**package** parse;  
  
  
**import** com.sun.rowset.internal.Row;  
**import** jxl.Sheet;  
**import** jxl.Workbook;  
**import** jxl.write.DateTime;  
**import** jxl.write.Label;  
**import** jxl.write.WritableSheet;  
**import** jxl.write.WritableWorkbook;  
  
**import** javax.swing.text.Document;  
**import** java.io.\*;  
**import** java.util.\*;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
**import** java.net.URLConnection;  
**import** java.nio.charset.Charset;  
**import** java.nio.charset.StandardCharsets;  
**import** java.util.zip.ZipEntry;  
**import** java.util.zip.ZipFile;  
**import static** org.apache.commons.lang3.StringUtils.\*;  
  
  
  
**public class** DEF14A {  
  
  
  
 **public static void** main(String[] args) **throws** Exception {  
  
*//* System.***out***.println(**"start processing"**);  
  
  
  
  
  
*// retrieve\_def14a\_filings\_txt();  
  
// extract\_content\_tables("/Reserch/2019 summer/First\_try/20\_Sample/20F\_rawFile",  
// "/Reserch/2019 summer/First\_try/20\_Sample/20F\_Contents",  
// "/Reserch/2019 summer/First\_try/20\_Sample/20F\_No\_Tables");  
//  
// search\_key\_words("million",  
// "/Reserch/2019 summer/First\_try/20\_Sample/20F\_Contents",  
// "/Reserch/2019 summer/First\_try/20\_Sample/Key\_Word\_Search/");  
 trim\_dollar*(**"/Reserch/2019 summer/First\_try/20\_Sample/25\_Contents"**,  
 **"/Reserch/2019 summer/First\_try/20\_Sample/Number of Numbers/"**);  
  
 System.***out***.println(**"end"**);  
  
  
 }  
  
  
 **public static void** extract\_proposals(String fileDir, String saveDir) **throws** IOException {  
 String fileName = fileDir.substring(fileDir.lastIndexOf(**"/"**), fileDir.length());  
 *//String saveFilePath = saveDir + File.separator + fileName.substring(0, fileName.lastIndexOf('.')) + ".html";* String saveFilePath = saveDir + File.***separator*** + fileName;  
 BufferedReader inputFile = **new** BufferedReader(**new** FileReader(fileDir));  
 PrintWriter outputFile = **new** PrintWriter(**new** FileWriter(saveFilePath));  
 outputFile.println(**"<link href=\"txtstyle.css\" rel=\"stylesheet\" type=\"text/css\" />"**);  
 **try** {  
 String lineOfText = inputFile.readLine();  
  
  
 **while** (!lineOfText.contains(**"</DOCUMENT>"**) && !lineOfText.contains(**"</document>"**)) {  
 **if** (!lineOfText.isEmpty()) {  
 outputFile.println(lineOfText);  
 }  
 lineOfText = inputFile.readLine();  
 }  
 outputFile.println(lineOfText);  
  
 inputFile.close();  
 outputFile.close();  
 } **catch** (FileNotFoundException ex) {  
 }  
 }  
  
  
 **public static boolean** is\_blank(String input) {  
 **for** (**int** i = 0; i < input.length(); ) {  
 **if** (input.charAt(i) == **' '**) {  
 ++i;  
 } **else** {  
 **return false**;  
 }  
 }  
 **return true**;  
 }  
  
  
 **public static boolean** contain\_only\_digit(String input) {  
 **for** (**int** i = 0; i < input.length(); ) {  
 **if** (!Character.*isDigit*(input.charAt(i))) {  
 **return false**;  
 } **else** {  
 ++i;  
 }  
 }  
 **return true**;  
 }  
  
 **public static void** get\_all\_def14a() **throws** MalformedURLException, IOException {  
 String edgarDataDir = **"https://www.sec.gov/Archives/edgar/data/"**;  
 String listPath = **"/Reserch/CIK.txt"**;  
 String CIK = **""**;  
 String currCompFiles = **""**;  
  
 **try** (BufferedReader br = **new** BufferedReader(**new** FileReader(listPath))) {  
 **while** ((CIK = br.readLine()) != **null**) {  
 currCompFiles = edgarDataDir + CIK;  
 *//System.out.println("This "+ currCompFiles);* URL url = **new** URL(currCompFiles);  
 URLConnection con = url.openConnection();  
 InputStream is = con.getInputStream();  
 BufferedReader inputFile = **new** BufferedReader(**new** InputStreamReader(is));  
 PrintStream out = **new** PrintStream(**new** FileOutputStream(**"/Reserch/DEF14A/"** + CIK + **".txt"**));  
 System.*setOut*(out);  
 **try** {  
 String line = **null**;  
 **while** ((line = inputFile.readLine()) != **null**) {  
 **if** (line.contains(**"Directory Listing"**)) {  
 String plain = line.replaceAll(**"(?s)<[^>]\*>(\\s\*<[^>]\*>)\*"**, **" "**);  
 String[] splited = plain.split(**"\\s+"**);  
 **for** (**int** i = 0; i < splited.**length**; ++i) {  
 String curr = splited[i];  
 **if** (*contain\_only\_digit*(curr)) {  
 System.***out***.println(curr);  
 }  
 }  
 **break**;  
 }  
 }  
 } **catch** (Exception e) {  
 }  
 }  
 } **catch** (IOException e) {  
 System.***err***.println(e);  
 }  
 }  
  
