Meseországi Villamos Művek (MVM) nyilvántartási rendszerének egyszerűsített objektummodelljét, majd valósítsa azt meg! A rendszerrel minimum a következő műveleteket kívánjuk elvégezni:

- ügyfél adatinak felvétele
- szolgáltatási szerződés kötése
- szolgáltatási díj előírása (számlázás)
- szolgáltatási díj befizetése
- egyenleg lekérdezése
- fogyasztás bejelentése

A rendszer lehet bővebb funkcionalitású, ezért nagyon fontos, hogy jól határozza meg az objektumokat és azok felelősségét.

1.2. Feladatspecifikáció

A feladat egy nyilvántartási rendszer készítése, amely alapvető ehhez szükséges funkiókat tud ellátni. A rendszerbe fel lehet venni új ügyfelet, az adataival, lehet új szerződést kötni, azaz felvenni a rendszerbe (szerződés kitöltése), számlát előállítani, szolgáltatási díjat befizetni, egy ügyfél egyenlegét lekérdezni, illetve fogyasztást bejelenteni.

A program a beolvasott adatokat, az aktuálisan mentett adatokat szöveges fájl(okban) tárolja, ezeket kilépéskor lehet menteni, illetve elvetni a módosításokat, illetve a program futása közben is lehet menteni, de ezt a program nem teszi meg magától. A program indulásakor autómatikusan beolvassa a fájlokat, ha nem léteznek, majd mentéskor létrehozza a számára szükséges fájlokat.

Az aktuálisan tárolt adatokat lehet szerkeszteni is, ezeket a szöveges fájlban tárolt mentésben is felül fogja írni a program, így természetesen a tárolt adat törlésére is van lehetőség.

A futás közben bekért adatokat mindig egyértelműen jelzi, hogy mit és milyen formában vár a felhasználótól, ezeket mindig enterrel lehet véglegesíteni.

2.1. Ponotosított feladatspecifikáció

Az MVM nyilvántartási rendszere, amely külön tárolja az Ügyfeleket és külön a hozzájuk tartozó szerződéseket. Az osztályok az alábbi explicit függvényekkel rendelkeznek:

- létrehozás
- megszüntetés
- másolás
- értékadás
- adatok lekérdezése és beállítása

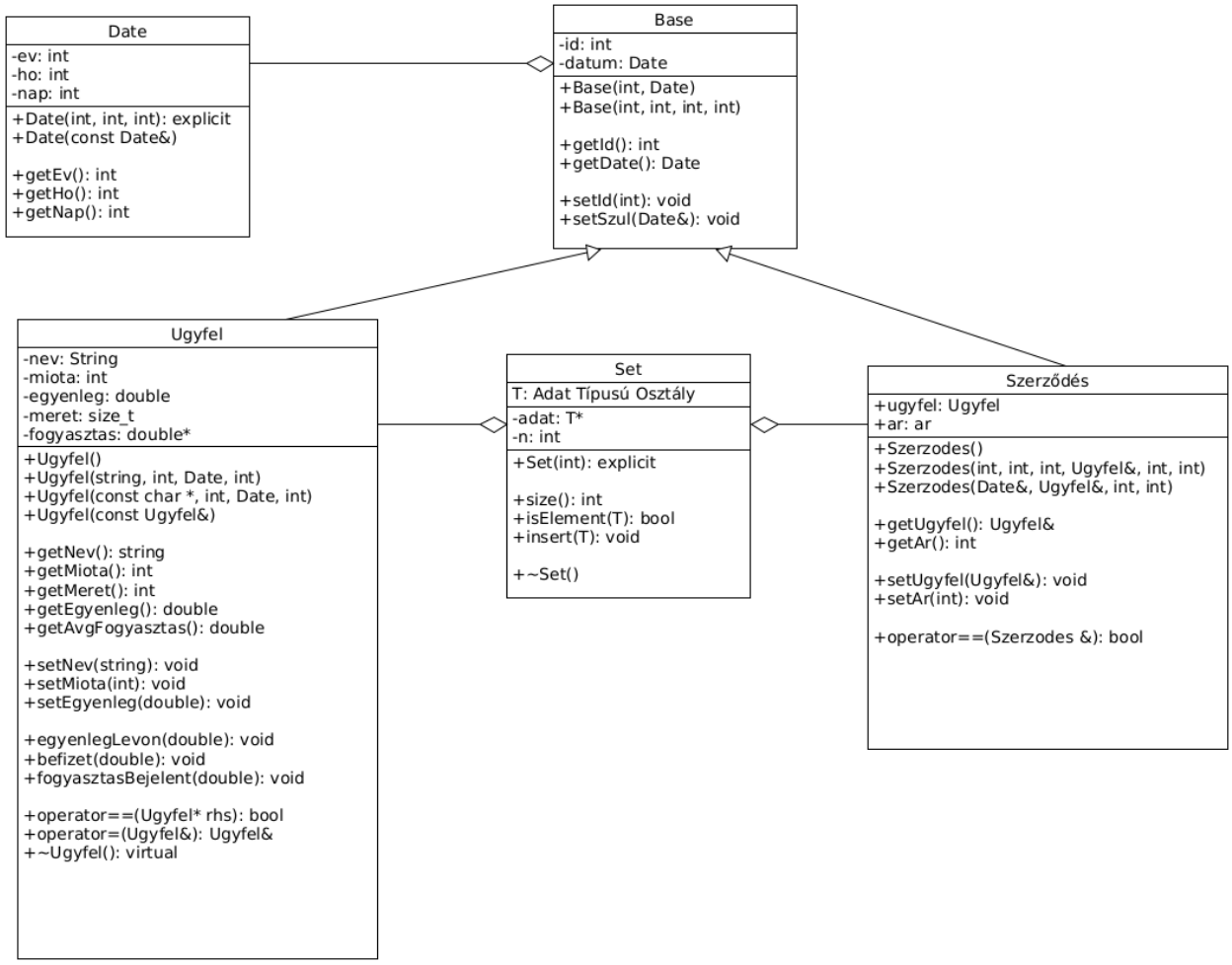
Több adat tárolásához ugyan tömb amely ami a következő tagfüggvényekkel rendelkezik:

