

# Koronavírus Java projekt – a tervezés és megvalósítás mérföldkövei

## Coronavirus Java project – milestones of design and implementation

KACZUR Sándor

it-tanfolyam.hu, Budapest, Magyarország, kaczursandor@gmail.com

### Abstract

*The lecture/article describes a case study used/usable in software development courses. The Java project collects data from two different websites: <https://koronavirus.gov.hu/elhunytak> and <https://www.worldometers.info/coronavirus/country/hungary/>. The data is stored in a local file system, converted, combined, and finally processed after some data cleaning. Related elements of the main topics of the it-tanfolyam.hu Java SE and Java EE software development courses: OOP-MVC, swing GUI, event-driven programming, handling text files, network communication, exception handling. The lecture/article describes the most important steps and milestones in the planning and implementation of the Java project.*

**Keywords:** software development, object-oriented programming, Java programming language, case study

### Kivonat

*Az előadás/cikk szoftverfejlesztő tanfolyamokon használt/használható esettanulmányt ismertet. A Java projekt két különböző weboldalról gyűjt össze adatokat, ezek: <https://koronavirus.gov.hu/elhunytak> és <https://www.worldometers.info/coronavirus/country/hungary/>. Az adatokat helyi állományrendszerben tárolja, konvertálja, összefésüli, végül némi adattisztítást követően feldolgozza. Az it-tanfolyam.hu Java SE és Java EE szoftverfejlesztő tanfolyamok tematikáinak kapcsolódó elemei: OOP-MVC, swing GUI, eseményvezérelt programozás, szövegfájlok kezelése, hálózati kommunikáció, kivételkezelés. Az előadás/cikk ismerteti a Java projekt tervezésének és megvalósításának fontosabb lépéseit, mérföldköveit.*

**Kulcsszavak:** szoftverfejlesztés, objektumorientált programozás, Java programozási nyelv, esettanulmány

## 1. BEVEZETÉS

Az algoritmusok, programozási tételek és a Java programozás oktatása során többször áll rendelkezésre olyan kritikus tömegű ismeretanyag, amely lehetővé teszi nagyobb léptékű projektmunkák, komplex esettanulmányok megtervezését, elkészítését, tesztelését. A Koronavírus Java projekt esetében a Java SE tanfolyamra épülő Java EE tanfolyam témaköreinek [1, 2] figyelembevételével (hivatkozva a szoftverfejlesztő tanfolyamok szakmai és orientáló moduljaira is):

- az előzmények, amire építeni kell: objektumorientált alapismeretek, POJO – szakterületi osztályként, összetartozó adatok és a rajtuk elvégezhető műveletek összességéként értelmezve, MVC architekturális és Factory tervezési minta, adatszerkezetek, programozási tételek, swing GUI,
- a projekt közben elsajátítandó ismeretek, amelyek alkalmazásra kerülnek: kivételkezelés, szövegfájlok kezelése, hálózati kommunikáció, generikus adatszerkezetek feldolgozása,
- a projekt kiegészíthető, ráépülhet: objektumfolyamok feldolgozása, Stream API műveletek használata, adatkonverzió fájl(ok)ból adatbázisba, teljes funkcionális, dinamikus elosztott alkalmazássá, webalkalmazássá alakítható a koncepció.

A projekt célja: egy evolúciós esettanulmány fejlesztése. Java projektként kerül megvalósításra. A Java projekt kötelezően kivételkezelést használ a következő két követelmény miatt. Szövegfájlokat kell kezelnie a `java.io` és `java.nio` csomagokkal [3, 4], helyi fájlrendszerben HTML és CSV formátumokban. Hálózati kommunikációt kell megvalósítani a `java.net`, `javax.net.ssl`, `java.security` és a `java.security.cert` csomagokkal, mivel távoli fájlok tartalmát kell összegyűjteni [5, 6, 7]. A Java projekt csoportmunkában valósul meg, kisebb, 3 főből álló fejlesztői/tesztelői csapatot feltételezve. Rutinosan kell ismerni a hivatkozott két tervezési mintát [8]: eseménykezelés, adatáramlás, adattárolás, alkalmazáslogika

szempontjából és világosan be kell tartani az egyes rétegek határait, köztük egyértelmű interfészek meghatározásával. Ez a csoportmunka során megkönnyíti a különböző részfeladatokra való specializálódás lehetőségét, a dokumentálást, valamint a továbbfejlesztést is.

## 2. TERVEZÉS

A tervezés során el kell dönteni, hogy az ismert/tipikus ötféle szerver/hálózat/kliens architektúra alapján melyik jöhet, melyek jöhetnek szóba az alábbiak közül: terminál emuláció, elosztott megjelenítés, távoli megjelenítés, elosztott funkcionalitás, távoli adatkezelés. Előfordulhat, hogy ezek közül akár többféle szemlélet is beépíthető a Java projektbe.

