<https://stackoverflow.com/questions/14918188/reading-text-file-with-utf-8-encoding-using-java>

whilenál két feltételt nem engedett egyszerre, nem enegdte a lista használatát

Streambuilder létrehozására, de nem találja az UTF-8-t

Stringbuilder és buffer között mi a különbég?

Találtam charsetet, rájöttem máshogy kell írni: <https://docs.oracle.com/javase/7/docs/api/java/nio/charset/StandardCharsets.html>

Mi a kül. A “” és a null között??

Tombök vagy listák használata

Oop2: utána kisbetűvel ad listát, nem szép

Továbbfejsztésre ötlet: szóközöket még válaszrol levagdosni

egyébként eszembe jutott egy olyan csalási lehetőség is, hogy angol a listánk, de a kiiratásba szólunk bele hogy mondjuk az elem classban csinálunk két tömböt ....

ja és egyébként még arra gondoltam, hogy valamit azzal lehetne manipulálni, hogy a fájlbeolvasásnál amúgyis bytokban vannak a szavak és ha jól értem amikor bekérünk valamit a felhasználtól aztis bytnak érti...szóval

Listák: egy lista amiben az elemek vannak

elemLista: széttörni őket szóközöknél, és bepakolni az elemeket listába, rálehet küldeni egy collections sortolást, hogy abcsorrendben legyenek

aJatekosLista: firstindexof

bJatekosLista:lastindexof

szétválogatás firstindexof meg lastindexof alapján, elinditva egy lépésszámlálót

menet:

1. először random kap egy elemet a listából mondjuk mondjuk aJatekos
2. másodikjatekos:fore-val megyünk végig a lista elemein: whileal mondjuk a kapott elem, vagyis a aJatekosLista utolsó elemének a lastindexofja egyenlő az elemListából random elkértelemnek a firstindexofjával && listasize nem 0, valasszunk kiegy elemet és pakoljuk bele bJatekoslistába ha van ilyen, lepesszam++

ötlet: akár megcsinálhatjuk kétféleképpen is, halmazból kiindulva vagy listábólkiindulós változat

ezt akár tesztelgethetjük i, hgy melyik gyorsabb, melyik eszik több memóriát, melyik több lépésszám stb…

package szolanc;

import java.util.ArrayList;

public class Elem {

private static String[] elemNev={"hidrogen","helium","litium","berillium","bor","szen","nitrogen","oxigen","fluor","neon","natrium","magnezium","aluminium","szilicium","foszfor","ken","klor","argon","kalium","kalcium","szkandium","titan","vanadium","krom","mangan","vas","kobalt","nikkel","rez","cink","gallium","germanium","arzen","szelen","brom","kripton","rubidium","stroncium","ittrium","cirkonium","niobium","molibden","technecium","rutenium","rodium","palladium","ezüst","kadmium","indium","on","antimon","tellur","jod","xenon","cezium","barium","lantan","cerium","prazeodimium","neodimium","prometium","szamarium","europium","gadolinium","terbium","diszprozium","holmium","erbium","tulium","itterbium","lutecium","hafnium","tantal","volfram","renium","ozmium","iridium","platina","arany","higany","tallium","olom","bizmut","polonium","asztacium","radon","francium","radium","aktinium","torium","protaktinium","uran","neptunium","plutonium","americium","kurium","berkelium","kalifornium","einsteinium","fermium","mendelevium","nobelium","laurencium","raderfordium","dubnium","sziborgium","bohrium","hasszium","meitnerium","darmstadtium","rontgenium","kopernicium","nihonium","flerovium","moszkovium","livermorium","tenesszium","oganeszon","ununennium","unbinilium","unbiunium","unbibium","unbikvadium","unbipentium","unbioktium","unbiennium","untrinilium","untriunium", "untribium"

};

private static String nev;

public Elem(String nev){ //ide majd exceptioneket bepakolni

this.nev = nev;

System.out.print(nev+" ");

}

// public static Elem getVeletlenElem(){

// String nev = elemNev[(int)(Math.random()\*(elemNev.length))];

// return new Elem(nev);

//

// }

public class elemLista{

public ArrayList<Elem> elemLista=new ArrayList<>();{

for (int i = 0; i <138; i++) {

String nev=elemNev[i];

elemNev[i].g

}

return elemLista();

