

# Házi feladat

## Programozás alapjai 2. Telefonkönyv terv

Jedla Martin  
DEC4F6

1.	Feladat .....	2
2.	Pontosított feladatspecifikáció .....	2
3.	Terv .....	4
4.	A beadott állományok.....	4

## 1. Feladat

Tervezze meg egy telefonkönyv alkalmazás egyszerűsített objektummodelljét, majd valósítsa azt meg! A telefonkönyvben kezdetben az alábbi adatokat akarjuk tárolni, de később bővíteni akarunk:

- Név (vezetéknév, keresztnév)
- becenév
- cím
- munkahelyi szám
- privát szám

Az alkalmazással minimum a következő műveleteket kívánjuk elvégezni:

- adatok felvétele
- adatok törlése
- listázás

A rendszer lehet bővebb funkcionálisú (pl. módosítás, keresés), ezért nagyon fontos, hogy jól határozza meg az objektumokat és azok felelősségét. Demonstrálja a működést külön modulként fordított tesztprogrammal! A megoldáshoz **ne** használjon STL tárolót!

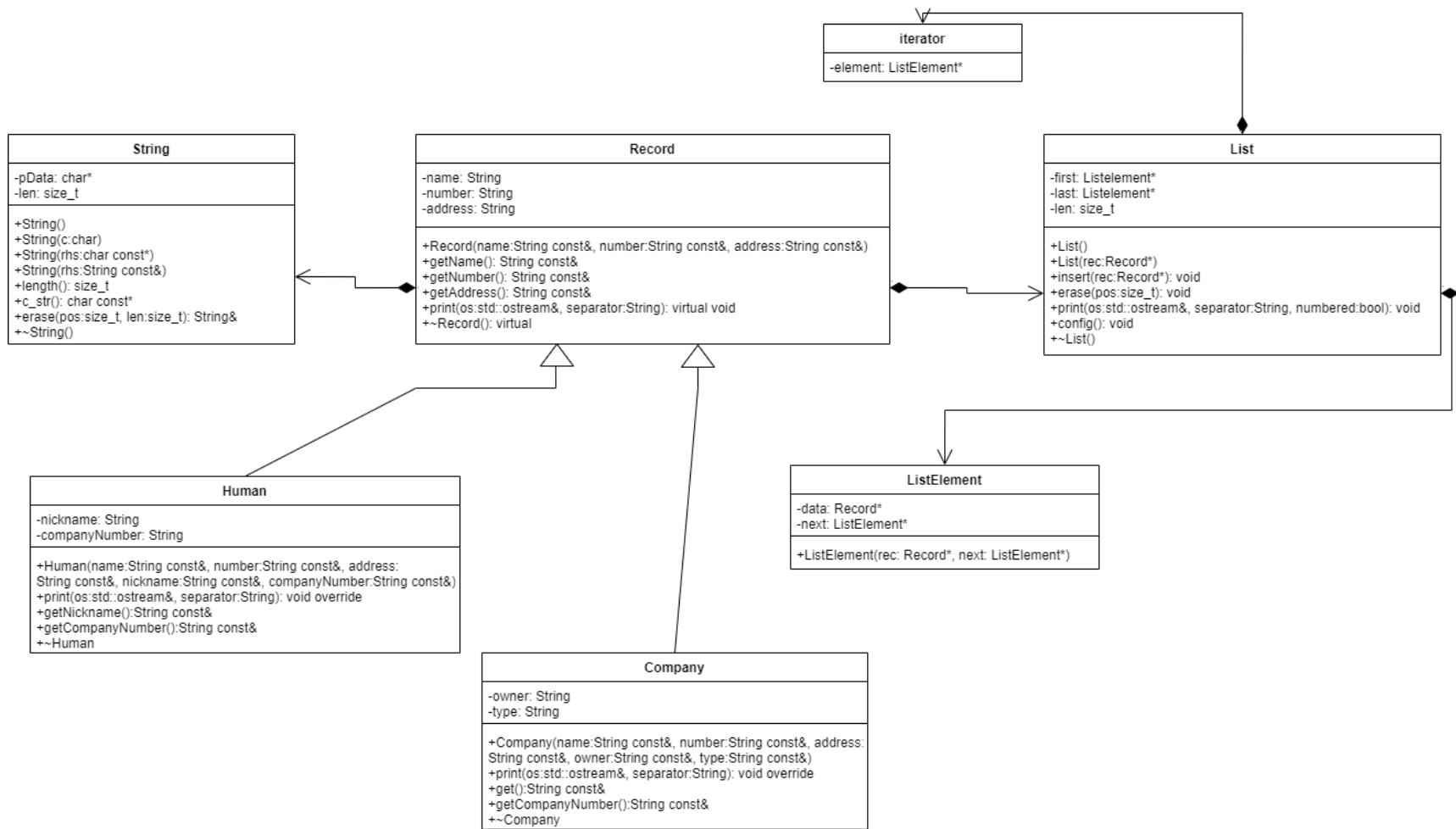
## 2. Pontosított feladatspecifikáció

A program konzolban működik. Futtatáskor megjelenik egy menürendszer, amik közül a menüpontok mellett megjelenő számokkal tudunk választani. Ezek a menüpontok a listázás, adatok felvétele, adatok törlése, illetve a kilépés.

A listázás opció egy újabb menüt hoz ki, ami közül választhatunk, hogy a természetes személyeket, cégeket vagy mindent ki akarunk listázni. Ezután a megfelelő opció választása után ki is listázza a program az adatokat. A kilistázás után bármilyen karakterrel visszakerülünk a főmenübe.

Az adatok felvétele opció egy menüt hoz ki. Választhatunk, hogy személy vagy cég adatait akarjuk felvenni. Ember esetén egy nevet, becenevet, címet, munkahelyi számot és privát számot lehet megadni. A név, privát szám megadása kötelező, a többi lehet üresen hagyni. Cég opció kiválasztása esetén szintén bekéri a megfelelő adatokat.

A törlés opció választásakor kilistáz minden adatot előtte egy sorszámmal, és annak a sorszámnak a kiválasztásával tudjuk törölni azt az adatot.