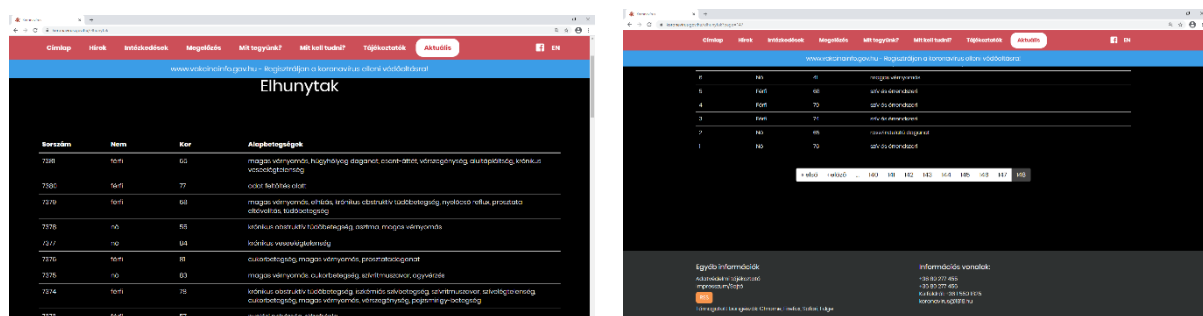
A szükséges lépések koncepcionális szinten:

- Adatszerzés távoli forrásból:  
<https://koronavirus.gov.hu/elhunytak>,  
<https://www.worldometers.info/coronavirus/country/hungary/>,
- Input HTML szövegfájlok feldolgozása helyi fájlrendszerben, adatok összefűtése CSV fájlban,
- Köztes CSV fájl létrehozása, betöltése, leképezése generikus POJO kollekcióvá,
- Belső adatszerkezet feldolgozása, szempont a hatékonyság [9],
- Output grafikonok előállítás:  
 swing GUI [10],  
 sablon alapján generált HTML fájlok.

A programnak – miután összegyűjtötte a szükséges adatokat – grafikonokat kell készítenie. A grafikonok kétféleképpen valósulnak meg: egyrészt a JFreeChart csomagjaival [11, 12] swing GUI felhasználói felületen a Java programban, másrészt a Google Chart mintaprogramjai [13] alapján több, önálló, dinamikus generált, böngészőben megjelenítendő weboldalként HTML és JavaScript kombinálásával.

### 2.1. Magyarországi adatforrás

A magyarországi adatforrás elsődleges weboldala a <https://koronavirus.gov.hu/elhunytak>. Táblázatos formában, ötvenesével szerepelnek rajta az koronavírus fertőzésben elhunytak összetartozó adatai: sorszám, nem, kor, alapbetegségek, retrospektív sorrendben, de a dátum nem jelenik meg (1. ábra). Ha lenne dátum, akkor természetesen -1 napot kellene figyelembe venni a feldolgozásnál. Az adatok közötti lapozás paraméteresen valósul meg (például: [.../elhunytak?page=1](https://koronavirus.gov.hu/elhunytak?page=1)) és ehhez kötődően a weboldalakon lévő táblázat alatt dinamikusan generált linkek találhatók: első, előző, ..., 147, 148. Az utolsó lapon lehet kevesebb adat is ötvennél. Az adatok frissítése (adatszolgáltatás) a koronavírus egyes hullámaiban eltérően valósult meg: napi frissítés, hétköznaponként és a hétvégi adatok a következő hétfőn jelentek meg, illetve heti ütemezés is előfordult [14].



1. ábra. Magyarországi adatforrás, koronavírus input adatok, webes nézet

Nincs külön REST API szolgáltatás, amin keresztül közvetlenül, XLS, XML vagy JSON formátumban lekérdezhetők lennének az adatok, ezért web scraping/crawling tevékenység során egyszerű robotot kell készíteni [5-7]. A folyamat magába foglalja a webhely HTML-kódjának (2. ábra) letöltését, a HTML-kód feldolgozását/elemezését (parszolás) és a kívánt adatok kinyerését.

Az adatokat – feldolgozás előtt – érdemes ellenőrizni, tisztítani, mert nem egységesek (nincs standardizálás). Például: eltérőek az alapbetegségek rövidítései, egyesek alkalmazottak az alapbetegségek magyar és latin elnevezései, eltérő karakterkódolással szerepelnek a szövegek (nem: nő, nő, nő, no; alapbetegségeknel is), nem egységes a szeparátor az alapbetegségeknel (tipikusan vessző szóközzel, de előfordul vessző,

pontosvessző, szóköz nélkül, utolsó utáni vessző is), előfordul hiányzó kor, előfordul ismétlődő sorszám, valamint többféleképpen rögzített, ha az elhunytak nem volt alapbetegsége.

```

195 <tr class="even">
196 <td class="views-field views-field-field-elhunytak-sorszam" >
197 7378 </td>
198 <td class="views-field views-field-field-elhunytak-nem" >
199 nő </td>
200 <td class="views-field views-field-field-elhunytak-kor" >
201 55 </td>
202 <td class="views-field views-field-field-elhunytak-alapbetegsegek" >
203 krónikus obstruktív tüdőbetegség, asztma, magas vérnyomás </td>
204 </tr>
205 <tr class="odd">
206 <td class="views-field views-field-field-elhunytak-sorszam" >
207 7377 </td>
208 <td class="views-field views-field-field-elhunytak-nem" >
209 nő </td>
210 <td class="views-field views-field-field-elhunytak-kor" >
211 84 </td>
212 <td class="views-field views-field-field-elhunytak-alapbetegsegek" >
213 krónikus veseelégtelenség </td>
214 </tr>

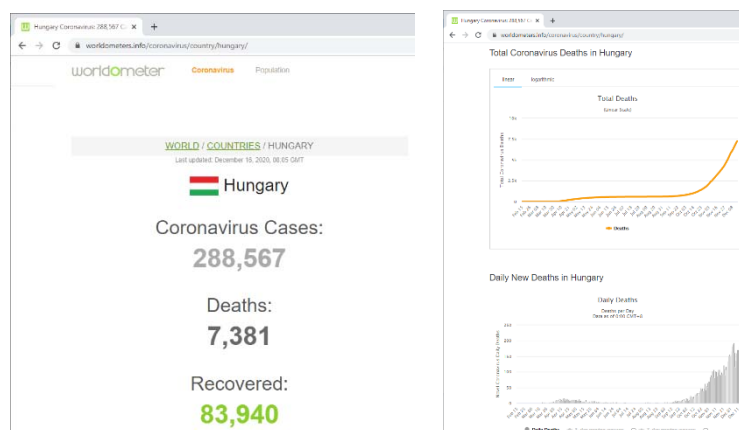
```