 **public static void** def14a\_matching() **throws** MalformedURLException, IOException {  
*// String fileName="/Reserch/DEF14A/Filing Index/320193.txt";  
// BufferedReader inputFile = new BufferedReader(  
// new FileReader(fileName));* String CIK = **"320193"**;  
 String lineOfText;  
 **int** i = 0;  
 String saveFilePath = **"/Reserch/DEF14A/output/320193.txt"**;  
 PrintWriter outputFile = **new** PrintWriter(**new** FileWriter(saveFilePath));  
 ArrayList <String> def14a = **new** ArrayList <>();  
 **try** (BufferedReader br = **new** BufferedReader(**new** FileReader(**"/Reserch/DEF14A/Filing Index/"** + CIK + **".txt"**))) {  
  
  
 **while** ((lineOfText = br.readLine()) != **null**) {  
 i = i + 1;  
 System.***out***.println(i);  
 **if** (!*is\_blank*(lineOfText)) {  
 String txt\_file = lineOfText.substring(0, 10)  
 + **"-"** + lineOfText.substring(10, 12)  
 + **"-"** + lineOfText.substring(12, lineOfText.length())  
 + **".txt"**;  
 URL url = **new** URL(**"https://www.sec.gov/Archives/edgar/data/"** + CIK + **"/"** + lineOfText + **"/"** + txt\_file);  
 Scanner s = **new** Scanner(url.openStream());  
 String line = s.nextLine();  
 **while** (!line.contains(**"CONFORMED SUBMISSION TYPE:"**)) {  
 line = s.nextLine();  
 }  
 **if** (line.contains(**"DEF 14A"**)) {  
 System.***out***.println(**"This is DEF 14A: "** + lineOfText);  
 def14a.add(lineOfText);  
 }  
 }  
  
 }  
  
 }  
 **for** (**int** j = 0; j < def14a.size(); j++) {  
 outputFile.println(def14a.get(j));  
 }  
 outputFile.close();  
 }  
  
 **public static void** search\_all\_index\_files() **throws** FileNotFoundException, IOException {  
 File dir = **new** File(**"/Reserch/2019 summer/First\_try/by\_year/"**);  
 File[] index\_files\_by\_year = dir.listFiles();  
 **if** (index\_files\_by\_year != **null**) {  
 **for** (File index\_file : index\_files\_by\_year) {  
 **if** (!index\_file.isHidden()) {  
 String input = index\_file.getPath();  
 String fileName = index\_file.getName();  
 String output = **"/Reserch/2019 summer/First\_try/20F\_output/"** + fileName + **".txt"**;  
 *access\_index\_file*(input, output);  
 }  
 }  
 }  
 }  
  
  
 **public static void** access\_index\_file(String input, String output) **throws** FileNotFoundException, IOException {  
 BufferedReader br = **new** BufferedReader(**new** FileReader(input));  
 String lineOfText = br.readLine();  
 PrintWriter outputFile = **new** PrintWriter(**new** FileWriter(output));  
  
 **while** (lineOfText != **null** && !lineOfText.contains(**"DEF 14A"**)) {  
 *//System.out.println(lineOfText);* lineOfText = br.readLine();  
 }  
 **try** {  
 **while** (lineOfText.contains(**"DEF 14A"**)) {  
 **int** start\_char = lineOfText.indexOf(**"edgar"**);  
 **int** end\_char = lineOfText.indexOf(**"txt"**) + 3;  
 String path = **"https://www.sec.gov/Archives/"** + lineOfText.substring(start\_char, end\_char);  
  
 outputFile.println(path);  
 lineOfText = br.readLine();  
 }  
 } **catch** (Exception e) {  
 System.***err***.println(**"Cannot access..."**);  
 }  
 br.close();  
 outputFile.close();  
 }  
  
 **public static void** file() **throws** MalformedURLException, IOException {  
 File dir = **new** File(**"/Reserch/DEF14A/index-original/index file/index file-second/1994"**);  
 String saveFilePath = **"/Reserch/DEF14A/index-original/output/1994.txt"**;  
 PrintWriter outputFile = **new** PrintWriter(**new** FileWriter(saveFilePath));  
 File[] index\_by\_year = dir.listFiles();  
 **if** (index\_by\_year != **null**) {  
 **for** (File index\_file : index\_by\_year) {  
 **if** (!index\_file.isHidden()) {  
 String input = index\_file.getPath();  
  