- létrehozás
- megszüntetés
- értékadás
- indexelés
- hibás index esetén kivétel dobás
- adatok lekérdezése és beállítása

A teszteléséhez egy olyan programot készítek, ami különböző módokon létrehozott tömbökkel és objektumokat a standard inputról is beolvasott adatok alapján hoz létre és végez műveletet. A tesztadatok között hibás indexelés is elő fog fordulni.

2.2. Objektum terv

A feladat megvalósításához 5 osztályt használok, egyet dátumhoz, egy alaposztályt örökléshez, egy Ügyfél egy Szerződés osztály az ügyfelek és a szerződések tárolására, valamint egy generikus tömb osztály a több Ügyfél és szerződés tárolására.



2.3. Algoritmusok

A tesztprogram előre megírt a programmal fordított teszteseteket ellenőriz, ezzel minden funkcionalitását leelenőrizve. A fájlbeolvasásnál a filevége jelig olvas, külön szöveges fájl-ból az ügyfeleket és a szerződéseket, majd ezeket eltárolja, hibás fájl vagy adat esetén pedig kivételt dob.

3. Dokumentáció

3.1 Osztályok

Base
Alaposztály, az Ügyfél és a Szerződés osztály örököl tőle. Getter és Setter függvényei vannak, a konstruktorok is mindenhova alapértelmezett nulla értéket állítanak be.

Date
Konstruktora paraméter nélkül is hívható, ekkor minden érték nulla lesz, illetve nem megadott paraméterek (év, hónap, nap) mindig nullára inicializálódnak, még ha az dátumként nem is értelmezhető.
A kivonás operátor kiszámolja két dátum közti eltelt napok számát, a szökőnapokat is beleszámolva, ezt a szökönapokSzama() függvény biztosítja, ami megadja, az időszámítás eleje óta eltelt szökőnapok számát. A destruktora az alapértelmezett.
Van insertere és kiíró operátora, az inserterben nem lehet megadni 1900 január elseje előtti dátumot, mivel, születési dátomokhoz, és szerződések keltezésére használjuk, illetve ugyan ezért az aktuális rendszeren a mai dátum utánit sem lehet beadni. Ehhez a ctime könyvtárat használja a program. Ha

rossz, vagy nem értelmezhető adatot adbe a felhasználó, az az adatot addig kéri be újra, amíg megfelelő adatot nem kap.

String
A string osztály, a 8. laboron elkészített osztály, pár új operátorral. Az új operátorok, az egyenlőségvizsgáló operátorok String és másik String között, valamint String is karaktertömb (const char*) között. Illetve egy értékadó operátor, ami karaktertömbből készít Stringet.

Ügyfél
Az alaposztálytól (Base) örököl, azonosítót és (születési) dátumot (Date). 4 konstruktora van, az alapértelmezett minden értéket 0-ra állít, illetve üresen hagy. Illetve a kezdés és az egyenleg paraméter opcionális, ha nincs megadva 0-ra lesz inicializálva. A másoló konstruktor, a Srting értékadasó operátorát hívja meg, illetve a fogyasztás dinamikus double tömböt másolja.
Getter fügvények visszadják a kívánt adattagot, a getAvgFogyasztas(), az átagos fogyasztást (ami a tömbben van minden adat átlaga) kiszámolja, és visszadja.
A befizet() függvény, a paraméterként kapott double értéket hozzáadja, az egyenleghez, ezzel az ügyfél egyenlegét növelve.
A fogyasztasBejelent() egy új tagot ad a fogyasztás tömbbhöz.
A destruktor a fogyasztás tömböt felszabadítja.

Az egyenlőség operátor a fogyasztás tömb elemeit nem hasonlítja össze. Illetve a tömb méretét sem, csak hogy mióta ügyfél. A tömb mérete, és hogy mióta ügyfél megegyezik, kivéve néha rövid időre változik meg.

