

Jarvis

The Avengers

Bálint Máté

Felkészítő tanár: Urbánné Jung Andrea

*Szegedi Tömörkény István Gimnázium, Művészeti Szakgimnázium és Technikum
6720 Szeged, Tömörkény utca 1.*

1. Bevezetés

Jarvis egy Javában írt virtuális asszisztens. Egy virtuális asszisztenssel szemben alapvető követelmény, hogy élő szóban adhassunk neki utasítást, amit a program megért, és reagálni is tud rá. Általában olyan feladatokra használják, amelyeket az emberek gyakran akarnak ismételni, de rendszeres információszerzésre is alkalmas. Az én virtuális asszisztensem az alábbi feladatkörökre képes: idő- és dátuminformációk megszerzése, emlékeztetés, zenelejátszás, időjárás-jelentés illetve filmajánlás.

2. Probléma megoldásának menete

2.1. Input, output

Az első és legfontosabb dolog az volt, hogy a mikrofon inputot Stringgé konvertáljam. Ehhez a sphinx4 könyvtárat használtam. A main metódusban egy végtelen loopot indítok, hogy a hangot rögzíthessem, majd ezt a voiceCommand Stringben tárolom. Innentől kezdve rengetek if, else szerkezettel döntöm el, hogy mit is csináljon a program.

A második legfontosabb egy virtuális asszisztens esetén, hogy válaszolni tudjon. Mivel a program eredetileg privát felhasználásra készült így nem TTS API-t használok, hanem csak simán audiófájlokat. A hangfájlok az eredeti IOS mobilapplikációból vannak, de némelyik fájl nevét megváltoztattam, az egyszerűbb használat érdekében (pl. amikor azt kérdezem, hogy milyen nap van ma). A változatosság kedvéért Jarvis nem mindig ugyan úgy válaszol, véletlenszerűen dől el a megerősítés.

2.2. Idő

A program parancsai közül ötnek köze van az időhöz. Három parancs a jelennel kapcsolatos, míg a másik kettő egyfajta emlékeztetőként szolgál. Az első három megoldásához Calendar osztályt használva szerzem meg a parancsnak megfelelő adatokat, majd lejátszom az adatoknak megfelelő hangfájlokat. Mivel a „számos” hangfájlok nullától kilencig úgy kezdődnek, hogy nulla plusz a szám, de a Calendar osztály úgy adja vissza, hogy csak szám, így ellenőriznem kell a Calendar-os String hosszát, hogy a jó audiót játszhasam le.

A triggered metódus a reminder parancs szíve, a felhasználó inputját felosztja a kettőspontoknál, majd kiszámolja, hogy az idő amit beírtunk

összesen hány másodperc. Ezután a Timer és a TimerTask osztály segítségével a kiszámolt idő ezerszeresével indítok egy időzítőt. (Ugyanis a schedule metódus ezredmásodpercet vár paraméterként.) Amikor az időzítő lejár a ReminderClass run metódusa kerül végrehajtásra. Az alarm parancs nagyon hasonló a reminder-höz. A megfelelő másodpercet úgy kapom meg, hogy a várni kívánt időpont másodpercre konvertált értékéből kivonom a jelenlegi időpont másodpercre konvertált értékét.

2.3. Internet

A WebReader osztályt arra használom, hogy oldalak forráskódját olvassam be, illetve hogy ellenőrizzem azt, hogy van-e internetje a felhasználónak. Amennyiben nincs, Jarvis szólni fog, majd kilép. Itt található még a giveTextBetween metódus is, amely az 1. ábrán látható (plusz kétszer felülírva). Az alapkoncepció az, hogy a source Stringben megkeressük a before String első előfordulását, majd ehhez hozzáadjuk a before String hosszát. Ettől az integer értéktől kezdjük el keresni az aftert a source-ban. Ez az end integer. Ezután visszaküldjük a start és az end közötti szöveget, úgy, hogy a végén és elején lévő felesleges space karaktereket is levágjuk a trim metódus segítségével.

```
public static String giveTextBetween(String source, String before, String after) {
    int start = source.indexOf(before);
    start += before.length();
    int end = source.indexOf(after, start);
    return source.substring(start, end).trim();
}
```

1. ábra: A giveTextBetween metódus

2.4. Zenelejátszás

A play vagy a music parancsal zenét hallgathatunk. Mivel a music paramétert elég, hogy tartalmazza (a jarissal együtt persze) így akár azt is mondhatom, hogy „jarvis, i want to listen to music” akkor is aktiválódni fog. Valójában a parancs nem ez lesz, mivel a beállításoknál megadott szótár fájl nem tartalmazza csak a „jarvis” és a „music” szót a mondatból, de azt a kettőt fel fogja ismerni, tehát aktiválódni fog. A PlayGUI osztályt az AWT Frame osztályból származtatom és kiegészítem az ActionListener és a KeyListener osztállyal, hogy a kattintást és az enter-t is támogassam. Osztályváltozóként létrehozom a GUI-hoz szükséges objektumokat, változókat, majd a konstruktorban be is állítom őket. A playSong metódus felel a link megszerzéséért és megnyitásáért, ezt a gomb kattintásával, vagy az enter lenyomásával érhetjük el. A textfieldben lévő Stringben kicserélem a szóközt egy plusz jelre, mivel a YouTube azt használja szóközként. Ezután a source Stringben tárolom az oldal forráskódját, ebben megkeresem a „/watch?v=” és az első idézőjel közti Stringet. Ez a videó azonosítója, miután ez megvan, megnyitom a linket a böngészővel.