  
 String lineofText;  
  
 **try** (BufferedReader br = **new** BufferedReader(**new** FileReader(input))) {  
 **while** ((lineofText = br.readLine()) != **null**) {  
 **if** (lineofText.contains(**"DEF 14A"**)) {  
  
 outputFile.println(lineofText);  
 }  
 }  
 br.close();  
  
  
 } **catch** (IOException e) {  
 System.***err***.println(e);  
 }  
 }  
 }  
 }  
 outputFile.close();  
  
 }  
  
 **public static void** download() **throws** IOException {  
 String filepath = **"/Reserch/2019 summer/First\_try/20F\_index.txt"**;  
 BufferedReader inputFile = **new** BufferedReader(**new** FileReader(filepath));  
 String website = **""**;  
  
  
 **while** ((website = inputFile.readLine()) != **null**) {  
 *//int start = website.indexOf("a/");* **int** end = website.indexOf(**".txt"**);  
 String cur = website.substring(40, end);  
 String delete = cur.replaceAll(**"-"**, **""**);  
 String saved = delete.replaceAll(**"/"**, **"-"**);  
 String savepath = **"/Reserch/txttest/"** + saved + **".txt"**;  
 PrintWriter outputfile = **new** PrintWriter(**new** FileWriter(savepath));  
  
  
 URL urlObject = **new** URL(website);  
 URLConnection urlConnection = urlObject.openConnection();  
  
  
 InputStream is = urlConnection.getInputStream();  
 BufferedReader input = **new** BufferedReader(**new** InputStreamReader(is));  
 **try** {  
 String line = **null**;  
 **while** ((line = input.readLine()) != **null**) {  
 outputfile.println(line);  
  
 }  
 outputfile.close();  
  
  
 } **catch** (Exception e) {  
 }  
  
  
 }  
 inputFile.close();  
  
  
 }  
  
 **public static void** filedownload() **throws** IOException {  
  
 File dir = **new** File(**"/Reserch/test websit"**);  
 File[] index\_by\_year = dir.listFiles();  
 **if** (index\_by\_year != **null**) {  
 **for** (File year\_file : index\_by\_year) {  
 **if** (!year\_file.isHidden()) {  
  
  
 String filepath = year\_file.getPath();  
  
 **try** (BufferedReader inputFile = **new** BufferedReader(**new** FileReader(filepath))) {  
  
  
 String website = **""**;  
  
  
 **while** ((website = inputFile.readLine()) != **null**) {  
  
 *//rename it* **int** end = website.indexOf(**".txt"**);  
 String cur = website.substring(40, end);  
 String delete = cur.replaceAll(**"-"**, **""**);  
 String saved = delete.replaceAll(**"/"**, **"-"**);  
 String savepath = **"/Reserch/test websit output/"** + saved + **".txt"**;  
 PrintWriter outputfile = **new** PrintWriter(**new** FileWriter(savepath));  
  
  
 URL urlObject = **new** URL(website);  
 URLConnection urlConnection = urlObject.openConnection();  
  
  
 InputStream is = urlConnection.getInputStream();  
 BufferedReader input = **new** BufferedReader(**new** InputStreamReader(is));  
 **try** {  
 String line = **null**;  
 **while** ((line = input.readLine()) != **null**) {  
 outputfile.println(line);  
  
 }  
 input.close();  
 outputfile.close();  
  
