

Szólánc-GUI-s változat



v1.5-t készítette: 3. csoport - Zágonyi Péter

v2.0-t készítette: 1. csoport - Horváth Ditta, Ruzska
Dorottya, Zsigmond Réka

V3.0 készítette: Zsigmond Réka

Programozás I.

Szoftverfejlesztő Esti 1.

Tartalomjegyzék

- Tervezés.....2.
- Feladatspecifikáció.....3.
- Folyamatábra.....5.
- Implementálás.....7.
- Tesztelés.....15.
- Fejlesztési lehetőségek.....17.
- Források, hivatkozások.....18.

Tervezés

Feladat:

- A program legyen képes fájlból szöveget beolvasni, majd az elemek neveiből objektumokat gyártani.
- Bekérni valamelyik lehetséges elemet a felhasználótól.
- Kitörölni a lehetséges elemek közül a már kiválasztottakat, és felkínálni a választott elem utolsó betűjével kezdődő, még játékban lévő elemeket a következő játékos részére.
- Számolni kell a lejátszott fordulók számát.
- Amikor elfogytak a választható elemek, ki kell írattni a győztest, és a lejátszott fordulók mennyiségét.
- Platform: NetBeans IDE 8.2

Feladatspecifikáció

Analízis

Megoldandó lépések:

- Egy metódus, ami képes fájlból szöveget beolvasni, levágni a felesleges részeket és objektumokat gyártani a fennmaradó hasznos részből.
- Szükségünk van egy olyan metódusra, ami előkészíti a játékot a kezdésre (kiíratja a kezdeti, teljes listát, létrehozza a *kontrollLista*-t és elindítja magát a játékot).
- Kell egy metódus ami levezényli magát a játékot.
- Speciális karakterek, ékezetek kezelése az inputban
- Magyar elemek használata

Feladatspecifikáció

Analízis

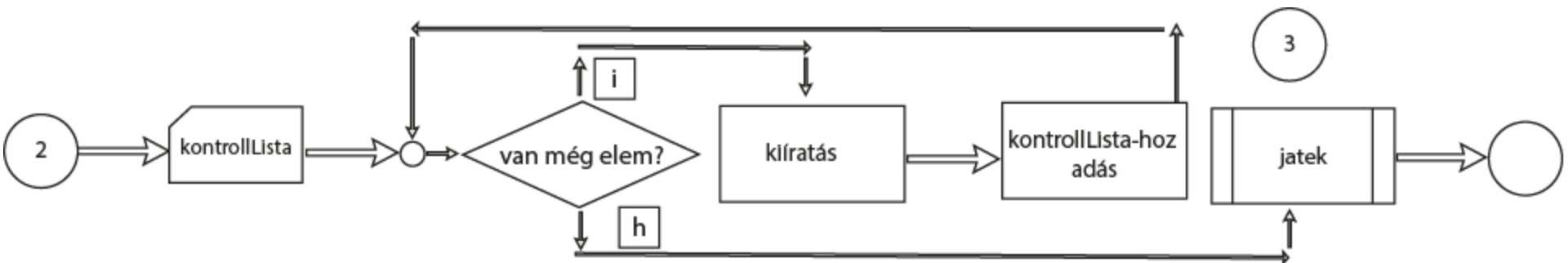
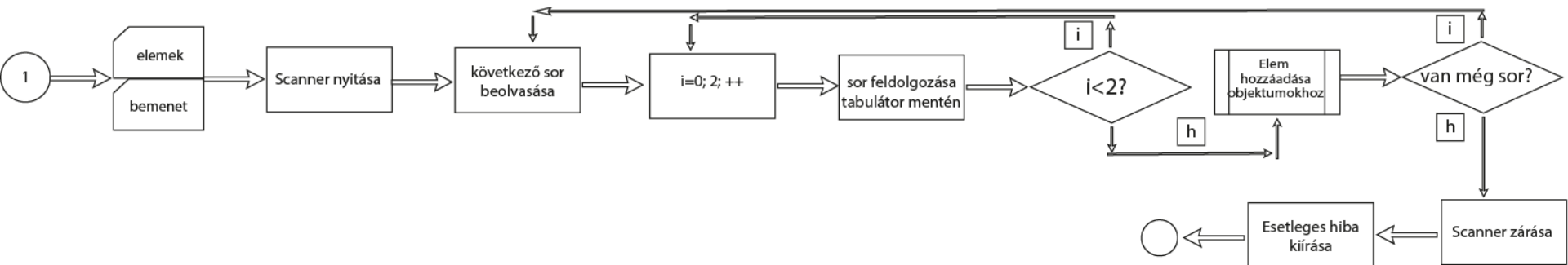
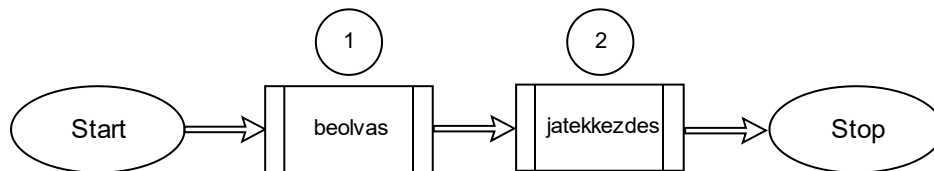
Szükséges adatszerkezetek :