2.5. Időjárás

Jarvissal lekérhetjük az aznapi Szegedi időjárás jelentést is. Az adatokat az időkép honlapjáról szerezem meg a WebReader osztály segítségével. Először is el kell döntenünk, hogy mínusz fok van-e vagy sem, ezt úgy tehetjük meg egyszerűen, hogy megnézzük, hogy a hőmérsékletet tartalmazó String tartalmazza-e a „-” jelet. Ezután abban az esetben, ha szeles az idő, azt is elmondja Jarvis, majd végül a jelenlegi időjárásra kerül a sor. Erre azért van szükség külön lépésben, mert a napos idő is lehet szeles, ugyanúgy mint az esős, ezért már az oldal is külön vette ezt a két dolgot.

2.6. Filmajánló

A movies paranccsal megtekinthetjük a legújabb és legnézettebb filmek listáját. Az időjáráshoz hasonlóan itt is a WebReadert használom. Az adatokat egy kétdimenziós String tömbben tárolom, amelynek az első dimenziójában annyi elem lehet, amennyi film van éppen az ajánlottak között. Ezt a számot Regex-el nyерem ki, és ez lesz az a szám ami hússzal megszorozva a Frame dinamikus magasságát adja. A második dimenzióba a három kerül, mivel egy filmhez három információt akarok tenni. A while loopban megkezdődik az adatkinyerés és feldolgozás folyamata. Ha az indexek kisebbek nullánál, az azt jelenti, hogy elfogytak az Stringek a sourceban. A filmcím és a nyelvhez a második ábrán látható giveTextBetween-t használom, a minőséghez pedig a harmadik ábrán láthatót. Ebben a metódusban azért használok +2-t mert a nyelv adat vége egy záró kacsacsőrrel és egy idézőjellel végződik, ami kettő karakter. Ezután a két karakter után már jön is a minőség. A loop végén újítom az index változókat úgy, hogy az előző index + 1-től újra elkezdem keresni a keresendő Stringeket. A +1-gyel előzőm meg azt, hogy önmagát érzékelye újra. Miután a loop feldolgozta az adatokat és belepakolta a tömbbe, a JTable osztály segítségével megjelenítem azt a felhasználónak.

```
public static String giveTextBetween(String source, String before, String after, int index) {
    int start = index;
    start += before.length();
    int end = source.indexOf(after, start);
    return source.substring(start, end).trim();
}
```

2. ábra: A felülírt giveTextBetween metódus

```
public static String giveTextBetween(String source, int start, String after) {
    start += 2;
    int end = source.indexOf(after, start);
    return source.substring(start, end).trim();
}
```

3. ábra: A másodjára felülírt giveTextBetween metódus

3. Elért eredmények

Jarvis volt az első projektem ezáltal rengeteget fejlődtem az írása közben. Első sorban a Stringekkel dolgoztam, így volt időm elsajátítani eme karaktertömböknek a metódusgyűjteményét és használatát. Ezzel párhuzamosan elsajátítottam az objektumorientált programozás alapjait. A következő egyik leghasznosabb tudás amit szereztem nem más mint az internetprogramozás. A legelső verziókban a JSoup HTML parsert használtam, majd szépen lassan áttértem a JDK-ban található osztályokra és a manuális adatszűrésre. Pár olyan tudást is elsajátítottam, amely nem látszik a programban. Ilyen például a többszálú programozás. Az emlékeztetőkhöz eleinte a Runnable interfészt használtam, de végül találtam egy jobb megoldást a TimerTask osztály segítségével, így téve Jarvis stabilabbá. Talán a legfontosabb ismeret amit szereztem az nem más mint a git. Mivel a projekt kezdett egyre nagyobb lenni, és szerettem volna új dolgokkal kísérletezni, így elkerülhetetlen volt egy verziókezelő megismerése és elsajátítása. Jarvissal rész vettem még a 17. Neumann, Nemzetközi Tehetségkutató Programtermék Versenyen. Azóta több hibát is kijavítottam, több ellenőrzés van és a hőmérséklet bemondásán kívül a teljes időjárás jelentés is támogatott. Jarvis és a forráskódja az alábbi linken érhető el¹.

¹ <https://drive.google.com/file/d/1dpzPqq6RAFGsPcs3QMKmf4XVgvPXO/view?usp=sharing>