  
 } **catch** (Exception e) {  
  
  
 }  
  
 }  
 inputFile.close();  
 } **catch** (Exception e) {  
  
 }  
 }  
  
 }  
  
 }  
  
  
 }  
  
  
 */\* download txt file by index \*/* **public static void** retrieve\_def14a\_filings\_txt() **throws** FileNotFoundException, IOException {  
 String def14a = **""**;  
 *//String listPath = "/Users/qingouyang/Desktop/DEF 14A/DEF14A Index/1994.1.txt";* File dir = **new** File(**"/Reserch/2019 summer/First\_try/by\_year/"**);  
 File[] file\_list = dir.listFiles();  
 **if** (file\_list != **null**) {  
 **for** (File seasonal : file\_list) {  
 **if** (!seasonal.isHidden()) {  
 String year = seasonal.getName().substring(0, seasonal.getName().indexOf(**'.'**));  
 File theDir = **new** File(**"/Reserch/2019 summer/First\_try/20F\_output/"** + year);  
 **if** (!theDir.exists()) {  
 **try** {  
 theDir.mkdir();  
 } **catch** (SecurityException se) {  
 }  
 }  
 **try** (BufferedReader br = **new** BufferedReader(**new** FileReader(seasonal))) {  
 **while** ((def14a = br.readLine()) != **null**) {  
 **if** (!*is\_blank*(def14a)) {  
 String filePath = def14a.substring(def14a.indexOf(**"data"**) + 5, def14a.length());  
 String CIK = filePath.substring(0, filePath.indexOf(**'/'**));  
 String fileName = CIK + **"."** + filePath.substring(filePath.indexOf(**'/'**) + 1, filePath.length());  
 URL url = **new** URL(def14a);  
 URLConnection con = url.openConnection();  
 InputStream is = con.getInputStream();  
 BufferedReader inputFile = **new** BufferedReader(**new** InputStreamReader(is));  
 PrintWriter outputFile = **new** PrintWriter(  
 **new** FileWriter(theDir.getPath() + **"/"** + fileName));  
 **boolean** html = *contains\_html\_tag*(def14a);  
 **if** (html) {  
 **try** {  
 String lineOfText = inputFile.readLine();  
 **while** (!lineOfText.contains(**"</DOCUMENT>"**) && !lineOfText.contains(**"</document>"**)  
 && !lineOfText.contains(**"</SEC-DOCUMENT>"**)  
 && !lineOfText.contains(**"</sec-document>"**)) {  
 **if** (!lineOfText.isEmpty()) {  
 outputFile.println(lineOfText);  
 }  
 lineOfText = inputFile.readLine();  
 }  
 outputFile.println(lineOfText);  
 inputFile.close();  
 outputFile.close();  
 } **catch** (Exception e) {  
 }  
 System.***out***.println(fileName + **" has been downloaded"**);  
 } **else** {  
 *//outputFile.println("<HTML>");  
 //outputFile.println("<TABLE>");  
 //outputFile.println("<PRE>");* **try** {  
 String lineOfText = inputFile.readLine();  
 **while** (!lineOfText.contains(**"</DOCUMENT>"**) && !lineOfText.contains(**"</document>"**)  
 && !lineOfText.contains(**"</SEC-DOCUMENT>"**)  
 && !lineOfText.contains(**"</sec-document>"**)) {  
 **if** (!lineOfText.isEmpty()) {  
 outputFile.println(lineOfText);  
 }  
 lineOfText = inputFile.readLine();  
 }  
 outputFile.println(lineOfText);  
 *//outputFile.println("</TABLE>");  
 //outputFile.println("</PRE>");  
 //outputFile.println("</HTML>");* inputFile.close();  
 outputFile.close();  
 } **catch** (Exception e) {  
 }  
 System.***out***.println(fileName + **" has been downloaded"**);  
 }  
  
  
 }  
  
 }  
  
 }  
 }  
 }  
 }  
  
 }  
  
 **public static boolean** contains\_html\_tag(String def14a) **throws** FileNotFoundException, IOException {  
 URL url = **new** URL(def14a);  
 URLConnection con = url.openConnection();  
 InputStream is = con.getInputStream();  
 BufferedReader inputFile = **new** BufferedReader(**new** InputStreamReader(is));  
 **try** {  
 String lineOfText = inputFile.readLine();  
 **while** (lineOfText != **null**) {  
 **if** (lineOfText.contains(**"<HTML>"**) || lineOfText.contains(**"<html>"**)) {  
 **return true**;  
 } **else** {  
 lineOfText = inputFile.readLine();  
 }  
 }  
  
 } **catch** (Exception e) {  
 }  
 **return false**;  
 }  
  
 */\*  
 \* Extracts contents outside of all the HTML tags by given directory  
 \*/* **public static void** extract\_content\_tables(String dir, String saveDir, String no\_table) **throws** FileNotFoundException, IOException {  
 File dr = **new** File(dir);  
 File[] list = dr.listFiles();  
 **if** (list != **null**) {  
 **for** (File f : list) {  
 **if** (!f.isHidden()) {  
 String p = f.getPath();  
 System.***out***.println(**"start processing: "** + f.getName());  
 *extract\_body*(p, saveDir, no\_table);  
 System.***out***.println(**"done processing: "** + f.getName());  
 }  
 }  
 }  
 }  
  
 */\*  
 \* Extracts contents outside of all the HTML tags by given file path  
 \* Leave out the tables to tableDir folder  
 \*/* **public static void** extract\_body(String fileDir, String saveDir, String no\_table) **throws** IOException {  
 String fileName = fileDir.substring(fileDir.lastIndexOf(**"/"**), fileDir.length());  
 String saveFilePath = saveDir + File.***separator*** + fileName;  
 String tablePath = no\_table + File.***separator*** + fileName;  
 BufferedReader inputFile = **new** BufferedReader(**new** FileReader(fileDir));  
 PrintWriter outputFile = **new** PrintWriter(**new** FileWriter(saveFilePath));  
 PrintWriter no\_table\_File = **new** PrintWriter(**new** FileWriter(tablePath));  
 *//System.out.println(fileName);* **try** {  
 String lineOfText = inputFile.readLine();  
  