}}

}

Első vrezió: package szolanc;

import java.util.ArrayList;

public class Elem {

private static String[] elemNev={"hidrogén","hélium","lítium","berillium","bór","szén","nitrogén","oxigén","fluor","neon","nátrium","magnézium","alumínium","szilícium","foszfor","kén","klór","argon","kálium","kalcium","szkandium","titán","vanádium","króm","mangán","vas","kobalt","nikkel","réz","cink","gallium","germánium","arzén","szelén","bróm","kripton","rubídium","stroncium","ittrium","cirkónium","nióbium","molibdén","technécium","ruténium","ródium","palládium","ezüst","kadmium","indium","ón","antimon","tellúr","jód","xenon","cézium","bárium","lantán","cérium","prazeodímium","neodímium","prométium","szamárium","európium","gadolínium","terbium","diszprózium","holmium","erbium","túlium","itterbium","lutécium","hafnium","tantál","volfrám","rénium","ozmium","irídium","platina","arany","higany","tallium","ólom","bizmut","polónium","asztácium","radon","francium","rádium","aktínium","tórium","protaktínium","urán","neptúnium","plutónium","amerícium","kűrium","berkélium","kalifornium","einsteinium","fermium","mendelévium","nobélium","laurencium","raderfordium","dubnium","sziborgium","bohrium","hasszium","meitnérium","darmstadtium","röntgénium","kopernícium","nihónium","fleróvium","moszkóvium","livermórium","tenesszium","oganeszon","ununennium","unbinilium","unbiunium","unbibium","unbikvadium","unbipentium","unbioktium","unbiennium","untrinilium","untriunium", "untribium"

};

private static String nev;

public Elem(String nev){ //ide majd exceptioneket bepakolni

this.nev = nev;

System.out.print(nev+" ");

}

// public static Elem getVeletlenElem(){

// String nev = elemNev[(int)(Math.random()\*(elemNev.length))];

// return new Elem(nev);

//

// }

private ArrayList<Elem> listaFeltolt(){

ArrayList<Elem> elemLista=new ArrayList<>();{

for (int i = 0; i <138; i++) {

String nev=elemNev[i];

elemNev[i].g

}

return elemLista();

}}

public static String getNev() {

return nev;

}

@Override

public String toString() {

return "A választott elem: "+nev.toUpperCase();

}

}

//class elemLista {

// private ArrayList<Elem> elemLista= new ArrayList<>();

//

//

// public void listafeltolt(){

//

Magyar abc szerint: Collections.sort(elemLista,(Elem e1, Elem e2) -> e1.OSSZEHASONLIT\_NEV.compare(e1, e2));

//sorbarendezés: magyar abc-vel bajok vannak

for (int i = 0; i < elemLista.size(); i++) {

System.out.print(elemLista.get(i).getNev()+" "); //kiirjuk a választékot

} ezzel sok hibat jelzett, clasban ezt irtam mellé: public Comparator OSSZEHASONLIT\_NEV = new Comparator() {

public int compare(Elem e1, Elem e2) {

Collator col = Collator.getInstance(new Locale("hu","HU"));

return col.compare(e1.nev, e2.nev);

}

@Override

public int compare(Object o1, Object o2) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

};

equAalsal próbálkoztam, containsel(itt az volt a baj, h bojektumot akart, a scanner meg string ugyebár

ugyanez equalssal, majd startswidth, de az if-be nem akar belemenni

public Listak kivalasztUjLista(Elem vObj, ArrayList<Elem> lista){ //atadni a kivalasztott elemet mint paramétert és azalapjan krealni az uj listakat

valamiért nem engedte, hogy listák osztályt adjon vissza a függvényem, mármint engedte, de túlindexelt, amikor meghívtam

Listak kivalasztLista= new Listak();

for (Elem elem : lista) {

if( vObj.getLastCharacter()==elem.getFirstCharacter())

kivalasztLista.lista.add(elem);

}

System.out.println("Most a másik válasszon az alabbiakból: "+kivalasztLista);

return kivalasztLista;

}}

Bajok vannak: ékezetes betűket nem ismeri fel

Sz és s betű kezelése?

Pptbe: több játékos játsza