



**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Irányítástechnika és Informatika Tanszék

Vasas Mónika  
I5PC3J

# **TELEFONKÖNYV**

Programozói dokumentáció

Üzemmérnök-informatikus BProf  
Objektumorientált programozás NHF

# 1 Általános elvek

- Java.io, Java.util, org.junit.Assert csomagok néhány osztályának, interfészének használata.
- Junit 5 használata a teszteléshez.
- Konzolos menürendszer numerikus menüpontokkal. Más formátumú funkcióválasztásra a program NumberFormatException-t dob hibaüzenettel. Nem létező számú opció esetén else ágban lévő hibaüzenet ír ki.
- Három alap osztályra épül a program: Main.java, Konyv.java, Kontakt.java és további kettőre, NoSuchContactException és EmailFormatException néven, kivételkezelés miatt. Utóbbiak mindegyike a telefonkonyv package-ben van.

## 2 Kontakt.java, Kontakt class

Egy publikus osztály, melyen belül létrehozom a telefonkönyvben tárolható kontaktok konstruktorát, get-set metódusait és a toString()-et. A következő adatokat lehet tárolni:

- Vezetéknév
- Keresztnév
- Becenévi
- Cím
- Munkahelyi szám
- Privát szám
- E-mail cím

A toString()-ben az átláthatóság miatt minden személy bejegyzése „Kontakt [„-al kezdődik. A [-en belül az adatok megadása után ]-el zárul.

### 3 Konyv.java, Konyv class

Egy publikus osztály, melyen belül a menürendszer funkcióinak metódusai vannak megírva.

Először létrehozok egy private, Kontakt típusú listát, ami a Konyv konstrukora új listaként definiálva. Utóbbinak van set-get metódusa és további egy, amely az előbb említett listát egy másik, szintén Kontakt típusú listára tudja felülírni. Ez azért szükséges, hogy egy, már meglévő telefonkönyv betöltése esetén a fájlkezelés megfelelően működjön.

**addKontakt(Kontakt k):** Hozzáad egy kontaktot a telefonkönyvhöz.

**listKontaktok():** Kilistázza a korábban hozzáadott összes kontaktot. Ezt Stringként adja vissza, hogy ehhez a metódushoz tartozó Junit teszt sikeresen lefusson.

**keresKontakt(String szam):** Munkahelyi vagy privát telefonszám alapú kontaktkeresést valósít meg úgy, hogy összehasonlítja, hogy a paraméterben átadott telefonszám valamelyik Kontakt valamelyik típusú telefonszámával megegyezik-e. Ha igen, visszaadja a kontaktot. Ha nem, akkor NoSuchContactException-t dob, amely jelzi, hogy nincs ilyen kontakt a telefonkönyvben. Tartozik ehhez a metódushoz egy NumberFormatException, amit a Main.java részben fejtek ki.

**keresKnev(String teljesnev):** Teljes név alapú keresés. Ha a kért kontakt név létezik a telefonkönyvben, visszaadja a kontaktot, egyébként NoSuchContactException-t dob.

**torolKontakt(String szam):** Ugyanazon az elven működik, mint a keresés, de ha megtalálja a megfelelő kontaktot, kitörli a könyvből. Ekkor true értékkel tér vissza, különben NoSuchContactException-t dob.

**torolKnev(String tnev):** Teljes név alapú törlés. Ha a kontaktot megtalálja így, kitörli a könyvől. Ekkor true értékkel tér vissza, különben NoSuchContactException-t dob.

**keresEmail(String email):** Kontakt keresése e-mail cím alapján. Ha a kért e-mail megtalálható a telefonkönyvben és megfelelő formátumú, visszaadja a kontaktot, egyébként NoSuchContactException-t vagy EmailFormatException-t dob.

**torolEmail(String email):** Kontakt törlése e-mail cím alapján. Ha a kért e-mail megfelelő formátumú megtalálható a telefonkönyvben, kitörli a kontaktot, majd true értékkel visszatér és egyébként NoSuchContactException-t vagy EmailFormatException-t dob.

**konyvmentese(String hova):** Ezzel a metódussal lehet elmenteni a telefonkönyvet, itt történik a fájlkezelés kiíratás része. A program mappájába, egy UTF-8 kódolású szövegfájlba mentődhet. A fájl nevét a felhasználó is megadhatja, a .txt formátum nélkül. Hiba esetén IOException-t dob.

**konyvOlvasasa(String honnan):** Ez írja le, hogyan lehet betölteni egy, már meglévő, UTF-8 kódolású szövegfájlban és csak a program mappájában lévő telefonkönyvet a programba. Itt létrehozok egy új listát, és a fájlbeolvasás után visszaadom ezt, új Konyv-ként. Hiba esetén IOException-t dob.

