

Programowanie Obiektowe i C++**Telefony w mojej głowie, ver. 1.1****Czas pisania: 180 min.**

Zaprojektuj mini system bilingowy rozliczający połączenia dla nowego operatora sieci komórkowej.

Operator umożliwia na razie tylko połączenia typu: głosowego, sms, dane, a w ofercie dostępne są następujące rodzaje kart sim:

- *Voice* - umożliwia wykonywanie połączeń głosowych oraz wysyłanie sms;
- *Smartphone* - połączenia głosowe i sms oraz dane;
- *Internet* - mobilny dostęp do sieci Internet, oprócz tego można wysłać smsa, nie da się zadzwonić, czyli wykonać połączenia głosowego

Do każdej karty sim przypisany jest dokładnie jeden, unikatowy numer telefonu. Do karty sim można dokupić pakiet(y). Do każdego klienta operatora przypisane jest dokładnie jedno konto. W ramach konta klient może mieć wykupionych wiele kart sim, np. dla członków swojej rodziny. Z konta nie da się zadzwonić, wysłać sms, itd. Do konta można dokupić pakiet, z którego mogą korzystać karty sim, które są przypisane do konta.

Klienci mogą zakupić następujące pakiety:

- *NumerNaKoncie*, bez ograniczeń i bez salda, tylko połączenia głosowe w ramach numerów zakupionych na danym koncie,
- *Rodzinny*, do wykorzystania przez wszystkie karty sim zarejestrowane w ramach konta na połączenia głosowe. Pakiet ma 2 salda: liczba sekund oraz liczba MB do wykorzystania,
- *IndywidualnySekundy*, ma 1 saldo: liczba sekund do wykorzystania,
- *IndywidualnyMB*, ma 1 saldo: liczba MB do wykorzystania.

W pakietach nie ma wzmianki o smsach. Oznacza to, że wszystkie smsy rozliczane są poza pakietami.

Dla karty sim rodzaju *Voice* użytkownik nie może kupić żadnych pakietów. Dla karty rodzaju *Smartphone* można kupić maksymalnie dwa pakiety: jeden na połączenia głosowe i jeden na połączenia typu dane (nie można dwu takiego samego typu). Dla karty sim typu *Internet* można dokupić tylko pakiet *IndywidualnyMB*.

Do konta użytkownik może dokupić maksymalnie jeden pakiet *NumerNaKoncie* i jeden *Rodzinny*.

System bilingowy, rozliczając kartę sim, musi mieć dostęp do bieżącego stanu wszystkich dostępnych pakietów dla danej karty sim (saldo pakietów). W momencie wykonania połączenia, system musi określić, do którego pakietu dane połączenie zostało przydzielone (zgodnie z priorytetami opisanymi poniżej) i w związku z tym zmniejszyć bieżącą wartość salda tego pakietu. Do tego celu powinna zostać wykorzystana wirtualna metoda *rozliczPołączenie* z kart sim.

Dla kart sim określone są priorytety przypisania połączenia (głosowe oraz dane) do pakietu, przy czym połączenie jest możliwe tylko wtedy, gdy dany pakiet obsługuje taki typ połączeń i odpowiednie saldo danego pakietu jest dodatnie w momencie rozpoczynania połączenia:

Dla kart sim typu *Smartphone* priorytety określone są następująco:

1. pakiet *NumerNaKoncie* - jeśli połączenie jest do jednego z numerów przypisanego do konta, pakiet jest bez limitu,
2. pakiet *Rodzinny*,
3. pakiet *IndywidualnySekundy* lub *IndywidualnyMB*,
4. naliczanie poza pakietami.

Dla kart sim typu *Voice* priorytety określone są następująco:

1. pakiet *Rodzinny*,
2. naliczanie poza pakietami.

Dla kart sim typu *Internet*, priorytety określone są następująco:

1. pakiet *IndywidualnyMB*,

2. pakiet Rodzinny,
3. naliczanie poza pakietami.

Dla każdej karty sim system powinien przechowywać historię wszystkich połączeń z informacją o:

- momencie wykonania połączenia (może być long, epoch time),
- typie połączenia (głosowe, sms, dane),
- nazwą pakietu, z którego zrealizowano połączenie,
- jeśli połączenie było typu głosowego, to liczba sekund, jeśli połączenie było typu dane, to liczba przesłanych MB.

Konto musi mieć możliwość wystawienia szczegółowej historii połączeń dla każdego numeru. Ponadto dla karty sim można w każdej chwili wyświetlić stan sald dla wszystkich pakietów. Można to zrealizować poprzez przeciążenie operatora <<.

W systemie nie powinno być możliwe zarejestrowanie karty sim bez istnienia konta. Pakiet można dodać/usunąć w dowolnym momencie.

