

Telefonkönyv programozói dokumentáció

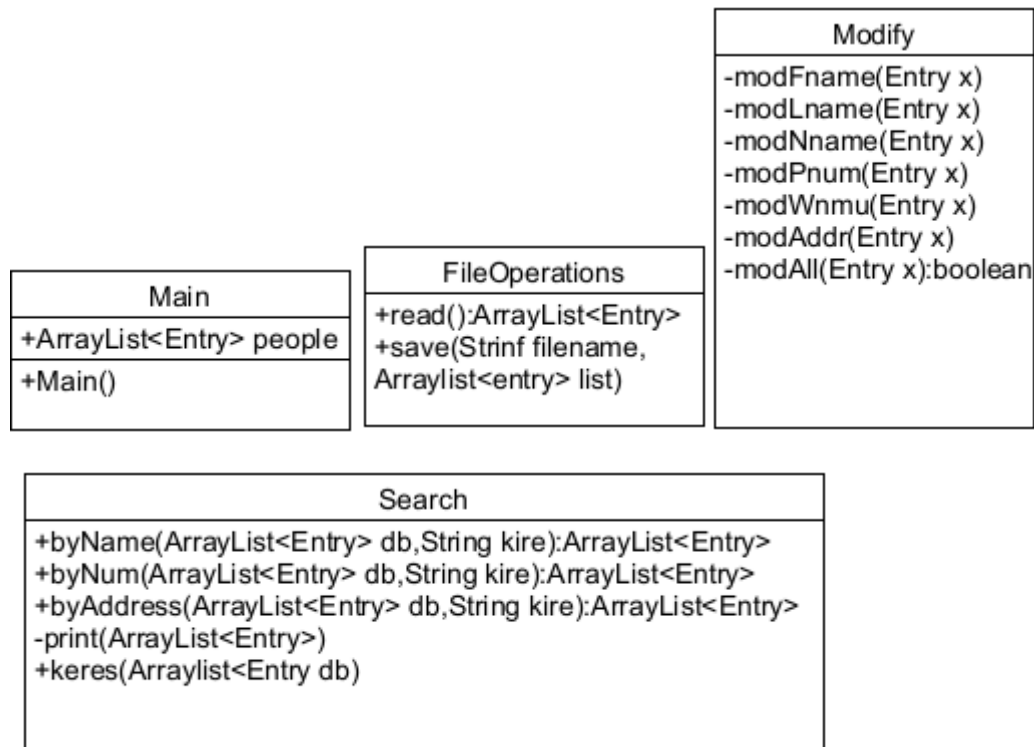
Contents

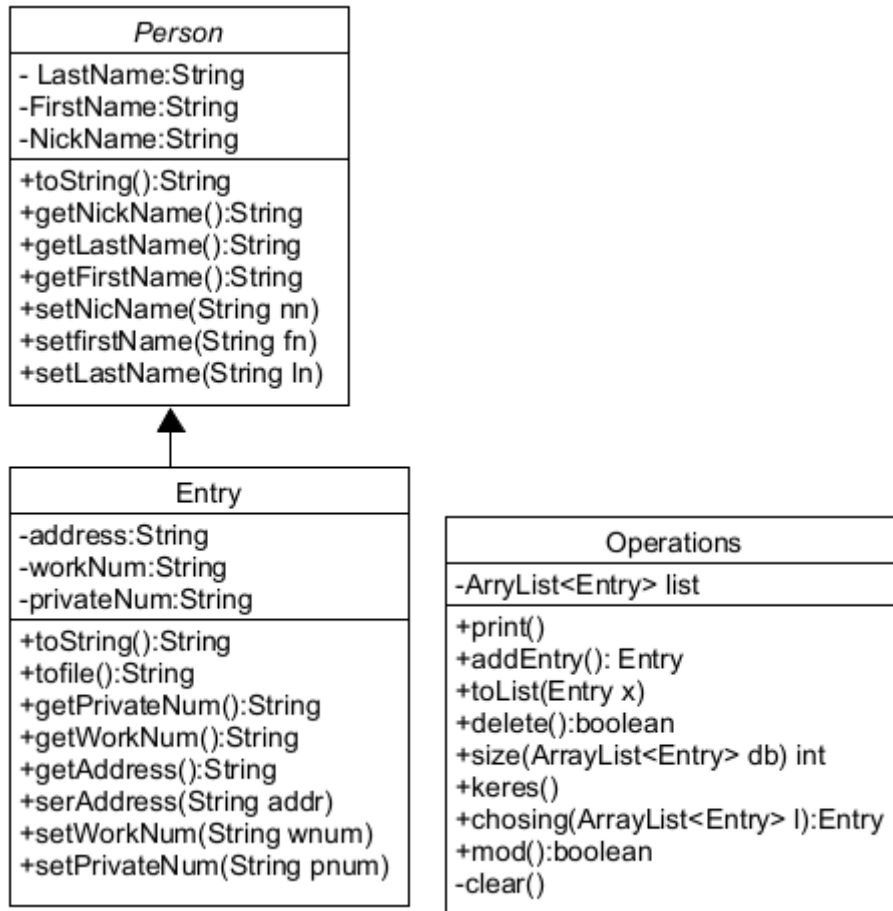
Osztályok	3
UML diagram	3
Osztályok és függvényke részletes leírása.....	5
Main.java	5
FileOperations.java.....	5
Read():	5
Save(String filename, ArrayList<Entry> list):.....	5
toEntry(String [] list):	5
Person.java	5
Entry.java.....	7
toFile():	7
Search.java	7
Byname():	7
byAddress():.....	7
byNum():.....	7
Print(ArrayList<Entry> temp):	7
Keres(ArrayList<Entry> db):.....	7
Operations.java	8
Print():.....	8
addEntry():.....	8
toList(Entry x):	8
Delete():.....	8
Keres():	8
Chosing(ArrayList<Entry> l):	8
Mod():.....	8
Clear():	8
Modify.java.....	9
modAll(Entry x):.....	9

Osztályok

- Main.java
- FileOperations.java
- Person.java
- Entry.java
- Search.java
- Operations.java
- Modify.java

UML diagram





Osztályok és függvényke részletes leírása

Main.java

Main():

Itt indul el a program. Tartalmaz egy ArrayList-et, amit a betöltött adatokat fogja tárolni. Ez a fő menüje a programnak is. Választhat a felhasználó, hogy mit szeretne csinálni. Itt is van ellenőrzés hogy mit választ a felhasználó.

1. Rekordok listázása.
2. Rekord hozzáadása
3. Rekord törlése
4. Keresés
5. Rekord szerkesztése

0.val meg i lehet lépni a programból.

FileOperations.java

Fájl kezelésért felel ez az osztály.

Read():