 **while** (lineOfText != **null**) {  
 **if** (lineOfText.length() <= 2000) {  
 *//System.out.println("within 2000");* **if** (lineOfText.contains(**"page"**) || lineOfText.contains(**"Page"**)) {  
  
 lineOfText = inputFile.readLine();  
  
 } **else** {  
  
 **if** (lineOfText.contains(**"<table"**) || lineOfText.contains(**"<TABLE"**)) {  
 **while** (!lineOfText.contains(**"</table>"**) && !lineOfText.contains(**"</TABLE>"**)) {  
 lineOfText = inputFile.readLine();  
 }  
 } **else** {  
 no\_table\_File.println(lineOfText);  
 String pureText = **""**;  
 **for** (**int** i = 0; i < lineOfText.length(); ) {  
 **if** (lineOfText.charAt(i) != **'<'**) {  
 **if** (!pureText.isEmpty() && !pureText.endsWith(**" "**)) {  
 pureText += **" "**;  
 }  
 **while** (lineOfText.charAt(i) != **'<'**) {  
 pureText = pureText + lineOfText.charAt(i);  
 ++i;  
 **if** (i >= lineOfText.length()) {  
 **break**;  
 }  
 }  
 } **else** {  
 **while** (lineOfText.charAt(i) != **'>'**) {  
 ++i;  
 **if** (i >= lineOfText.length()) {  
 **break**;  
 }  
 }  
 ++i;  
 }  
 }  
 **if** (pureText.contains(**">"**)) {  
 pureText = pureText.substring(pureText.indexOf(**">"**) + 1, pureText.length());  
 }  
  
 **while** (pureText.contains(**"&"**) && pureText.contains(**";"**)  
 && (pureText.indexOf(**';'**) - pureText.indexOf(**'&'**)) <= 8) {  
 pureText = *delete\_symbolPair*(pureText);  
  
 }  
  
 **if** (!pureText.isEmpty() && !*all\_blank*(pureText)) {  
 outputFile.println(pureText);  
 }  
 }  
 }  
 } **else**{  
 *//System.out.println("out of 2000");* String single = *process\_one\_line*(lineOfText);  
 no\_table\_File.println(single);  
 *//tableFile.println(empty);* outputFile.println(*parse\_one\_line*(single));  
 }  
  
 lineOfText = inputFile.readLine();  
 }  
 inputFile.close();  
 outputFile.close();  
 no\_table\_File.close();  
  
 } **catch** (Exception ex) {  
 System.***err***.println(**"Oops...string index out of bound: "**);  
 }  
  
 }  
  
 **public static** String delete\_symbolPair(String input) {  
 **if** (input.indexOf(**'&'**) == -1 || input.indexOf(**';'**) == -1) {  
 **return** input;  
 } **else if** (input.contains(**"&ndash"**)) {  
 **return** input.replace(**"&ndash;"**, **"-"**);  
 } **else if** (input.contains(**"&#151;"**)) {  
 **return** input.replace(**"&#151;"**, **"-"**);  
 } **else if** (input.contains(**"&#146;"**)) {  
 **return** input.replace(**"&#146;"**, **"'"**);  
 } **else if** (input.contains(**"&#160;"**)) {  
 **return** input.replace(**"&#160;"**, **" "**);  
 } **else if** (input.contains(**"&#8212;"**)) {  
 **return** input.replace(**"&#8212;"**, **" "**);  
 } **else if** (input.contains(**"&#8211;"**)) {  
 **return** input.replace(**"&#8211;"**, **" "**);  
 } **else if** (input.contains(**"&mdash;"**)) {  
 **return** input.replace(**"&mdash;"**, **"-"**);  
 } **else if** (input.contains(**"&nbsp;"**)) {  
 **return** input.replace(**"&nbsp;"**, **" "**);  
 } **else** {  
 **int** beg = input.indexOf(**'&'**);  
 **int** end = input.indexOf(**';'**, beg);  
 **if** (beg == 0 && end == input.length() - 1) {  
 **return ""**;  
 } **else if** (beg == 0) {  
 **return** input.substring(end + 1, input.length());  
 } **else if** (end == input.length() - 1) {  
 **return** input.substring(0, beg);  
 } **else** {  
 String s1 = input.substring(0, beg);  
 String s2 = input.substring(end + 1, input.length());  
 **return** s1 + s2;  
 }  
 }  
 }  
  
 **public static boolean** all\_blank(String input) {  
 **for** (**int** i = 0; i < input.length(); ) {  
 **if** (Character.*isWhitespace*(input.charAt(i))) {  
 ++i;  
 } **else** {  
 **return false**;  
 }  
 }  
 **return true**;  
 }  
  
  
 **public static** String process\_one\_line(String input) {  
 input = input.toLowerCase();  
 **while**(input.contains(**"page"**)||input.contains(**"Page"**)){  
  