## 4 Main.java

Itt valósul meg a menürendszer és kerülnek meghívásra a Konyv.java-ban definiált metódusok. A publikus Main osztályban létrehozásra került egy Scanner és egy metódus, ami annak egy következő adatot kérő, levágott formátumával tér vissza. Továbbá van egy **checkNumber(String szam)** és **checkEmail(String e\_mail)** metódus, ami a telefonszám és az e-mail cím formátumának helyességét ellenőrzi Regex kifejezésekkel. Hiba esetén NumberFormatException-t és EmailFormatException-t dob, hibaüzenettel.

A main metódusban a Scanner-nek beállítom, hogy \n legyen az elválasztója és létrehozok egy új Konyv objektumot, majd egy while loopon belül elkezdem felépíteni a menürendszert:

### 1. Menüpont: Új kontakt felvétele

Új Kontakt objektum kerül létrehozásra, majd bekérem a felhasználótól az adatokat. A fentebb említett checkNumber metódust itt is alkalmazom a privát és munkahelyi telefonszám bekérésénél. Ezek után az addKontakt()-al hozzáadom a kontaktot a telefonkönyvhöz. Nem megfelelő formátum esetén NumberFormatException-t dob.

### 2. Menüpont: Kontaktok listázása

A listKontaktok metódus meghívása a Konyv objektumra, és az eredmény kiírása.

### 3. Menüpont: Kontakt törlése telefonszám alapján

Megkérdezem a törölni kívánt kontakt telefonszámát, majd azt, hogy a felhasználó biztosan törölni szeretné-e a kontaktot. „Igen”-el „i” betűvel, „Nem”-el „n” betűvel válaszolhat, amit jelzek is felé. Nem kis-és nagybetű érzékeny. „Igen” esetén a torolKontakt meghívásra kerül, majd kiíratom, hogy a kontakt sikeresen törlésre került. Ha nincs ilyen kontakt, NoSuchContactException-t

dob, „Nem” esetén nem csinállok semmit, csak jelzem, hogy a művelet vissza lett vonva.

#### **4. Menüpont: Kontakt törlése név alapján**

Bekérem a törölni kívánt kontakt vezetéknevét, keresztnévét. Egy másik String-ben ezeket összefűzöm, hogy megkapjam a személy teljes nevét. Ezután meghívom rá a torolKnev metódust. Ha létezik a kontakt, akkor a törlés sikeres, melyről a felhasználót is értesíti. Ha nem, NoSuchContactException-t dob.

#### **5. Menüpont: Kontakt törlése e-mail cím alapján**

A felhasználótól bekérem a törölni kívánt kontakt e-mail címét, majd ha a formátum megfelelő, meghívom rá a torolEmail metódust. Ha létezik a kontakt, a törlés sikeres, melyről a felhasználót is értesíti. Ha nem, NoSuchContactException-t dob.

#### **6. Menüpont: Kontakt keresése telefonszám alapján**

Bekérem a felhasználótól a keresett telefonszámot, majd ellenőrzöm annak formátumát a checkNumber metódussal. Ha a formátum megfelelt, létrehozok egy új Kontakt objektumot és meghívom a keresKontakt metódust. Ha a Kontakt nem null, az az létezik ilyen kontakt, akkor kiíratom az összes további adatát a konzolra. Ha nem létezik, NoSuchContactException-t dob. Más hibás formátum esetén NumberFormatException-t, hibaüzenettel.

#### **7. Menüpont: Kontakt keresése név alapján**

Bekérem a felhasználótól a keresett kontakt vezetéknevét, majd a keresztnévét. Egy másik String-ben utóbbi kettőt összefűzöm, hogy megkapjam a személy teljes nevét. Létrehozok egy új Kontakt objektumot és meghívom a fentebb említett String-re a keresKNev metódust. Ha a Kontakt nem null, az az létezik ilyen nevű kontakt, akkor kiíratom az összes további adatát a konzolra. Ha nem létezik, NoSuchContactException-t dob.

## **8. Kontakt keresése e-mail cím alapján:**

A keresett e-mail cím bekérése után léterhozok egy új Kontakt objektumot és ha a formátum megfelelő, meghívom rá a keresKontakt metódust. Ha létezik a Kontakt, akkor kiíratom az összes további adatot. Ha nem létezik NoSuchContactException-t dob.

## **9. Menüpont: Telefonkönyv mentése**

A fájlnev bekérésre kerül, majd meghívódik a konyvMentese metódus. Sikeres mentés esetén szövegesen jelzem ezt a felhasználónak.

## **10. Menüpont: Meglévő telefonkönyv betöltése**

A fájlnev bekérésre kerül, majd meghívódik a konyvOlvasasa metódus. Sikeres importálás esetén szövegesen jelzem ezt a felhasználónak.

## **11. Menüpont: Kilépés: System.exit(0)**

Végül bezárom a Scanner-t.



## **Tartalom**

<b>1 Általános elvek.....</b>	<b>2</b>
<b>2 Kontakt.java, Kontakt class.....</b>	<b>3</b>
<b>3 Konyv.java, Konyv class.....</b>	<b>4</b>
<b>4 Main.java .....</b>	<b>6</b>