Uproszczenia:

- Pomijamy to co się dzieje po zamknięciu miesięcznego cyklu abonamentu i automatyczne odnowienie sald pakietów;
- Wszędzie pomijamy kwestię naliczania opłat. Rozliczenie i wykonanie połączenia obejmuje jedynie sekundy, szt. smsów oraz MB danych. Mogą to być wartości całkowite typu int;
- Jako daty należy wykorzystać epoch time, tj. liczbę sekund jaka upłynęła od daty 1970/01/01 00:00 i do tego celu nadaje się typ danych long;
- Jeśli w ramach danego konta użytkownik ma same karty sim typu *Internet*, to nie blokujemy możliwości dodania pakietu NumeryNaKoncie, albo pakietu Rodzinnego;
- Jeśli połączenie zostało zakwalifikowane do danego pakietu, ale długość/rozmiar danych jest większy od dostępnego salda tego pakietu, to nie dzielimy połączeń na więcej niż 1 pakiet - saldo będzie miało stan ujemny.

Zaimplementuj metody potrzebne do wykonania załączonego dalej fragmentu programu. Jeśli wymagana metoda występuje w wielu wersjach (w podklasach) zaimplementuj wszystkie wersje. Jeśli metoda wywołuje metody niestandardowe, to je też należy zaimplementować. Do reprezentowania napisów możesz użyć klasy *string*. Jeśli nie potrafisz zapisać jakiejś operacji w postaci operatora, to zapisz jako metodę (będzie to nieco niżej punktowane).

Oceniana będzie implementacja *rozliczPolaczenie* **tylko** dla kart sim typu *Smartphone*.

Jeśli korzystasz z komputera, pamiętaj o zapisywaniu co pewien czas. Życzymy powodzenia!

Można wykorzystać poniższe typy danych:

```
enum TypPolaczenia{glosowe, sms, dane};
enum TypPakiet{NumeryNaKoncie, Rodzinny, IndywidualnySekundy, IndywidualnyMB, Brak};
struct Polaczenie{
    long epochTime;
    TypPolaczenia typPolaczenia;
    TypPakiet typPakiet;
    long liczbaZuzytychJednostek; };

```

Bibliografia

Tytuł zadania pochodzi z piosenki już nieistniejącego zespołu Republika, którego lider Grzegorz Ciechowski już niestety nie żyje.

Verte ->

Przykładowy fragment programu korzystający z systemu bilingowego:

```
SystemBilingowy sysB;
Konto k("Jan","Kowalski");
sysB += k;
k += new SimSmartphone("48123456789");
k += new SimVoice("48123456790");
k += new SimInternet("48123456791");
//dodajemy pakiety do karty sim
k["48123456789"].dodajPakiet(new PakietIndywidualnySekundy(100*60));
k["48123456789"].dodajPakiet(new PakietIndywidualnyMB(1024));
k["48123456791"].dodajPakiet(new PakietIndywidualnyMB(30*1024));
//dodajemy 2 pakiety do konta
k.dodajPakiet(new PakietNumeryNaKoncie());
k.dodajPakiet(new PakietRodzinny(60*60, 10*1024));
//Parametry to: liczba sekund, liczba MB
cout << sysB << endl;
k["48123456789"].wykonanoPolaczenieGlosowe("48987654321", 23*60, 1487503809);
// Parametry to: numer docelowy, długość połączenia w sekundach, data rozpoczęcia
// połączenia (epoch time)
k["48123456789"].wykonanoPolaczenieSMS("48987654321", 1487504809);
// Parametry to: numer docelowy, data rozpoczęcia połączenia
k["48123456789"].wykonanoPolaczenieData(250, 1487603809);
// Parametry to: rozmiar przesłanych danych MB (int), data rozpoczęcia połączenia
cout << k << endl;
// Poniższa instrukcja jest niedozwolona, bo dla usługi typu Internet nie można wykonywać połączeń głosowych:
// k["48123456791"].wykonanoPolaczenieGlosowe("48987654321", 23*60, 1487503809);
k["48123456789"].biling();
```

Na wyjściu powinno wyświetlić coś podobnego do:

```
Konto(Jan,Kowalski,pakietNumeryNaKoncie, saldoPakietRodzinny(3600,10240),
[Numer(48123456789,PakietIndywidualnySekundy(6000),PakietIndywidualnyMB(1024)),Numer(48123456790),Nu
mer(48123456791,PakietIndywidualnyMB(30720))])

Konto(Jan,Kowalski,pakietNumeryNaKoncie, saldoPakietRodzinny(2220,10240),
[Numer(48123456789,PakietIndywidualnySekundy(6000),PakietIndywidualnyMB(1024)),Numer(48123456790),Nu
mer(48123456791,PakietIndywidualnyMB(30720))])

[1487503809,glosowe,PakietIndywidualnySekundy,1380]
[1487503809,sms,brak,1]
[1487503809,dane,PakietIndywidualnyMB,250]
```

Po uruchomieniu powyższego kodu powinny zostać usunięte z pamięci wszystkie stworzone obiekty.