

# Dokumentáció

## A feladat

Van egy országunk több mobilszolgáltatóval, számos előfizetővel, akik a különböző szolgáltatóknál különböző előfizetéseket használva végeznek telefonhívásokat, üzeneteket küldenek, interneteznek, stb...

A feladat az előfizetők tevékenységeinek nyomonkövetése és a szolgáltatók ebből származó bevételeinek kiírása

## Tervezés

A fenti feladat alapján a következő objektumelvű szerkezet alakítható ki:

Az ország (**Country**) tekinthető a rendszer fő objektumának. Van neve, hozzá kapcsolódik az összes szolgáltató (**ServiceProvider**), előfizető (**Subscriber**). Az előbbiekből új példányt is innen lehet létrehozni. Egy ország nyilvántartja a szolgáltatók adatait.

Egy használat (**Usage**) egy adatainak esemény eltárolása, lehet hívás(**Call**), internethasználat(**InternetUse**), üzenet(**Message**). Az internethasználatnak kilobyteban mért mennyisége, a hívásnak hossza percben, az üzenetnek pedig karakterben. Egy használat eltárolja megtörténtének időpontját dátum-idő (**DateTime**) formátumban és tudja magáról, hogy mennyibe kerül. Egy előfizetés lehet előfizetési(**ContractSubscription**) és feltöltőkártyás(**PrePaySubscription**), de mindenképpen rendelkezik egy számmal, egy előfizetőhöz és szolgáltatóhoz tartozik. Maga az előfizetés absztrakt, mivel a fizetés folyamata nem valósítható meg egyértelműen. Az előfizetés minden használatot(**Usage**) eltárol, a későbbi a szolgáltató felé irányuló jelentés elkészítéséhez.

Az előfizetőnek van neve, rendelkezik akár több előfizetéssel(**Subscription**), tud telefonálni, internetezni, sms-t küldeni, pénzt feltölteni vagy számláját kifizetni.

A szolgáltatónak(**ServiceProvider**) is van neve több előfizetés tartozik hozzá, vannak tarifái, bevétele és nyilvántartja előfizetési adatait.

## Megvalósítás

Az osztályok privát adattagokat alkalmaznak (leszármazottal rendelkezők protected-et). Ezeket külső osztályokból csak getterekkel és setterekkel érhetjük el. Listákhoz elemeket az "add"-dal kezdődő metódusokkal adhatunk.

Az országban hozhatunk létre szolgáltatókat, előfizetéseket, ( **addServiceProvider()**, **addSubscriber()** ), egy szolgáltatónál pedig előfizetőnél pedig lehet létrehozni előfizetéseket ( **addSubscription()** ), amik egyúttal a "megnevezett" ügyfél előfizetéseinek listájába is bekerülnek.(Megj.: az ügyfél **addSubscriber()** metódusát nem szabad használni mert hibát okozhat a pontos nyilvántartásban, ezt a metódust csak az adatbevitel elve miatt szükséges volt megvalósítani)

Egy ország **toString()** függvénye összegyűjti az egyes előfizetések **toString()**-jeit, amik pedig az egyes tevékenységek **toString()**-jeit foglalják egybe. Ezzel nyomon követi az összes szolgáltató bevételeit előfizetésekre és azon belül használatokra bontva vmint az előfizetés tulajdonosát is rögzítve.

Amennyiben egy előfizető telefonál, levelezik, használja az internetet, automatikusan létrejön a használt előfizetés listájában egy használat (**Usage**) példány dinamikusan a megfelelő leszármazottal (**Call**,

**Message, InternetUse**) példányosítva, majd kiszámolván a saját költségét a jellemzőkön(*length*, *minutes*, *dataAmount*) és a tarifákon(**messageRate**, **callRate**, **dataRate**) alapuló egyedi függvények segítségével, elvégzi az előfizetés a **pay()** metódusával a számlázást. A folyamat különböző típusú előfizetéseknél kisebb eltéréseket is mutat:

- számlás előfizetés esetén a számlát(**score**) növeli a számolt értékkel(**cost()**) és a szolgáltató bevételeit (**income**) nem növeli(azt majd az előfizető hajtja végre a számlatartozás befizetésénél) és létrehozza a használatpéldányt.
- feltöltőkártyás előfizetés esetén egy ideiglenes tesztpéldánnyal megvizsgálja hogy van-e elég egyenleg(**balance**) a szolgáltatás teljesítéséhez
  - amennyiben igen létrehoz egy végleges példányt a használatok listájában, növeli a szolgáltató bevételeit a költséggel (**raiseIncome()**), és leveszi ugyanezen összeget az egyenlegről
  - amennyiben viszont nincs a példányt nem véglegesíti és hibaüzenetet ír ki a konzolra a sikertelen az előfizetés adataival(**Subscriber.name**, **Subscription.number**)

Az előfizető más tevékenységet is végezhet:

- Feltöltőkártyás fizetését feltöltheti pénzzel a **refill()** metódus segítségével, a pénz összegét meghatározhatja
- Számlás előfizetés esetében azonban az egyszerűség kedvéért csak a teljes adósságot törlesztheti a **payScore()** metódus segítségével. Ekkor a szolgáltató bevételeit(**income**) is megnöveli ezen összeggel.