2. ábra. Magyarországi adatforrás, koronavírus input adatok, HTML, CSS forráskód

Adódtak durva hibák is, például: hiányzó nem, valamint nőnemű elhunyt prosztatarákkal az alapbetegségei között. Az alapbetegségek egyetlen mezőben szerepelnek, ami adatbázis analógiával denormalizált.

## 2.2. Nemzetközi adatforrás

A nemzetközi adatforrás weboldala a <https://www.worldometers.info/coronavirus/country/hungary/>. Itt összesített adatok találhatóak: összes koronavírus fertőzött, összes halott, összes gyógyult [15]. Az adatok grafikonokon jelennek meg (3. ábra).



3. ábra. Nemzetközi adatforrás, koronavírus input adatok, webes nézet

A weboldal forráskódjából ki kell olvasni a grafikonok megjelenítéséhez beágyazott, JavaScript adatsorokat (`xAxis.categories` és `series.data`), amelyek azonos indexszel összeköthetők (4. ábra).

```

9174 <div class="row_graph_row">
9175 <div class="col-md-12">
9176 <h3>Daily New Deaths in Hungary</h3>
9177 <div id="graph-deaths-daily"></div>
9178 <script type="text/javascript">
9179 Highcharts.chart("graph-deaths-daily", {
9180   chart: {
9181     type: 'column'
9182   },
9183   title: {
9184     text: 'Daily Deaths'
9185   },
9186   subtitle: {
9187     text: 'Deaths per Day<br>Data as of 0:00 GMT+8'
9188   },
9189   xAxis: {
9190     categories: ["Feb 15", "Feb 16", "Feb 17", "Feb 18", "Feb 19", "Feb 20", "Feb 21", "Feb 22",
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200
9201
9202
9203
9204
9205
9206
9207
9208   series: [{
9209     name: 'Daily Deaths',
9210     color: '#999',
9211     lineWidth: 4,
9212     showCheckbox: false,
9213     data: [null,null,null,null,null,null,null,null,null,null,null,null,null,null,
9214     {
9215       name: '3-day moving average',

```

4. ábra. Nemzetközi adatforrás, koronavírus input adatok, HTML, CSS, JavaScript forráskód

Így adott dátum(ok)ra megkaphatjuk az elhunytak számát.

## 2.2. Adatforrások összefésülése

A magyarországi adatforrásból érdemes weboldalanként ötvenesével menteni az adatokat. Ez weboldalanként 32-37 kB méretű. A nemzetközi adatforrás egyetlen fájlba menthető, ami kb. 76 kB méretű.

koronavirus-elhunytak-page064	html	37 372	20.11.21 23:15	-a--
koronavirus-elhunytak-page065	html	37 314	20.11.21 22:31	-a--
koronavirus-elhunytak-page066	html	36 920	20.11.21 22:49	-a--
koronavirus-elhunytak-page067	html	37 049	20.11.21 23:03	-a--
koronavirus-elhunytak-page068	html	36 856	20.11.21 23:07	-a--
koronavirus-elhunytak-page069	html	36 908	20.11.21 23:12	-a--
koronavirus-elhunytak-page070	html	37 653	20.11.21 22:39	-a--
koronavirus-elhunytak-page071	html	36 931	20.11.21 22:45	-a--
koronavirus-elhunytak-page072	html	37 008	20.11.21 23:02	-a--
koronavirus-elhunytak-page073	html	30 961	20.11.21 22:24	-a--
worldometers-coronavirus-hungary	html	77 627	20.11.21 20:41	-a--

0 k / 2 753 k - 0 / 75 fájl

5. ábra. Webről mentett adatok HTML formátumban

Ezután a már helyi állományrendszerben elérhető adatok (5. ábra) feldolgozásával előállíthatók átmeneti CSV fájlok. Ezekben könnyen, táblázatos formában áttekinthetőek az adatok Jegyzetomb vagy Excel segítségével (6. ábra).

