

Feladat:**Mobil szolgáltató**

Egy mobilszolgáltatónál egy egyedi nyilvántartó programmal szeretnék kezelni az ügyfeleket. Az ügyfeleknek van neve és címe, valamint telefonszáma, ami egyben az egyedi azonosítjuk is. A szolgáltató jelenleg három csomagot biztosít ügyfeleinek: Alap, MobiNet és SMSMax, de később több csomag is lehet. Minden csomaghoz más percdj és SMS díj tartozik, valamint a számítás módszere is eltérő lehet. A MobiNet csomag esetén pl. az is megadható, hogy hány SMS-t küldhet az ügyfél ingyen. A program egy fájlból olvassa be az ügyfelek adatait és választott díjcsomagot. Egy másik fájlból pedig az adott hónapban küldött SMS darabszámot és a lebeszélte percekét. A program írja ki, hogy az egyes ügyfelek mennyit fizetnek a forgalom alapján.

A megoldáshoz **ne** használjon STL tárolót

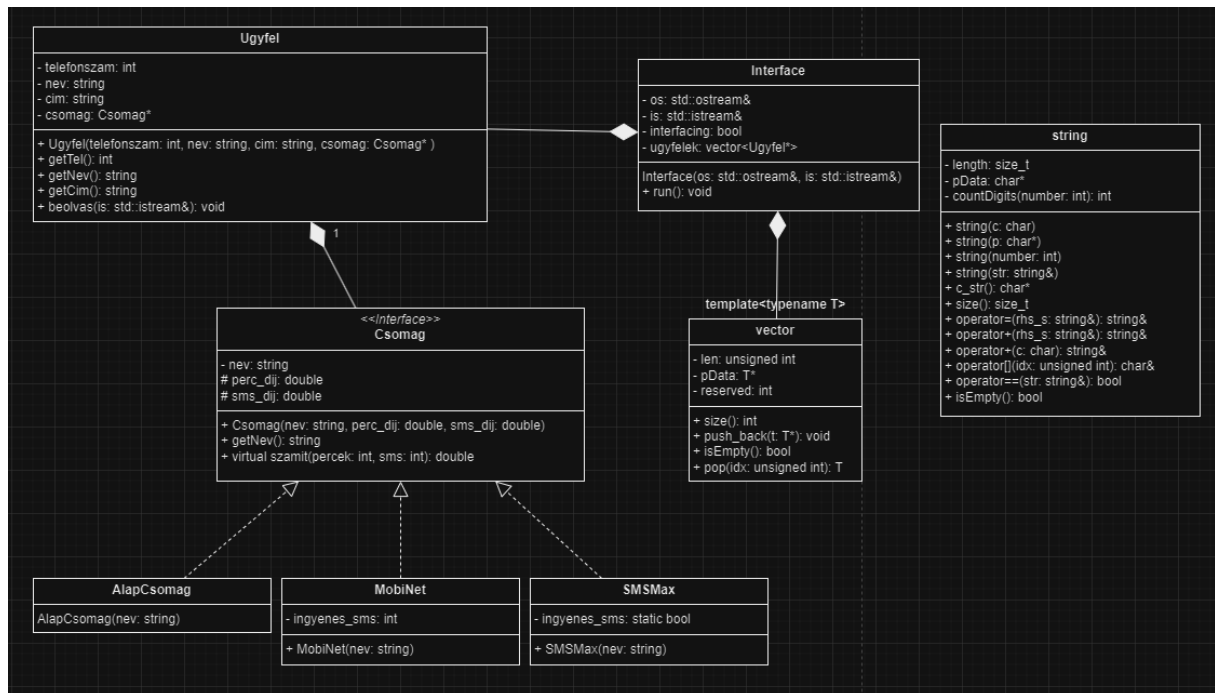
A program főmenüje, amelyhez a program minden művelet után visszatér a következőképpen opciókat tartalmazza:

0. Kilépés a programból  
A program futása befejeződik, amennyiben nincs fájlba írva, az állapota elveszik.
1. Ügyfél felvétele  
Az adatok megadása után az ügyfél felvételre kerül.
2. Ügyfelek listázása  
Kilistázza az összes ügyfelet.
3. Ügyfél törlése  
Az ügyfél pontos adatait kell megadni, hogy pontosan be lehessen azonosítani a törlendő ügyfelet.
4. Ügyfelek fájlba írása  
Meg kell adni a fájl nevét, és a program oda menti az ügyfelek adatait. Ha a fájl nem létezik, akkor a program létrehozza, különben felül írja a fájl tartalmát.
5. Ügyfelek betöltése fájlból  
Az ügyfeleket betölti a megadott file-ból. A jelenlegi állapot felülíródik.
6. Számlázás  
Meg kell adni, hogy melyik fájl tartalmazza az ügyfelek e havi lebeszélte perkeit, illetve smseinek számát. A program a beolvasott fájl alapján kiszámolja, hogy egyes felhasználó mennyit fizet. A program megkérdezi, hogy ezt a végeredményt szeretnénk-e fájlba írni. Ha nem akkor csak hagyjuk üresen a mezőt. A végeredmény a konzolon mindenképp megjelenik.

A program .txt típusú állományokat kezel, amelyben egy soron belül az adattagok tabulátorral vannak elválasztva. Például egy ügyfél adatai tabulátorral vannak elválasztva, és minden ügyfél új sorban kezdődik.

# Terv

## Terv az osztályokról



A string osztály használatát a diagramon nem jelöltem, mert minden osztály használja (a vector-t kivéve).

A vector osztály egy generikus tároló, amelyet a diagramon azzal jelöltem, hogy T típusú pontereket tárol, illetve felé írtam a template-t. Nincs feltüntetve minden tagfüggvénye.

Az Interface osztály valósítja meg a kommunikációt a felhasználóval, kimenetét a kezdetben megadott ostream-re helyezi, bemenete a kezdetben megadott istream. A belső működéshez szükséges függvényeket (egyes menüpontok megnyitása, input kezelése) a diagramon nem tüntettem fel. A fájlkezelés is itt történik. Az interface példányosítása után, run függvényének meghívásával az interface elindul és fut, amíg a felhasználó a kilépés opciót nem választja.

Az AlapCsomag az absztrakt Csomag osztályból származik, extra funkcióval nem rendelkezik. A MobiNet szintén a Csomagból származik, megadja hány smst lehet ingyenesen küldeni. Az ingyenes smsek felett viszont már drágább az sms küldés darabonként. Az SMSMax (szintén Csomag leszármazott) szerződése szerint az sms csak időszakosan ingyenes. Ezt a programban futásidőben állíthatja az operátor, az interfacen keresztül. Minden csomagnak van neve, amely a csomag azonosításában játszik szerepet, de futásidőben nem változik. A perc és sms díjat protected-re tettem, hogy a származott osztályok kalkulációban egyszerűbben tudják használni, esetleg módosítani, ha a kalkuláció úgy igényli.

Ügyfél osztály az ügyfelek adatait tartja számon. A rendszerben minden telefonszám +36 kezdetű, amelyet az ügyfél telefonszám mezője nem tárol, csak a telefonszám ezt követő részét. Minden ügyfélhez egyetlen csomag tartozhat.