

## Konzultáció – 2013. december 18.

### (Buktatók, gyakori hibák, tanácsok a második OAF-zh-val kapcsolatban)

- ne felejtjük el `include`-olni a használt *sablonosztályokat*. Ha elfelejtjük, a megvalósított-származtatott osztályok definíciójánál kapunk hibaüzenetet
- ne felejtjük el kirakni a `public` szócskát az *öröklődésnél*! Ha ezt nem tesszük meg, örökölt osztályunk nem fogja látni az ősoosztály mezőit
- figyeljünk oda a felüldefiniálendő metódusok *pontos szignatúrájára*, és ezt használjuk. Ahol `Item` meg hasonló típusnevek vannak, azt írjuk át arra, amit a típus sablonparaméterében is megadtunk, de *minden mást* hagyjunk ugyanúgy. Ne feledkezzünk el. pl a `const` jelölésről sem. Ennek hiányában nem felüldefiniálnánk (*override*), hanem túlterhelnénk (*overload*) az eredeti függvényt. Ilyenkor a fordító vagy hibát fog jelezni, hogy nem tudja eldönteni, melyiket hívja meg a kettőből, vagy az eredetit fogja ezáltal hatástalanná téve a megírt függvényünket. Amennyiben egy *absztrakt függvényt* definiálunk felül rosszul, akkor még esetleg `incomplete type` hibát is kaphatunk
- konstruktort, meg általában a *publikus függvényeket* `public`-ba rakjuk, minden mást `private`-ba. Amíg nem írunk külön láthatóságot, a `class`okban `private` a `struct`okban `public` az alapértelmezett
- a tételek hívásakor, előbb létrehozzuk a tétel *objektumát*, majd felparaméterezzük egy *bejáró-mutatóval*, majd *lefuttatjuk* (`Run()`), majd esetleg, ha olyan, lekérjük az *eredményét*
- ha egy *összegzés tétellel* egy már létező objektumba akarunk írni (*contra*, `file`-ba, ...), akkor azt át kell adni paraméterként a konstruktorának és ezzel a konstruktorral a `Summation` egyparaméteres konstruktorát is meg kell hívni
- ha a tételnek átadott felsorolóval nem szándékozunk a *teljes felsorolandó intervallumot* egy alkalommal felsorolni (pl. amikor összevonunk sorokat azonos számlaszám, eha-kód stb. szerint...), akkor a származtatott **tétel** `First`-jét felül kell definiálni üres törzsű metódusra
- bizonyos esetekben a változókat 0-ra kellhet inicializálni...
- ha minden megírandó függvényt egy forrásfájlba készítesz el, akkor figyelni kell a *deklaráció sorrendjére*: ha olyat deklarálsz hamarabb, ami használ egy a kód lejjébi szakaszában deklarált elemet, akkor az fordítási hibát fog adni
- `Counting<int>` esetében az `int` az `Item` és a `ResultType` is, `Item` azért, mert ezt adtuk meg neki, `ResultType` pedig azért, mert a *számlálásé* mindig az. A `ResultType` pl. `Counting<double>` esetben is `int`.
- a `Func` függvény végül is egy `Item`  $\rightarrow$  `Value` alakú függvény
- ha valamilyen listát „ki kell írni” konzolra vagy fájlba, azt `Summation`-nal tudjuk megoldani. Ha a listából csak a bizonyos tulajdonságúakat (progalap: kiválogatás tétéle), akkor a `Summation Cond` függvényét *is* írjuk meg az `Add` mellett
- *paraméterezett tételeket* egy új privát mező felvételével, annak konstruktorban való inicializálásával lehet készíteni
- ha konstruktorban meghívjuk az ősz konstruktorát, ne felejtjük el kiírni a sablonparamétereket