 }  
 **while** (input.contains(**"<table"**)) {  
 **int** table\_beg\_index = input.indexOf(**"<table"**);  
 **int** table\_end\_index = input.indexOf(**"/table>"**);  
 **int** new\_start = table\_end\_index + 7;  
 input = input.substring(0, table\_beg\_index) + input.substring(new\_start);  
 }  
 **return** input;  
 }  
  
 */\*  
 \* very similar to remove\_tags, by adding " " between words that are separated by tags  
 \*/* **public static** String parse\_one\_line(String input) {  
 String output = **""**;  
 **int** i = 0;  
 **while** (i < input.length()) {  
 **if** (input.charAt(i) == **'<'**) {  
 **while** (input.charAt(i) != **'>'**) {  
 ++i;  
 }  
 **if** (!output.isEmpty() && !output.endsWith(**" "**)) {  
 output += **" "**;  
 }  
 **if** ((i + 1) < input.length() && input.charAt(i + 1) == **' '**) {  
 **while** (input.charAt(i + 1) == **' '**) {  
 ++i;  
 }  
 }  
 } **else** {  
 output += input.charAt(i);  
 }  
 ++i;  
 }  
 **if** (output.contains(**">"**)) {  
 output = output.substring(output.indexOf(**">"**) + 1, output.length() - 1);  
 }  
 **while** (output.contains(**"&"**)) {  
 output = *delete\_symbolPair*(output);  
 }  
 **return** output;  
 }  
  
 */\*  
 \* Searches keywords by listing all the existing words in the documents  
 \* Searches one keyword or combined keywords  
 \* Case insensitive  
 \*/* **public static void** search\_key\_words(String key\_words, String inputDir, String saveDir) **throws** FileNotFoundException, IOException {  
 File searchDir = **new** File(inputDir);  
 PrintWriter outputFile = **new** PrintWriter(  
 **new** FileWriter(saveDir + key\_words + **".txt"**));  
 String key = key\_words.toLowerCase();  
 String[] keys = key.split(**" "**);  
 **int** size = keys.**length**;  
 File[] list = searchDir.listFiles();  
 **if** (list != **null**) {  
 **for** (File f : list) {  
 System.***out***.println(f.getName() + **" start processing"**);  
 BufferedReader inputFile = **new** BufferedReader(**new** FileReader(f));  
 **int** count = 0;  
 List <String> Words = **new** ArrayList <String>();  
 String line = inputFile.readLine();  
 **while** (line != **null**) {  
 List <String> non\_blank = **new** ArrayList <String>();  
 String l = line.toLowerCase();  
 String[] words = l.split(**" "**);  
 **for** (**int** i = 0; i < words.**length**; ++i) {  
 **if** (!*all\_blank*(words[i])) {  
 **if** (!*trim\_blank*(words[i]).isEmpty()) {  
 non\_blank.add(*trim\_blank*(words[i]));  
 }  
 }  
 }  
 **for** (**int** i = 0; i < non\_blank.size(); ++i) {  
 String mot = non\_blank.get(i);  
 **int** check = *symbol\_in\_between*(mot);  
 **if** (check == -1) {  
 Words.add(mot);  
 } **else** {  
 String mot1 = mot.substring(0, check);  
 String mot2 = mot.substring(check + 1, mot.length());  
 **if** (!mot1.isEmpty()) {  
 Words.add(mot1);  
 }  
 **if** (!mot2.isEmpty()) {  
 Words.add(mot2);  
 }  
 }  
 }  
 line = inputFile.readLine();  
 }  
  
 **for** (**int** i = 0; i + size - 1 < Words.size(); ) {  
 String front = *trim\_front\_symbol*(Words.get(i));  
 **if** (front.equals(keys[0])) {  
 List <String> word\_list = **new** ArrayList <String>();  
 word\_list.add(front);  
 **int** s = i + 1;  
 **while** (word\_list.size() < keys.**length**) {  
 **if** (word\_list.size() != keys.**length** - 1) {  
 word\_list.add(Words.get(s));  
 ++s;  
 } **else** {  
 String back = *trim\_back\_symbol*(Words.get(s));  
 **if** (!back.equals(**""**)) {  
 word\_list.add(back);  
 } **else** {  
 ++s;  
 }  
 }  
  
 }  
 System.***out***.println(word\_list);  
 **int** c = 0;  
 **for** (**int** j = 0; j < size; ) {  
 **if** (word\_list.get(j).equals(keys[j])) {  
 ++j;  
 c = j;  
 } **else** {  
 **break**;  
 }  
 }  
 **if** (c == size) {  
 ++count;  
 }  
 i += c;  
 } **else** {  
 ++i;  
 }  
 }  
  
 **if** (count != 0) {  
 System.***out***.println(f.getName() + **": "** + count);  
 outputFile.println(f.getName() + **": "** + count);  
 }  
 inputFile.close();  
  
