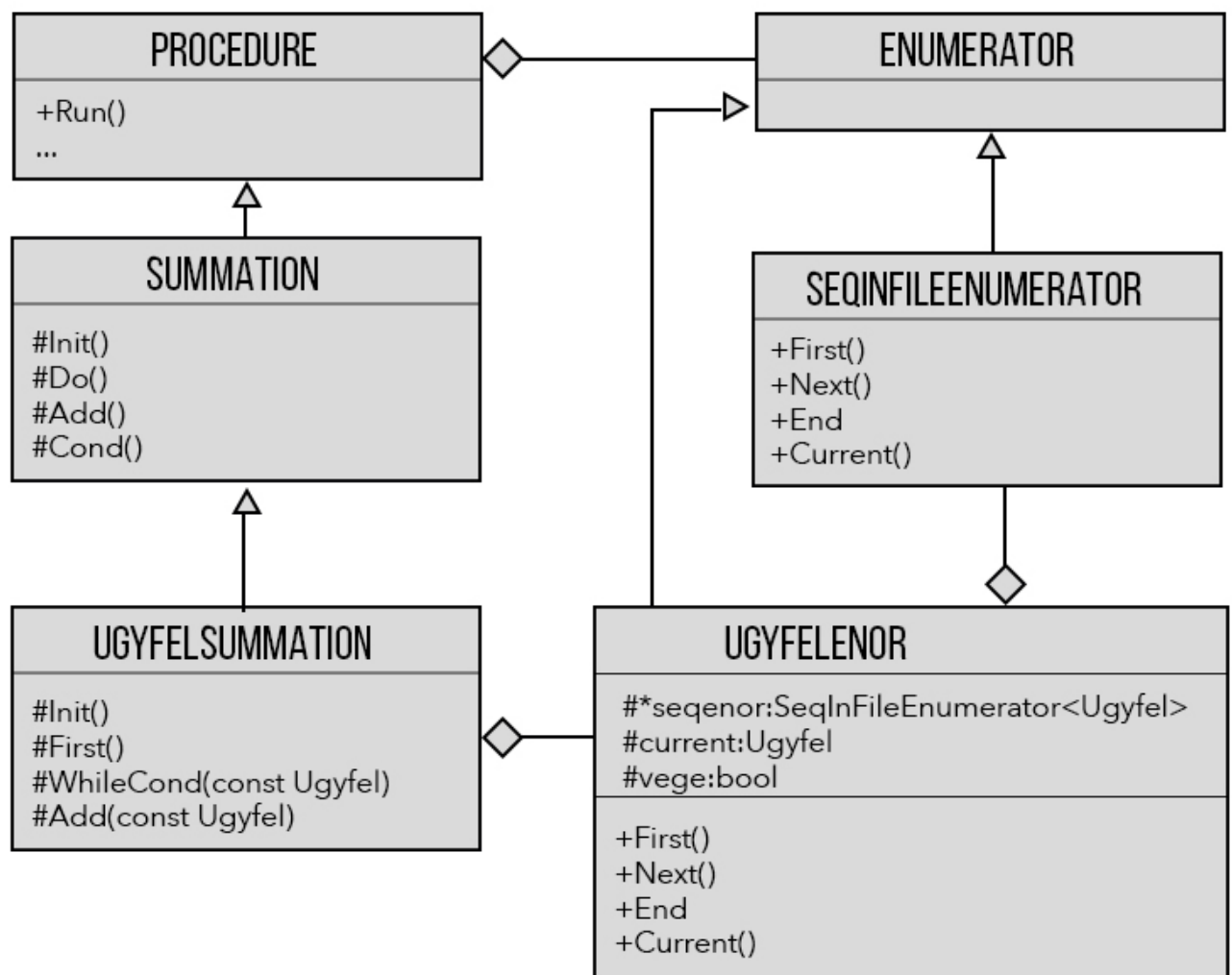


4. Egy szöveges állományban számlaszám-befizetés párokat tartalmazó sorokat helyeztünk el. (A számlaszám 8 karakter hosszú, utána egy szóköz jön, azt követően pedig egy egész szám.) Az állomány számlaszám szerint növekedően rendezett. Írjuk ki egy szöveges állományba az egyes számlák összesített forgalmát számlaszám-forgalom párokat tartalmazó sorok formájában! (A kiíratáson kívül csak egyetlenegy üres „else” ágú elágazást használjon!)

input	output
00000000 10000	00000000 8000
00000000 -2000	11111111 7000
11111111 7000	

Osztálydiagram



Ugyfel felsorolom

```
class UgyfelEnor : public Enumerator<Ugyfel>
{
protected:
    SeqInFileEnumerator<Ugyfel> *sequenor;
    Ugyfel current;
    bool vege;

public:
    UgyfelEnor(const std::string &nev)
    {
        try
        {
            sequenor = new SeqInFileEnumerator<Ugyfel>(nev);
        }
        catch (SeqInFileEnumerator<std::string>::Exceptions ex)
        {
            std::cout << "Hibas inputfile!\n";
            return;
        }
    }
    ~UgyfelEnor() { delete sequenor; }
    void First();
    void Next();
    bool End() const { return vege; }
    Ugyfel Current() const { return current; }
};
```

Felsorolo metodusai

```
void UgyfelEnor::First()
{
    sequenor->First();
    Next();
}

void UgyfelEnor::Next()
{
    vege = sequenor->End();
    if (vege) return;
    current = sequenor->Current();
    UgyfelSummation sum(current);
    sum.AddEnumerator(sequenor);
    sum.Run();
    current = sum.Result();
}
```

UgyfelSummation

```
class UgyfelSummation : public Summation<Ugyfel,Ugyfel>
{
public:
    UgyfelSummation(const Ugyfel &x) : Summation<Ugyfel,Ugyfel>(),
    ugyfel(x) {}
protected:
    Ugyfel ugyfel;
    void Init() //inicializalas
    {
        result->bank = ugyfel.bank;
        result->ossz = 0;
    }
    void First() {}
    bool WhileCond(const Ugyfel &x) const { return x.bank == ugyfel.bank; }
    //feltetelig dolgozik
    void Add(const Ugyfel &x); //ciklusmag
};
```

Főprogram

```
int main()
{
    UgyfelEnor inF("/in.txt");

    for (inF.First(); !inF.End(); inF.Next()) {
        cout<<inF.Current().bank<<" "<<inF.Current().ossz<<endl;
    }

    return 0;
}
```

Ugyfel

```
struct Ugyfel
{
    std::string bank;
    int ossz;

    Ugyfel (std::string _bank="", int _ossz=0)
    {
        bank = _bank;
        ossz = _ossz;
    }

    friend std::istream& operator>>(std::istream& f, Ugyfel &x);
};
```