A	B	C	D
1	Azonosító	Nem	Eletkor
2	3689	Férfi	56
3	3688	Férfi	72
4	3687	Nő	83
5	3686	Nő	88
6	3685	Nő	86
7	3684	Nő	82
8	3683	Férfi	68
9	3682	Nő	68
10	3681	Nő	82
11	3680	Nő	69
12	3679	Nő	84
13	3678	Nő	78
14	3677	Nő	78
15	3676	Nő	75
16	3675	Nő	86

A	B
1	Datum
2	2020.02.15 null
3	2020.02.16 null
31	2020.03.15 null
32	2020.03.16 0
33	2020.03.17 0
34	2020.03.18 0
35	2020.03.19 0
36	2020.03.20 3
262	2020.11.01 69
263	2020.11.02 70
264	2020.11.03 84
265	2020.11.04 90
266	2020.11.05 84
267	2020.11.06 103
268	2020.11.07 107
269	2020.11.08 81
270	2020.11.09 55
271	2020.11.10 103

6. ábra. Webről mentett adatok CSV formátumban

A két webes adatforrásból [14, 15] származó adatokat egyesíteni, párosítani, összekötni, másképpen összefésülni kell. Sorrendjük alapvetően különböző, ezt figyelembe kell venni. Ezután már 5 db összetartozó adat áll rendelkezésre a további elemzéshez/feldolgozáshoz: dátum, sorszám, nem, kor, alapbetegségek (7. ábra).

A	B	C	D	E
1	Datum	Azonosító	Nem	Eletkor
2	2020.03.20	1	Nő	76
3	2020.03.20	2	Nő	65
4	2020.03.20	3	Férfi	74
5	2020.03.22	4	Férfi	79
6	2020.03.22	5	Férfi	68
7	2020.03.23	6	Nő	41
8	2020.03.23	7	Férfi	38
9	2020.03.24	8	Férfi	53
10	2020.03.25	9	Nő	90
11	2020.03.28	10	Férfi	75
12	2020.03.29	11	Férfi	80
13	2020.03.29	12	Férfi	73
14	2020.03.30	13	Nő	86
15	2020.03.30	14	Férfi	92
16	2020.03.31	15	Férfi	94
17	2020.04.01	16	Nő	59
18	2020.04.01	17	Nő	86
19	2020.04.01	18	Nő	67
20	2020.04.01	19	Férfi	66
21	2020.04.02	20	Férfi	91

7. ábra. Magyarországi és nemzetközi adatforrásokból összefésült koronavírus adatok

Nem ismerjük a két adatforrásként használt weboldal frissítési időpontjait. Ezért előfordulhat, hogy a legújabb adatokat nem tudjuk összefésülni. Például tudjuk, hogy a nemzetközi adatforrás legutolsó dátuma esetén, hogy mennyi az esetszám, de már „elfogytak” a magyarországi adatok, azaz nincs mivel összekötni. Erre érdemes felkészülni egy kiegészítő, hiánypótló, inkrementális jellegű adatgyűjtés megvalósításával.

A további adatfeldolgozás előtt ebben a fázisban következhet egy kézi vagy gépi adattisztítás. A kézi adattisztítást csak elrettentő példaként érdemes bemutatni, érzékelte az óriási problémát. A projektet megvalósító diákok, hallgatók felkészültségétől és szakjától (OKJ, tanfolyam, BSc, MSc) függően említhető vagy kiadható részfeladat lehet különböző szövegfeldolgozó, klaszterizáló, statisztikai jellegű adatfeldolgozás, tanuló algoritmusok testre szabásával, több körben elvégzendő gépi adattisztítás.

## 2.4. A Java projekt tervezése

Az összefésült adatok rendelkezésre állnak CSV formátumban. Ebből a fejléc kihagyásával, soronként egy-egy objektumot kell készíteni. A POJO tartalmazza az összetartozó 5 db adatot egész típusú számként és szöveges formátumban, és kiegészül néhány konverziós függvénnyel (például dátumformátum) és a denormalizált alapbetegségekből előállít azokból egy generikus listát, amellyel hatékonyan támogatja a további adatfeldolgozást. Érdeemes privát konstruktorral létrehozni az osztályt és statikus Factory metódusra bízni a külső adatok és belső adattagok, valamint a generikus lista közötti leképezést/megfeleltetést. Menet közbeni teszteléshez hasznos – amíg nincs GUI és/vagy grafikon – ha `toString()` metódus is készül. Ez az `OsszesAdat` osztály. A `Modell` réteg/osztály szintén generikus adatszerkezettel működik, amely `OsszesAdat` típusú és a neve `osszesAdatLista`. Tehát rendelkezésre áll az a beépített funkció, amely az összefésült CSV-ből generikus POJO listát állít elő [16].

Igény szerint [9] építhető – egy-egy konkrét részfeladat követelményei alapján meghatározva – további lista, halmaz, leképezés, sor adatszerkezet, például: `List`, `ArrayList`, `HashSet`, `TreeSet`, `HashMap`, `TreeMap`. Ezek hagyományos utasításszerkezettel is, de akár funkcionális lambda kifejezésekkel is (Stream API alkalmazásával) is előállíthatók.

Az adatszerkezetek feldolgozásához bármelyik elemi és összetett programozási tétel felhasználható és ezeken kívül hasznos még a kétszintű csoportváltás algoritmus ismerete és implementálása (Java nyelvtől függetlenül és konkrét adatszerkezettől függően).