 }  
 }  
 outputFile.close();  
 }  
  
  
 **public static** String trim\_blank(String lineOfText) {  
 **int** n = 0;  
 **int** m = lineOfText.length();  
 **for** (**int** i = 0; i < lineOfText.length(); ) {  
 **if** (Character.*isWhitespace*(lineOfText.charAt(i))) {  
 ++i;  
 n = i;  
 } **else** {  
 **break**;  
 }  
 }  
 **for** (**int** j = lineOfText.length() - 1; j >= n; ) {  
 **if** (Character.*isWhitespace*(lineOfText.charAt(j))) {  
 --j;  
 m = j + 1;  
 } **else** {  
 **break**;  
 }  
 }  
 **if** (m != n + 1) {  
 **return** lineOfText.substring(n, m);  
 } **else** {  
 **return ""**;  
 }  
 }  
  
  
 **public static int** symbol\_in\_between(String input) {  
 **int** i = 0;  
 **for** (**int** m = 0; m < input.length(); ) {  
 **if** (Character.*isAlphabetic*(input.charAt(m))) {  
 i = m;  
 **break**;  
 } **else** {  
 ++m;  
 }  
 }  
 **int** j = 0;  
 **for** (**int** m = input.length() - 1; m >= 0; ) {  
 **if** (Character.*isAlphabetic*(input.charAt(m))) {  
 j = m;  
 **break**;  
 } **else** {  
 --m;  
 }  
 }  
  
 **if** (i == j || i + 1 == j) {  
 **return** -1;  
 } **else** {  
 **for** (**int** n = i; n <= j; ) {  
 **if** (!Character.*isAlphabetic*(input.charAt(n)) && input.charAt(n) != **'\''**) {  
 **return** n;  
 } **else** {  
 ++n;  
 }  
 }  
 **return** -1;  
 }  
 }  
  
 **public static** String trim\_front\_symbol(String input) {  
 String output = **""**;  
 **int** n = 0;  
 **for** (**int** i = 0; i < input.length(); ) {  
 **if** (!Character.*isAlphabetic*(input.charAt(i))) {  
 ++i;  
 n = i;  
 } **else** {  
 n = i;  
 **break**;  
 }  
 }  
 output = input.substring(n, input.length());  
 **return** output;  
 }  
  
  
 **public static** String trim\_back\_symbol(String input) {  
 String output = **""**;  
 **int** n = input.length() - 1;  
 **for** (**int** i = input.length() - 1; i >= 0; ) {  
 **if** (!Character.*isAlphabetic*(input.charAt(i))) {  
 --i;  
 n = i;  
 } **else** {  
 n = i;  
 **break**;  
 }  
 }  
 output = input.substring(0, n + 1);  
 **return** output;  
 }  
  
 */\* count the dollar \*/* **public static void** trim\_dollar(String inputDir, String saveDir) **throws** Exception {  
 File searchDir = **new** File(inputDir);  
*// WritableWorkbook workbook = Workbook.createWorkbook(new File("output.xls"));  
// WritableSheet sheet = workbook.createSheet("First Sheet", 0);* WritableWorkbook workbook = Workbook.*createWorkbook*(**new** File(**"/Reserch/2019 summer/First\_try/20\_Sample/Number of Numbers/output.xls"**));  
  
 **int** fileNum=0;  
 WritableSheet sheet = workbook.createSheet(**"First Sheet"**, 0);  
 File[] list = searchDir.listFiles();  
 **if** (list != **null**) {  
 **for** (File f : list) {  
  
  
 **int** count\_Num=0;  
 PrintWriter outputFile = **new** PrintWriter(  
 **new** FileWriter(saveDir + f.getName() + **".txt"**));  
  
 System.***out***.println(f.getName() + **" start processing"**);  
 BufferedReader inputFile = **new** BufferedReader(**new** FileReader(f));  
 List <String> Words = **new** ArrayList <String>();  
 String line = inputFile.readLine();  
 **while** (line != **null**) {  
 List <String> non\_blank = **new** ArrayList <String>();  
 String l = line.toLowerCase();  
 String[] words = l.split(**" "**);  
 **for** (**int** i = 0; i < words.**length**; ++i) {  
 **if** (!*all\_blank*(words[i])) {  
*// System.out.println(words[i]);  
// if (!trim\_blank(words[i]).isEmpty()) {  
// System.out.println(words[i]);  
// non\_blank.add(trim\_blank(words[i]));  
// }* non\_blank.add(words[i]);  
 }  
 }  
 **for** (**int** i = 0; i < non\_blank.size(); ++i) {  
 String mot = non\_blank.get(i);  
 **int** check = *symbol\_in\_between*(mot);  
 **if** (check == -1) {  
 Words.add(mot);  
 } **else** {  
 String mot1 = mot.substring(0, check);  
 String mot2 = mot.substring(check + 1, mot.length());  
 **if** (!mot1.isEmpty()) {  
 Words.add(mot1);  
 }  
 **if** (!mot2.isEmpty()) {  
 Words.add(mot2);  
 }  
 }  
 }  
 System.***out***.println(Words);  
 line = inputFile.readLine();  
 }  
  
