

# HÁZI FELADAT

Programozás alapjai 2.

Feladatválasztás/feladatspecifikáció

Dukát Nándor  
YZV8QV

2023. április 16.

---

## TARTALOM

1. Feladat .....	2
2. Feladatspecifikáció .....	2
3. Terv .....	2
3.1 Osztálydiagram.....	3

# 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 funkcionalitású (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. Feladatspecifikáció

A program indításakor a konzolban megjelenik a főmenü, amely lehetővé teszi a felhasználó számára az alkalmazás egészének irányítását. A főmenüből különböző gombok leütésével kezelhetők a névjegyek. Például a "k" gomb + Enter billentyű lenyomásával beléphetünk a keresés funkcióba.

A program lehetőséget biztosít az adatok (névjegyek) felvételére, amelyek tartalmazzák a teljes nevet, becenevet, otthoni címet, munkahelyi telefonszámot és privát telefonszámot. Az adatokat egyesével vagy akár az összeset egyszerre is törölhetjük. Emellett lehetőség van az adatok listázására és módosítására is. Az adatok mentésére manuálisan is van lehetőség a főmenüből, de a program automatikusan menti azokat kilépéskor.

Az adatok egy txt fájlba mentődnek egymás alá. Felvételük a konzol bemenetén történik, a program ebben segítséget nyújt, hogy mikor, mit kell megadni, milyen formátumban stb.

A program hibás használata esetén jelzi a hibát és/vagy kivételt dob. Az adatok dinamikusan tárolódnak, így a felhasználó végtelen mennyiségű adatot tud felvenni. Az adatok mentése fájlba történik, és a már létező adatokat a program indításakor beolvassa.

A tesztelésre egy olyan programot készíték, amely teszteli a program funkcióit a felvett adatok és a szabványos bemeneti adatok felhasználásával. A tesztelés során szándékosan hibás adatokat is használok, hogy azokat a program kivételezésének szemléltetésére felhasználjam.

## 3. Terv

A program 9 osztályt használ, amelyben egy sablonként megvalósított lista osztály (List) található. Ez a lista dinamikusan képes adatokat tárolni, keresni és törölni. A program ezt a listát használja a telefonkönyv adatainak tárolására, amelyeket sablonként vesz át (*Contact\**).

A Contact egy absztrakt osztály, ezt örökölik a Személy (*Person*) és a Cég (*Company*) osztályok. Ezt a két típusú adatot tudjuk felvenni a telefonkönyvbe, de lehetőség van korlátlanul bővíteni ezt az osztály a későbbiekben. A Kontakt (*Contact*) osztály a következőket tárolja: Név (*Name*): Vezetéknév, Keresztnév és Becenév, Cím (*Address*): Irányítószám, Város, Utca és Házszám, Telefon (*Phone*): Telefonszám. Mindezen adatok külön osztályként vannak megvalósítva a későbbi bővíthetőség és módosíthatóság érdekében.

A Name, Address és Phone osztályokban az adatok String típusúak. Ez egy általam létrehozott osztály, amely dinamikusan tárolják a szövegeket.

### 3.1 Osztálydiagram