Elvárás lehet, hogy minimalizáljuk a konkrét függőségeket, alkalmazzunk általános megközelítést, valamint készüljünk fel a jövőre (ésszerű határok között, amennyire lehet). Előbbiekre példa: nem minden napon van adat és/vagy elhunyt. Utóbbira példa: a nemzetközi adatforrásban 2020. február 15-től kerültek be adatok, „Feb 15”, „Feb 16”... címkékkel. 2020 novemberében (amikor a projekttel először foglalkoztunk) már sejthető volt, hogy előbb-utóbb szükséges lesz (évváltáskor, vagy legkésőbb „Feb 14” után) az év szerepeltetése is a JavaScript adatsorban. Lehet, hogy ez visszamenőleg is maga után von majd egy módosítást, vagy csupán a jövőbeli adatoknál lesz kiegészítés. Mindenképpen érdemes lenne egységesen kezelni a dátumot. Időközben ez megoldódott. Nem rajtunk múlt a döntés, hiszen csupán alkalmazkodunk ahhoz, hogy milyen adatokat érünk el publikusan weboldalakon.

Az adatok mentése a magyarországi adatforrás esetében a weboldal esetében atipikus terhelést jelenthet, mert az ötvenesével publikált összes adatot lekérheti (weboldalanként külön-külön). Ezért érdemes beépíteni néhány megelőző szempontot, például: az elkészült program csak 22 és 5 óra között éjszaka futhat, egy-egy weboldal lekérését követően legyen 10-20-30 másodpercnyi várakozás, biztonságos SSL kapcsolat felépítése szükséges, UTF 8 kódolást kell feldolgozni, hasznos előzetesen engedélyt kérni és egyeztetni a User-Agent szövegét. Ha inkrementális mentést építünk be a Java projektbe, akkor azt feltételezzük, hogy múltbeli adatok már nem fognak megváltozni a jövőben (erre mégis találtunk ellenpéldákat).

Meg kell ismerkedni a specifikációban meghatározott kétféle (JFreeChart és Google Chart) grafikon elkészítéséhez szükséges osztálykönyvtárral, adatformátumokkal [11-13].

A JFreeChart típusú grafikonok önálló MVC mintát követő objektumok. Saját felelősségük adataik tárolása, kezelése (modell), kirajzolása, megjelenítése egy téglalap alakú területen (nézet) és a fókusz kezelése kiválasztáshoz, egérműveletek megvalósítása, eseménykezelés (vezérlő). Tipikus adatszerkezeteik az `org.jfree.data.category` csomagból, például: `CategoryDataset` és `DefaultCategoryDataset`, illetve az `org.jfree.data.general` csomagból: `DefaultKeyedValues2DDataset`, `DefaultPieDataset`.

A Google Chart típusú grafikonok is MVC mintát követnek. Külön kezelik saját adataikat (többnyire `DataTable` típust használnak (modell), azok megjelenítését befolyásoló beállításait (jelmagyarázatok, feliratok, színek, méretek, JS és CSS vegyesen (nézet), illetve eseménykezelést (oszlopdiagramra fókuszálva részletes adatok jelenhetnek meg) is biztosítanak. Érdeemes előkészíteni egy szöveges sablont, amely a fix/konstans adatokat, beállításokat tartalmazza, ezt betölteni fájlból, előállítani Java nyelven a szükséges adatokat JS kompatibilis módon, majd kicserélni (pl.: `##GRAFIKON_ADAT##`) a sablonban, amit kell és menteni HTML formátumban az elkészült grafikont beágyazva tartalmazó weboldalra.

A továbbiak előtt érdemes tanulmányozni egy részletes elemzést, szakterületi publikációt [17].

## 2.5. Csoportmunkában megvalósítandó projektfeladatok

A Java projekt megvalósítása során elérkezik a megvalósítás során az a kritikus tömegű beépített funkcionalitás, ami után kijelölhetőek a befejezéshez szükséges, csoportmunkában megvalósuló projektfeladatok. Ezek a korábban, közösen összefésült CSV adatokat generikus POJO listába leképező, modell rétegre épülő alap swing GUI nézet réteg és egy-egy JFreeChart és Google Chart minta megtervezésével záruló fázisra építve a következők voltak/lehetnek.

Megvalósítandó JFreeChart grafikonok:

- Korfa diagram: az elhunytak csoportosítása korcsoportonként és nemenként,
- Kördiagram: alapbetegségek, amelyek legalább az esetek 2%-ában előfordulnak,
- Álló oszlopdiaagram: elhunytak száma hetenként,
- Fekvő oszlopdiaagram: szívprobléma alapbetegségek előfordulása (benne van: „szív”, „pitvar”, „műbillentyű”, „pacemaker”, kis- és nagybetű nem számít),
- Vonaldiagram: elhunytak száma havonta és nemenként,
- Idősor diagram: elhunytak száma havonta és nemenként,
- Kördiagram: TOP 5 leggyakrabban előforduló alapbetegség nemenként (2 diagram egymás mellett),
- Fekvő oszlopdiaagram: TOP 3 alapbetegség havonta.

Megvalósítandó Google Chart grafikonok:

- Timeline: elhunytak száma havonta,
- Bar Chart: daganatos betegségek előfordulása (benne van: „tumor”, „daganat”, „rák”, „áttét”),
- Pie Chart: elhunytak alapbetegségeinek száma,
- Combo Chart: TOP 5 szívbetegséghez és TOP 5 cukorbetegséghez köthető alapbetegség,
- Line Chart: elhunytak száma hetente, nemenként,
- Column Chart: elhunytak száma havonta, nemenként,
- Line Interval: TOP 5 leggyakrabban előforduló alapbetegség havonta, nemenként,
- Calendar Chart: elhunytak száma hetente.