- Egy osztály amiben eltároljuk a beolvasott elemeket, azoknak fontosabb adatait (Magát az elem nevét, első- illetve utolsó betűjét)
- Egy lista amiben az összes még használható objektum szerepel.
- Egy másik lista amiben az éppen felhasználható elemek szerepelnek.

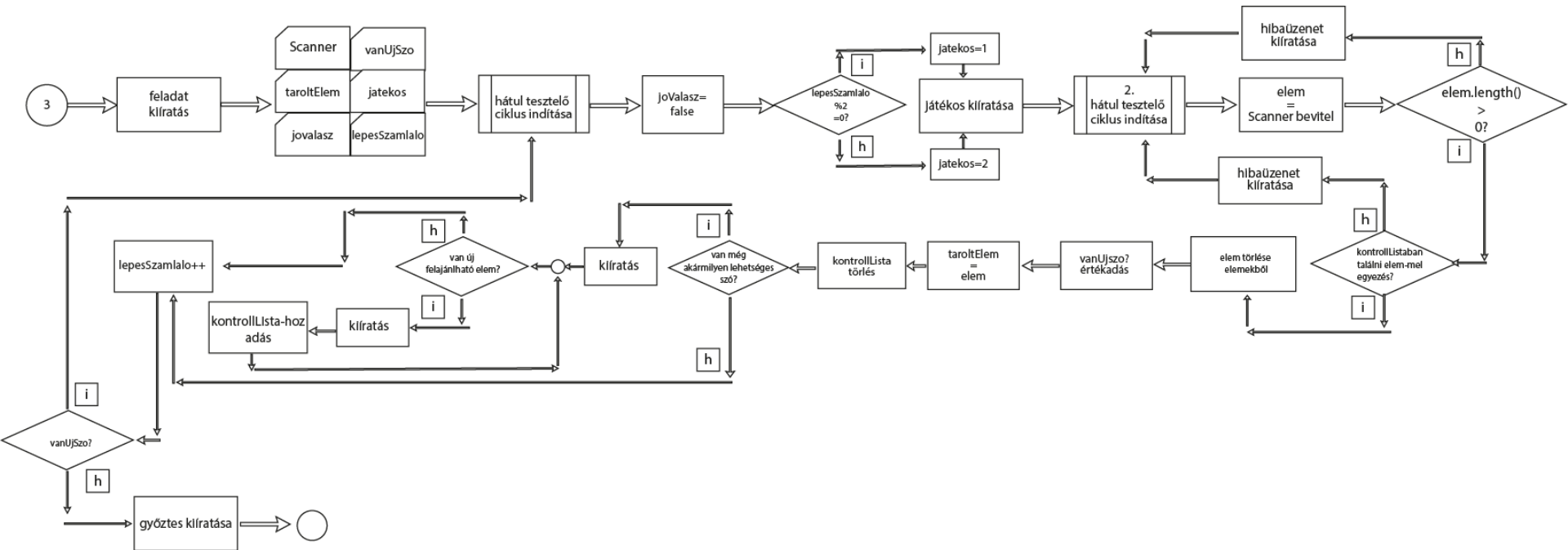
és változók:

- Kell egy int típusú változó, amin a fordulók számát tároljuk.
- Egy String illetve egy-egy char változó az objektumokon belül, a lényegi adatoknak(név, első-, utolsó betű).
- Egy int típusú változó a játékosokkal kapcsolatos kiírások egyszerűsítése érdekében (az aktuális játékost jelöli).
- Egy – egy boolean típusú változó a bevitt szó ellenőrzésére, illetve a játék végének meghatározására.
- Egy String típusú változó a bevitt elem hosszabb tárolása céljából, illetve későbbi vizsgálatokhoz (.stream()) metódus miatt az eredeti beviteli String felhasználása korlátozott).

Folyamatábra



Folyamatábra



Implementálás (Kódolás)

POJO:

Képes kezelni a speciális karaktereket

```
11 class Elemek {
12
13     private char elsoBetu;
14     private char utolsoBetu;
15     private String elem;
16
17     public Elemek(String elem) {
18         this.elsoBetu = elem.charAt(0);
19         this.utolsoBetu = elem.charAt(elem.length() - 1);
20         this.elem = elem;          //helyes névhasználat
21     }
22
23
24     public String getElemek() {
25         return Normalizer.normalize(elem.toLowerCase(), Normalizer.Form.NFD)
26             .replaceAll("\\p{InCombiningDiacriticalMarks}+", "").replaceAll("[^a-z]", "");
27     }          //magyar elemek kezelése
28
29     public char getUtolsoBetu() {
30         return utolsoBetu;
31     }
32
33     public char getElsoBetu() {
34         return elsoBetu;
35     }
36
37     @Override
38     public String toString() {
39         return elem;
40     }
41 }
```


GUI-s változat

- MVC modell használat
- Swinges és awt elemek használata
- Magyar elemek használata
- Cél: a feladatok egyértelmű kiszervezése

Implementálás (Kódolás)

Nézet kialakítása

```
public class NezetSheldon extends JFrame {  
  
    private ModellSheldon modell = null;  
    private MainSheldon vezerlo = null;  
  
    private JLabel teteje = new JLabel(" ");  
    private JPanel pnSheldon = new JPanel();  
    private JPanel listaPanel=new JPanel();  
    public JButton jatekKezdGomb = new JButton();  
    //    //majd kereteket csinalni//  
    public Border border = BorderFactory.createEmptyBorder(10, 10, 10, 10);  
    public Border borderList = BorderFactory.createLineBorder(Color.GRAY, 1);  
}
```

Implementálás

nézet

```
private Border gombBorder = BorderFactory.createEmptyBorder(10,
0, 0, 0);
private Color szin = new Color(215,
220, 222);    // világosabb szürke legyen a szürke
private Color szin2 = new Color(228, 232, 235);
public DefaultListModel balLista =
new DefaultListModel();    //hogy dolgozni is tudjak vele a háttérbe
n, ne csak megjeleníteni
public DefaultListModel jobbLista = new DefaultListModel();
public JList elemekLista = new JList(balLista);
// public JLabel kivalasztottElemekLista = new JLabel();
public JPanel kivalasztottPanel=new JPanel();
public int valasztottIndex;
```

Implementálás

nézet

- `public NezetSheldon(ModellSheldon modell, MainSheldon vezerlo) {`
- `this.modell = modell;`
- `this.vezerlo = vezerlo;`
- `jobbLista=modell.getElemekLista();`
- `setTitle("Szólánc - kémiai elemek");`
- `setDefaultCloseOperation(EXIT_ON_CLOSE);`
- `setSize(350, 550);`
- `Font betu = new`
- `Font(Font.DIALOG, Font.BOLD, 14);` `//majd ezt beállítani`
- `setResizable(false);`
- `setLocationRelativeTo(this);`
- `setLayout(new BorderLayout());`
- `// setBackground(Color.red);`
- `add(pnSheldon);`
- `pnSheldon.add(teteje, BorderLayout.NORTH);`
- `pnSheldon.setBorder(border);`
- `pnSheldon.setBackground(szin);d text`

Implementálás(Kódolás)

JListat használok, mint fő adatszerkezetet, ami mögött DefaultListModel dolgozik. Ezekbe kerülnek bele az elemnevek az ELEMNEV nevezetű tömbből.

```
} ...
```

Implementálás (Kódolás)

A nézet felelős a megejelnitésért, az akciókezelő `ListSelectionListener` üzen a vezérlőnek, ami a `jatek()` metódust irányítja le.