  
  
*// System.out.println(f.getName() + ": " +Words.size());* **for** (**int** i = 0; i < Words.size();i++ ) {  
 String value=*getDigits*(Words.get(i));  
*// LinkedList<String> content\_digit=new LinkedList <>();* **if**(!value.equals(**""**) ) {  
 **if** (*is\_brackets*(Words.get(i)) || Words.get(i).contains(**"-"**)) {  
*// i++;* } **else** {  
 System.***out***.println(**"there is a number:"**+value);  
 LinkedList<String> content\_digit=**new** LinkedList <>();  
  
 **if** (i - 1 >= 0) { *//get the word before number* content\_digit.add(Words.get(i - 1));  
 }  
 content\_digit.add((Words.get(i))); *//get that number* **if** (i + 1 < Words.size()) { *//get the word after number* content\_digit.add(Words.get(i+1));  
 }  
 **if** (!*is\_year*(content\_digit)) { *//if it is not year* count\_Num++;  
 StringBuffer sb = **new** StringBuffer();  
 **for** (**int** k = 0; k < content\_digit.size(); k++) {  
 String cur\_content = content\_digit.get(k);  
 **if** (*meaningful\_String*(cur\_content)) {  
 sb.append(content\_digit.get(k) + **" "**);  
 }  
 }  
 outputFile.println(sb);  
 }  
 }  
 }  
  
 }  
 outputFile.close();  
 inputFile.close();  
 System.***out***.println(**"done "**+ f.getName());  
  
 sheet.addCell(**new** Label(0, fileNum, f.getName()));  
 sheet.addCell(**new** Label(1, fileNum, String.*valueOf*(count\_Num)));  
 fileNum++;  
 }  
 }  
 workbook.write();  
 workbook.close();  
 }  
  
 **public static boolean** is\_year(LinkedList<String> content\_digit){  
 **boolean** hasYear = **false**;  
 **for**(**int** i=0;i<content\_digit.size();i++) {  
 String text = content\_digit.get(i);  
 String text\_Number = text.replaceAll(**"[^0-9]"**, **"#"**); *//simple solution for replacing all non digits.* String[] arr = text\_Number.split(**"#"**);  
  
 **for** (String s : arr) {  
 **if** (s.matches(**"^[0-9]{4}$"**)) {  
 hasYear= **true**;  
  
 }  
 }  
  
 **if**(text.equals(**"january"**)||text.equals(**"february"**) || text.equals(**"march"**)  
 ||text.equals(**"april"**)|| text.equals(**"may"**)||text.equals(**"june"**)  
 ||text.equals(**"july"**)||text.equals(**"august"**)||text.equals(**"september"**)  
 ||text.equals(**"october"**)||text.equals(**"november"**)||text.equals(**"december"**)  
 ||text.equals(**"jan"**)|| text.equals(**"feb"**)||text.equals(**"mar"**)  
 ||text.equals(**"apr"**)||text.equals(**"jun"**)||text.equals(**"jul"**)  
 ||text.equals(**"aug"**)||text.equals(**"sept"**)||text.equals(**"oct"**)  
 ||text.equals(**"nov"**)||text.equals(**"dec"**) || text.contains(**"section"**)  
 ||text.contains(**"item"**)||text.contains(**"note"**)){  
 hasYear= **true**;  
  
 }  
  
*// else if(text.equals("section")||text.equals("item")){  
// hasYear= true;  
// }* }  
 **return** hasYear;  
 }  
  
*//  
// public static boolean is\_monthday(LinkedList<String> content\_digit){  
// boolean monthday=false;  
// for()  
// }* **public static boolean** meaningful\_String(String s){  
 **if**(s.equals(**"and"**) || s.equals(**"the"**) ||s.equals(**"of"**) || s.equals(**"or"**)  
 ||s.equals(**"with"**) ||s.equals(**"from"**)||s.equals(**"was"**)){  
 **return false**;  
 }  
 **return true**;  
 }  
  
 **public static boolean** is\_brackets(String s){  
 **int** len=s.length();  
 **if**(s.charAt(0)==**'('** && s.charAt(len-1)==**')'**&& len <5){  
 **return true**;  
 }  
 **if**(s.contains(**"("**) && s.contains(**")"**)){  
 **int** index\_start=s.indexOf(**"("**);  
 **int** index\_end=s.indexOf(**")"**);  
 String part=s.substring(index\_start+1,index\_end);  
 **if**(*isAlphaSpace*(part)){  
 **return true**;  
 }  
 **return false**;  
  
 }  
 **return false**;  
 }  
  
  
  
}