## 3. MEGVALÓSÍTÁS

A cikk terjedelmi korlátai miatt csupán az elosztott hálózati funkció Java nyelvű megvalósítását (forráskódját) mutatja be a szerző, amely a magyarországi adatforrásból végzi el az adatok mentését.

```

284 private void webrolMentAazonositoNemEletkorAlapbetegsegek() throws IOException {
285     //csak 22 óra után és 6 óra előtt futtatható
286     int aktOra=new GregorianCalendar().get(Calendar.HOUR_OF_DAY);
287     if(6<aktOra && aktOra<22) //percet nem vesz figyelembe
288         return;
289     //start url: https://koronavirus.gov.hu/elhunytak
290     //továbbiak: https://koronavirus.gov.hu/elhunytak?page=1
291     //...
292     // https://koronavirus.gov.hu/elhunytak?page=73
293     //2020.11.21. 22:00-kor
294     List<String> urlLista=new ArrayList<>();
295     urlLista.add("https://koronavirus.gov.hu/elhunytak");
296     for (int i = 1; i <= 73; i++) //lehetne általánosítani
297         urlLista.add("https://koronavirus.gov.hu/elhunytak?page="+i");
298     Collections.shuffle(urlLista);
299     try {
300         for (String url : urlLista) {
301             String mentettHtmlOldalNev=mentettHtmlOldalNev(url);
302             File htmlFajl=new File(mentettHtmlOldalNev(url));
303             webrolMentAazonositoNemEletkorAlapbetegsegek(url, htmlFajl);
304             System.out.println("Sikeresen mentett fájl: "+mentettHtmlOldalNev);
305             int varakozikMasodperc=(int) (Math.random()*31+30); //30-60 s
306             TimeUnit.SECONDS.sleep(varakozikMasodperc);
307         }

```

8. ábra. Részlet a Java forráskódból

A `webrolMent...()` metódusok hivatkoznak egymásra. Először a futtatás időbeli korlátozását kell megoldani, `aktOra` (8. ábra, 286. sor). Ezután az ötvenesével elérhető adatok között a „lapozást” konkrét értékig (73) megvalósítani (296. sor), vagy amíg a részeredmények feldolgozásával elő nem áll, hogy az 1. sorszámú elhunyt adatait is elértük (vagy amíg inkrementális mentésnél elegendőt lépkedtünk vissza a jelenből a múltba). A hálózati kapcsolat és a fájlkezelés miatt egyaránt kötelező a kivételkezelés alkalmazása (299. sor). A paraméterezéssel előállított `urlLista` adatszerkezeten való végig haladás (300. sor) során beépített várakozás is szerepel (`varakozikMasodperc`). Hasznos lehet összekeverni az `urlLista` elemeit (298. sor) véletlenszerűen.

Ezután weboldalanként külön-külön fel kell építeni a biztonságos és engedélyezett kapcsolatot a távoli adatforrással (9. ábra, 316. sor), az `sc` objektumot SSL paraméterrel inicializálva (317. sor). Ekkor végrehajtható a beállított `User-Agent`-en keresztül megfelelő böngészőt szimulálva/emulálva (336. sor) a távolról megkapott HTML tartalom mentése helyi fájlba, megfelelő (UTF8) karakterkódolással (341. sor).

```

314 private void webrolMentAzonositoNemEletkorAlapbetegsegek(
315     String url, File htmlFajl) throws IOException {
316     try {
317         SSLContext sc=SSLContext.getInstance("SSL");
318         sc.init(null, new TrustManager[] {
319             new X509TrustManager() {
320                 @Override
321                 public X509Certificate[] getAcceptedIssuers() { return null; }
322                 @Override
323                 public void checkClientTrusted(X509Certificate[] certs, String authType) {}
324                 @Override
325                 public void checkServerTrusted(X509Certificate[] certs, String authType) {}
326             }
327         }, new SecureRandom());
328         HttpURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
329     }

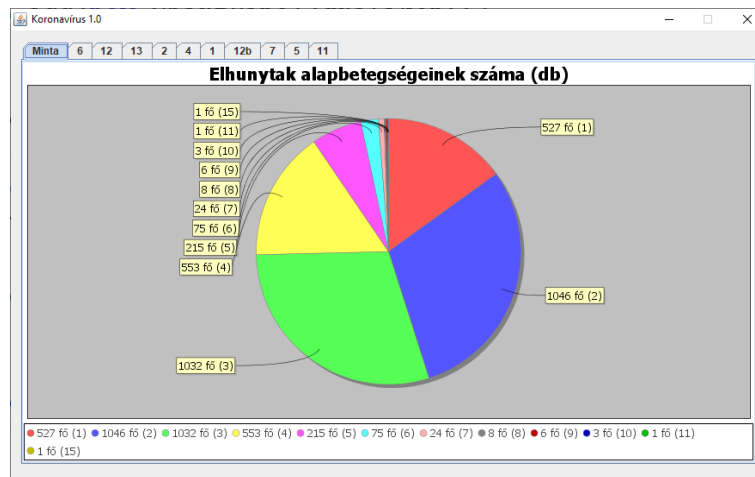
334     HttpURLConnection c=(HttpURLConnection) (new URL(url)).openConnection();
335     c.setRequestMethod("GET");
336     c.addRequestProperty("User-Agent",
337         "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) "+
338         "Chrome/51.0.2704.103 Safari/537.36");
339     c.connect();
340     InputStream is=c.getInputStream();
341     BufferedReader br=new BufferedReader(new InputStreamReader(is, "UTF-8"));
342     StringBuilder htmlTartalom=new StringBuilder();
343     String sor="";
344     while((sor=br.readLine())!=null)
345         htmlTartalom.append(sor).append("\n");
346     br.close();
347     is.close();
348     c.disconnect();
349     Files.write(Paths.get(htmlFajl.getPath()),
350         htmlTartalom.toString().getBytes(), StandardOpenOption.CREATE);
351 }