Ez az a függvény, ami megvalósítja a fájlból való beolvasást. CSV falból olvassa be az adatokat majd Entry-vé alakítja. Egy ArrayList-tel tér vissza, ami Entry típusú objektumokat tartalmaz. Mindig egy listával tér vissza akkor, ha nem találja a fájlt, akkor üres lesz a lista.

Save(String filename, ArrayList<Entry> list):

Paraméterei:

- Filename: a fájl neve, amibe menteni szeretnénk.
- List: aza a lista, ami tartalmazza az adatokat.

Ez az a függvény, ami megvalósítja a fájlba való írást. CSV formátumba fogja kiírni az adatokat, amit a listában kapott.

toEntry(String [] list):

paraméterei:

- List: egy tömb, amit a beolvasás során kap, ami feldarabolja majd Entry-vé alakítja oket.

Egy új Entry objektummal tér vissza.

Person.java

Egy abstract osztály, ami egy ember Keresztnevét, vezetéknévét és becenevét tárolja.

Attribútumai:

- Private String LastName
- Private String FirstName
- Private String NickName

Vannak getter, setter és toString függvényei.

Entry.java

A Person-ból öröklődik. Ezt az osztályt használjuk adattárolásra.

Attribútumai:

- Private String address
- Private String workNum
- Private String privateNumber
- +az ősz osztály

Vannak getter, setter és toString függvényei. Van még egy toFile függvénye is.

toFile():

Ez az a függvény az Entry adatokat olyan formátummá alakítja, ami kiírható CSV-formátumba. „,„ -vel elválasztva egymás mögé fűzi az adatokat.

Search.java

Ez az az osztály, ami a kereséseket valósítja meg Regex-szel.