A *taroltElem* funkciója, hogy a később bevitt elemet a saját ciklusán kívül is vizsgálni tudjuk,

Az ciklus addig futtatja újra és újra a játékot amíg van új válaszható szó.

Majd amikor már csak egy szó lehet a listában meghívja a `jatekStop()` metódust.

Itt üzenünk a játékosnak, hogy hány lépésben és ki nyert.

Implementálás (Kódolás)

```
95 do {
96     String beolvasottSzoveg = sc.nextLine().toLowerCase();
97     if (beolvasottSzoveg.length() > 0) {
98         String elem2 = Normalizer.normalize(beolvasottSzoveg.toLowerCase(), Normalizer.Form.NFD)
99             .replaceAll("\\p{InCombiningDiacriticalMarks}+", "").replaceAll("[^a-z]", ""); //magyar elemek kezelése
100         String elem=elem2;
101         jovalasz = kontrollLista.stream().anyMatch(e -> e.getElemek().equals(elem));
102         if (jovalasz && !(elem.equals(taroltElem))) { //hiba javítása
103             elemek.removeIf(e -> (e.getElemek().equals(elem)));
104             vanUjSzo = elemek.stream().anyMatch(e -> e.getElsoBetu() == elem.toLowerCase().charAt(elem.length() - 1));
105             taroltElem = elem;
106             kontrollLista.clear();
107         } else
108             System.out.println("Hibás elem, kérem ellenőrizze a válaszható elemek listáját!");
109     } else
110         System.out.println("Kérem írjon be választ, mielőtt továbblépne!");
111 } while (!jovalasz);
```

Egy Scanner() segítségével kérjük be a kiválasztott elemet - ezt kisbetűs, ékeztelenített, speciális karakterektől megszabadított szöveggé alakítjuk.

Ezt egy do-while ciklus fogja közre, amit további két for ciklus egészít ki, hogy elkerüljük a helytelen beviteleket.

A bevitt adat ellenőrzésében és a további lehetséges szavak vizsgálatában a .stream().anyMatch() függvény segít, ami biztosítja, hogyha bárhol talál egy egyezést a program a bevitt érték és a vizsgált feltétel között, akkor ne fusson tovább a keresés mint pl. egy foreach ciklus esetében, hanem egy boolean értékkel egyszerűen csak mutassa ki ezek meglétét vagy hiányát.

Továbbá jó válasz megadása esetén kitöröljük a választott elemet a választékok közül. Ezt a részt szintén egy do-while ciklus fogja közre, hogy csak megfelelő adatokkal léphessen tovább a program.

Fejlesztési lehetőségek

- A játékosok előre megadhatnák a neveiket
- Menükezelési lehetőség a játék közben
- Más objektumokat tartalmazó listák használata

Források:

- Iskolai gyakorlatok:

<http://kaczursandor.hu/SZKI/2019-2020-1/E-Gyakorlas.zip>

- Folyamatábra:

<http://www.adony.hu/wp-content/uploads/2011/10/INF-501a.pdf?>

- Stream.anyMatch() metódus:

<https://stackoverflow.com/questions/23308193/break-or-return-from-java-8-stream-foreach/40490130>

- removeIf() metódus:

<https://www.geeksforgeeks.org/arraylist-removeif-method-in-java/>

- Magyar elemek kezelése:

- <https://stackoverflow.com/questions/14918188/reading-text-file-with-utf-8-encoding-using-java>
- <https://github.com/SanFranciscoBoJottem/StreamsExample/blob/master/src/inandout/Reader.java>