```

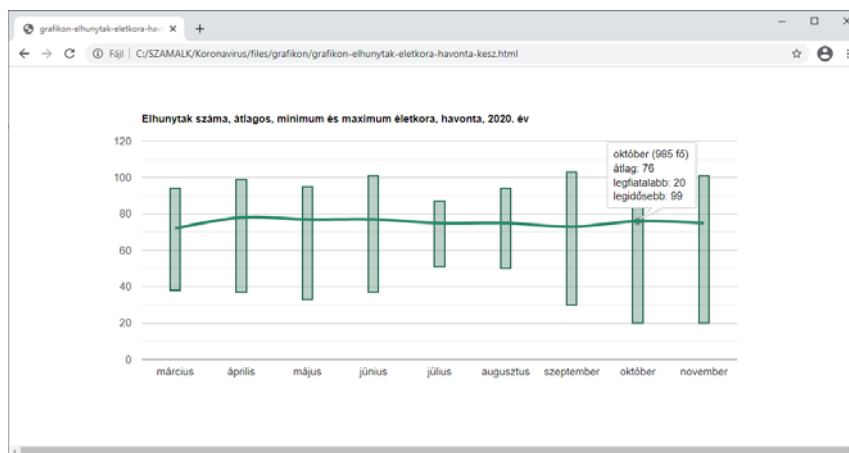
9. ábra. Részlet a Java forráskódból

## 4. EREDMÉNYÜL KAPOTT GRAFIKONOK

Az MVC-s Java projekt kétféle nézet réteget valósít meg. Az egyik nézet az asztali alkalmazások esetén használható swing GUI, amely beépített csomagként hivatkozik a JFreeChart osztálykönyvtárra és képes a grafikont kirajzolni egy téglalap alakú `JPanel` típusú panelre [10]. A másik nézet pedig önállóan generált teljes weboldalakat jelent, amelynek a böngésző a megjelenítője. Egy-egy példát láthatunk (10. és 11. ábra) ezek közül.

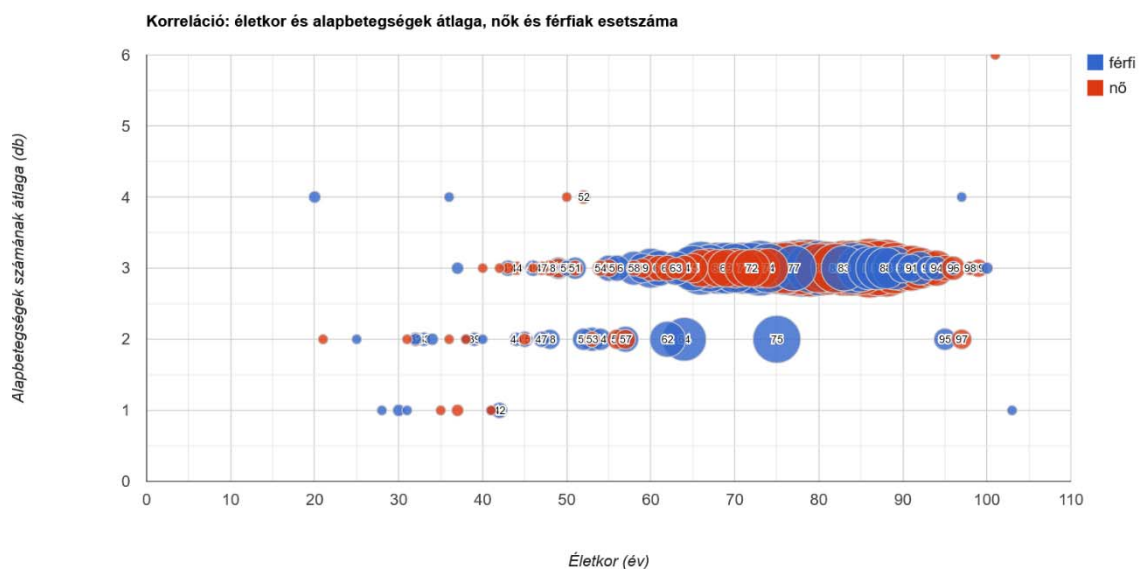


10. ábra. Eredményül kapott JFreeChart típusú grafikon (swing GUI) Elhunytak alapbetegségeinek száma (db)