A kereső függvények:

- byName
- byAddress
- byNum

Mind a 3 függvény paramétere egy ArrayList<Entry> amibe keresni fog és egy String, hogy mire fog keresni. Mindig egy listával fog visszatérni, ha volt találat akkor az a lista fogja tartalmazni, vagy 0 elem lesz neki.

Byname():

A felhasználó név alapján tud keresni. (Kereszt, Vezeték vagy becenév alapján)

byAddress():

A felhasználó cím alapján tud keresni.

byNum():

A felhasználó telefonszám alapján tud keresni. (magán vagy munkás telefonszám alapján)

Print(ArrayList<Entry> temp):

Ki írja a paraméterben kapott listát.

Keres(ArrayList<Entry> db):

Paramétere az a lista, amibe keresni fog.

Választhat a felhasználó, hogy milyen módon szeretne keresni. Ellenőrzi a bementet is, hogy számot irt be a felhasználó és 0-3 van. 0-val lehet visszamenni.

1. Név alapján (byname)
2. Telefonszám alapján (byNum)
3. Cím alapján (byAddress)

Operations.java

Attribútuma az a lista, ami az adatokat tartalmazza. Konstruktor ként kapja meg.

Ez az osztály tartalmazza azokat a műveleteket, amit a felhasználó végre tud hajtani.

Print():

Előtte letörli a konzolt. Ha van a listában tárolt adat akkor ki listázza az indexükkel együtt. Ha nincs akkor ki írja, hogy nincs tárolt adat.

addEntry():

Előtte letörli a konzolt. Amikor a felhasználó új névjegyet szeretne hozzáadni akkor, ez kéri be az új adatokat és egy új Entry-vel tér vissza.

toList(Entry x):

Ez adja hozzá a listához az új Entryt amit paraméterként kap.

Delete():

Letörli a konzolt. Majd kiválaszthatja a felhasználó, hogy melyiket szeretné törölni a Chosing függvény segítségével. Ha a törlés sikeres volt True-val ha nem akkor false-al tér vissza.

Keres():

Letörli a konzolt. A Search.java-bol a keres()-t hívja meg.

Chosing(ArrayList<Entry> l):

Paraméternek az a t listát kapja mibe dolgozni fog.

Letörli a konzolt. Itt lehet kiválasztani, hogy melyik névjeggyel szeretnénk dolgozni a törléskor meg a módosításkor. Ha lista üres akkor null-lal tér vissza, ha nem üres akkor lehet választani az indexük alapján. Itt is történnek ellenőrzések. Hogy számot adott-e meg a felhasználó és egy bizonyos tartományon belül van. Ha ezen mind átmegy akkor visszatér a kiválasztót névjeggyel.

Mod():

Ez a függvény, amivel módosítani lehet egy névjegyet. Letörli a konzolt. Majd a chosing()-al választ a felhasználó egy névjegyet amit módosítani akar. Ha sikerült a módosítás true-val, egyébként meg false-al tér vissza.

Clear():

Ez az a függvény, ami letörli a konzolt.

Forrás: <https://stackoverflow.com/questions/2979383/how-to-clear-the-console>

Modify.java

Függvényei:

- `modFname(Entry x)` keresztnév módosítására
- `modLname(Entry x)` vezetéknév módosítására
- `modNname(Entry x)` becenév módosítására
- `modPnum(Entry x)` privát telefonszám módosítására
- `modWnum(Entry x)` munkás telefonszám módosítására
- `modAddr(Entry x)` cím módosítására

Ezek függvények mindegyiknek paramétere `Entry x` ami egy általunk kiválasztott névjegy. Mindegyik mikor meghívjuk akkor kiírja a névjegy korábbi vezetéknévét, keresztnévét, becenevét, címét vagy valamelyik telefonszámát. Ezután a felhasználó megadhat egy újat.

`modAll(Entry x)`:

paraméter egy általunk választott névjegy, aminek valamelyik paraméterét módosítani akarjuk. A felhasználó választhat, hogy mit szeretne módosítani. Itt is történik ellenőrzés, hogy mit adott meg a felhasználó.

1. Vezetéknév
2. Keresztnév
3. Becenév
4. Privát telefonszám
5. Munkás telefonszám
6. Cím

Módosítás után ha sikerült akkor `true`-val egyébként `false`-sal tér vissza a fő menőbe.