### 3. Terv

#### List::config():

A függvény beolvas a „records.txt” fájlból. A fájlban minden sorban egymáshoz tartozó adatok vannak pontosvesszővel elválasztva. Először a típus (1-ember, 2-cég), utána sorban a név, telefonszám, cím majd az adott típushoz tartozó adatok. A függvény soronként olvas be a String::getline segítségével (hasonlít a getline()-hoz, csak saját String típussal működik). Majd a Stringet, amibe beolvasott, feldarabolja pontosvesszőnként, és létrehozza a megfelelő objektumot a megfelelő adatokkal. Majd a listához adja abc sorrend szerint, amiben a futási idő közben van minden adat.

#### List::insert(Record\*):

A paraméterként kapott adatot beszúrja a listában, a név szerint növekvő sorrendben. Feltételezi, hogy a név nagy kezdőbetűvel kezdődik.

#### List::erase(size\_t):

Kitörli a listából a paraméterként megadott sorszámú elemet (az első elem az 1).

#### List::print(std::ostream&, String)

Kiírja a lista tartalmát a megfelelő formátumban a megadott std::ostreamre, illetve a megadott String-el szeparálva.

#### String::erase(size\_t, size\_t):

A Stringből kitörli az első paraméter indextől a második paraméter mennyiséget. Megváltoztatja a hosszt.

#### isPhonenumber(String const&):

Ellenőrzi a paraméterként átadott Stringen, hogy megfelelő formátumú. A megfelelő formátum a tíz karakteres hossz, illetve csak szám karakterekből álló String.

#### Menu(List&):

A menükezelő függvény. Az átadott listával dolgozik. Lehet az átadott lista kilistázását, a listához adat hozzáadását illetve a listából elem törlését választani.

### A beadott állományok

telefonkonyv.pdf – dokumentáció

standard\_input.txt – bemenet a consolera

main.cpp – meghívja a Menu függvényt, azaz megvalósítja a konzolos vezérlést

string5.h – std::stringhez hasonló osztály header fájlja, nagyrésze a laboron megírt, a függvényeinek a deklarációját tartalmazza

string5.cpp – string5.h-ban található függvények definiálása

list.h – egy lista megvalósítása iterátorral, a függvények deklarációját tartalmazza

list.cpp – a list.h-ban deklarált függvények definíciója

record.h – az űsosztályt tartalmazza, a függvényei deklarációját

record.cpp, , human.cpp, human.h, company.cpp, company.h, menu.cpp, menu.h – forrásfájlok

records.txt – ebbe a fájlba menti a program az adatokat, és innen olvassa ki