11. ábra. Eredményül kapott Google Chart típusú grafikon (weboldal böngészőben) Elhunytak száma, átlagos, minimum és maximum életkora, havonta, 2020. év

A 2.3. pontban felsoroltak alapján, ha a projektmunkában résztvevők rendelkeznek elegendő matematikai, statisztikai ismeretekkel, akkor összetettebb feladatot is kaphatnak. Például (12. ábra):



12. ábra. Eredményül kapott Google Chart típusú grafikon (weboldal böngészőben) Korreláció: életkor és alapbetegségek számainak átlaga, nők és férfiak esetszáma



## KÖSZÖNETNYILVÁNÍTÁS

A szerző köszönetet mond minden közreműködőnek a projekt alkotói folyamatában történő részvételükért, a kitarató tesztelésért, észrevételeikért, akik:

- a SZÁMALK-Szalézi Technikum és Szakgimnázium másodéves OKJ szoftverfejlesztő szak tanulói, akik a Java alkalmazások fejlesztése tantárgyban kapták a projektet a szakképzés duális szakaszában (2020 ősszel), akik a 80%-ban elkészült projektet befejezték a sokféle projektfeladatot, grafikonokat készítettek csoportmunkában (a szak követelményei miatt szerepelnek magyar elnevezések a Java forráskódban),
- az it-tanfolyam.hu Java EE szoftverfejlesztő tanfolyam hallgatói, akik komplex, online vizsgafeladatként kapták [18] a projekt 50%-ban elkészült változatát és csoportmunkában, valós feladat részfeladatait modellezve (2021 tavasszal), különféle – programozáshoz kötődő – szakterületekkel foglalkozva befejezték a projektet, ezzel lezárva tanulmányaikat,
- az UH Erasmus+ Computer Science MSc hallgatói, akik a szakdolgozati konzultációjukra való felkészítő kurzusukban komplex fejlesztői feladatként kapták meg a projekt 20%-ban elkészült változatát (2020, 2021 és 2022 ősszel) és többféle szövegfeldolgozó és tanuló algoritmust paraméterezve adattisztítást végeztek.

## IRODALMI HIVATKOZÁSOK

- [1] Java SE szoftverfejlesztő tanfolyam, tematika, szakmai modul, orientáló modul, <https://it-tanfolyam.hu/java-se-szoftverfejleszto-tanfolyam/>, 2022.08.20.
- [2] Java EE szoftverfejlesztő tanfolyam, tematika, szakmai modul, orientáló modul, <https://it-tanfolyam.hu/java-ee-szoftverfejleszto-tanfolyam/>, 2022.08.20.
- [3] Java NIO vs. IO, <https://www.javatpoint.com/java-nio-vs-input-output>, 2020.10.07.
- [4] Java I/O, NIO, and NIO.2, <https://docs.oracle.com/javase/8/docs/technotes/guides/io/index.html>, 2020.10.07.
- [5] Java Code Examples for javax.net.ssl.SSLContext, <https://www.programcreek.com/java-api-examples/?api=javax.net.ssl.SSLContext>, 2020.10.20.
- [6] Introduction to Web Scraping With Java, <https://www.scrapingbee.com/blog/introduction-to-web-scraping-with-java/>, 2020.10.07.
- [7] Reading Directly from a URL, <https://docs.oracle.com/javase/tutorial/networking/urls/readingURL.html>, 2020.10.07.
- [8] Java Swing MVC Example, <https://examples.javacodegeeks.com/core-java/java-swing-mvc-example/>, 2020.10.16.
- [9] Algorithms Complexity and Efficiency of Data Structures, <https://www.slideshare.net/introprogramming/19-algorithms-andcomplexity>, 2022.05.01.
- [10] How to Use Tabbed Panes, <https://docs.oracle.com/javase/tutorial/uiswing/components/tabbedpane.html>, 2020.10.16.
- [11] JFreeChart Samples, <https://www.jfree.org/jfreechart/samples.html>, 2021.07.20.
- [12] JFreeChart Tutorial, <https://www.tutorialspoint.com/jfreechart/index.htm>, 2021.07.20.
- [13] Google Chart Gallery, <https://developers.google.com/chart/interactive/docs/gallery>, 2021.07.20.
- [14] Koronavírus (magyarországi adatforrásként használt weboldal), [https://koronavirus.gov.hu/elhunytak\\_](https://koronavirus.gov.hu/elhunytak_), 2022.08.20.
- [15] Hungary COVID - Coronavirus Statistics – Worldometer (nemzetközi adatforrásként használt weboldal), <https://www.worldometers.info/coronavirus/country/hungary/>, 2022.08.20.
- [16] Generics Methods (The Java™ Tutorials), <https://docs.oracle.com/javase/tutorial/java/generics/methods.html>, 2022.05.01.
- [17] Ferenci, Tamás: A magyarországi regisztrált koronavírusos halottak adatainak elemzése, <https://github.com/tamas-ferenci/C19MortalityLineListingHUN>, 2021.10.03.
- [18] Hogyan értékeljük az online vizsgafeladatot?, <https://it-tanfolyam.hu/hogyan-ertekeljuk-az-online-vizsgafeladatot/>, 2022.08.20.
- [19] Erasmus+ program – Szakdolgozatok – kaczursandor.hu, <https://kaczursandor.hu/erasmus-program-szakdolgozatok/>, 2022.08.20.