Az osztream operátor, csak a std::cout standard kimenetre íráshoz használatos mivel szövegfile-ba formázás nélkül csak adatokat írunk, a kiíró operátor pedig szöveges leírást ad a kiírt adatokhoz.

A beolvasó operátor a String és a Date beolvasó operátorait használja, és azok hibakezelésére támaszkodik, ezért fontos a név megadása után az új sor, vagy a lezáró nulla, ugyanis a String beolvasása csak ezekre áll le. Illetve az azonosító számszerűségét vizsgálja.

Szerződés
Az alaposztálytól (Base) örököl, azonosítót és (keltezési) dátumot (Date). Az ügyfélnél csak az azonosítót tárolja, de nem ellenőrzi, hogy van-e olyan ügyfél, csak akkor ha másik függvény használja az ügyfél azonosítóját. Az egyenlőség operátor mind a négy adattagjának egyezését vizsgálja.

Az inserter oprátor, formázottan ír ki, ezért fájlba íráshoz nem azt használja a program. A beolvasó operátor a Date hibakezelésére támaszkodik, illetve egész számok egész mivoltát ellenőrzik, és addig kérnek be adatot, amíg ez nem teljesül.

Set
A nyolcadik laboron elkészített Set osztály, kisebb módosításokkal. Egy új függbény a lookup(), ami visszadja, a paraméterként kapott azonosítójú elemet, feltételezve, hogy van olyan, mivel csak Ügyfeleket és Szerződéseket tárolunk ez teljesül. Ha a tárolt adatnak nincs getId() függvénye, ez nem fog működni.

Egyéb függvények
szamlaz(), egy szerződést két dátumot (mettől meddig), és az ügyfelek Set-jét (tömbjét) kap. Kiszámolja, a két dátum között hány hónap telt el (hányszor 30 nap), és a havidíjjal megszorozva, levonja az adott ügyfélnek, levonja, az egyenlegéből.

befizet() ugyan az mint az Ügyfél befizet tagfüggvénye, csak kell neki még egy Ugyfel& paraméter, akinek az egyenlegéhez hozzáad.

egyenlegLekerdez() visszaadja a paraméterként kapott ügyfél egyenlegét, ugyan az, mint a Ugyfel::getEgyenleg(), csak globális.

fileKiir() paraméterként kapott Ügyfél és Szerződés Set-et, ír ki, az „ugyfelek.txt” és a „szerződések.txt”-be. Az „ügyfelek.txt” minden sora egy ügyfelet tartalmaz, a fájl végén nincs üres sor. Minden sor egy Ügyfél azonosítóját, a Nevét, majd szóköz nélkül, az egyenlegét, illetve a születési dátumát szóközzel elválasztva tartalmazza rendre. A „szerződések.txt”, a szintén minden sorban, egy szerződést, tartalmaz, az adatok rendre a következők, szóközzel elválasztva: szerződés azonosítója, ügyfél azonosítója, a megállapított havidíj, és az szerződés keltjének dátuma.

3.2 Tesztelés

A program 15 tesztesetet futtat le, ezzel tesztelve az osztálykat és tagfüggvényeiket. Ezek kikapcsolhatók a main.cpp 9. sorában lévő makró nem definiálásával. A tesztek és tesztesetek nevük tartalmazzák, hogy mit tesztel az adott teszteset. Az első kettő a Date osztáyt teszteli. Majd az Ugyfél 5 tesztje következik. Aztán a szerződés osztályának és a számlázás globális függvénynek van 1 teljes tesztje. Aztán a következő 5 teszt a 8. laborról számarzó Set osztályhoz tartozó tesztesetek, kis módosításassal átírva, hogy az Ugyfél és a Szerződsé osztályt tesztelje. Az utolsó teszt pedig a fájlból olvasást teszteli.

3.3 Menü

A menü kikapcsolható a main.cpp 8. sorában lévő makró nem definiálásval.
A menü opciói a következők:

- Új elem felvétele: [uj]
  - Ügyfél adatinak felvétele: [u]
  - Szolgáltatási szerződés kötése: [sz]
- Tárolt adatok kiírása: [ki]
  - szerződések: [sz]
  - ügyfelek: [u]
- szolgáltatási díj előírása: [szamla]
- szolgáltatási díj befizetése: [bef]
- Ügyfél egyenleg lekérdezése: [el]
- Fogyasztás bejelentése: [bf]
- Kilépés: [q]/[exit]

A szögletes zárójelben van a beírandó parancs, a parancsot enterrel (új sor, vagy lezátó nulla) kell véglegesíteni, ismeretlen parancs esetén új parancsra vár. Ahol azonosítót vár, ott ellenőrzi a beolvasás sikerességét. Egyébként pedig az osztályok hibakezelésére hagyatkozik.

Kilépéskor menti a fileba az aktuálisan tárolt adatokat, a fileKiir() függvénnyel.