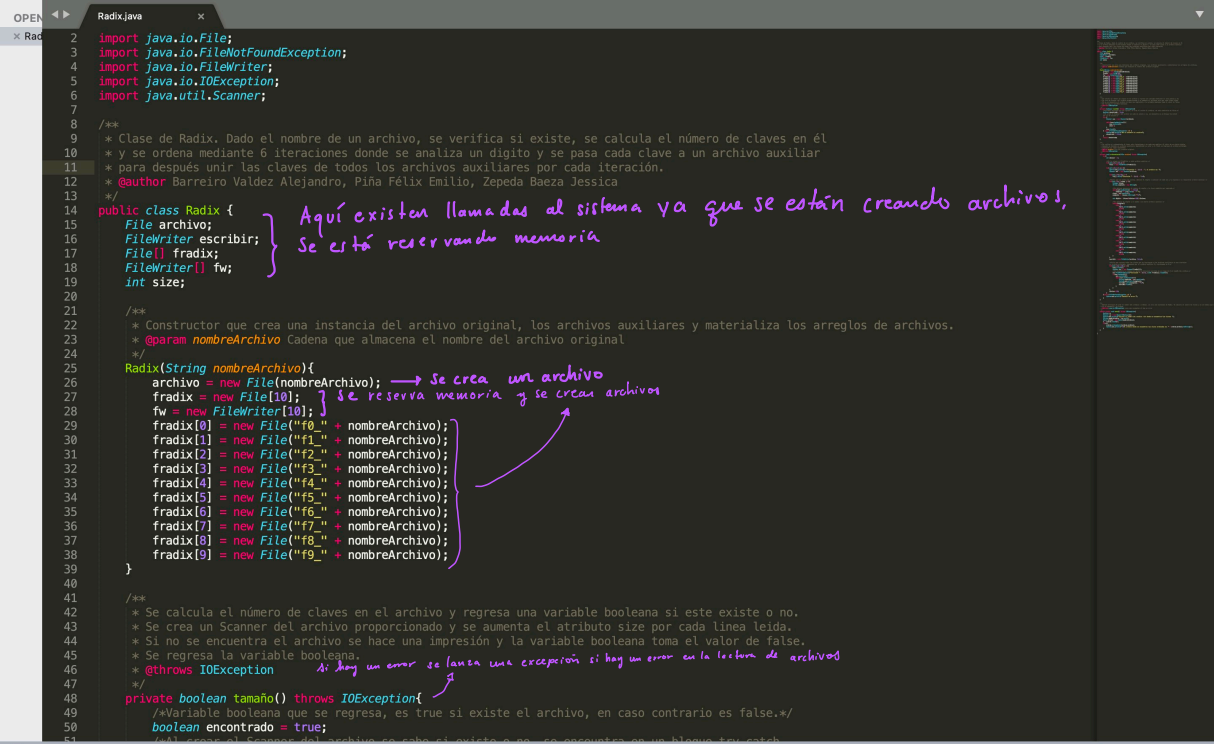
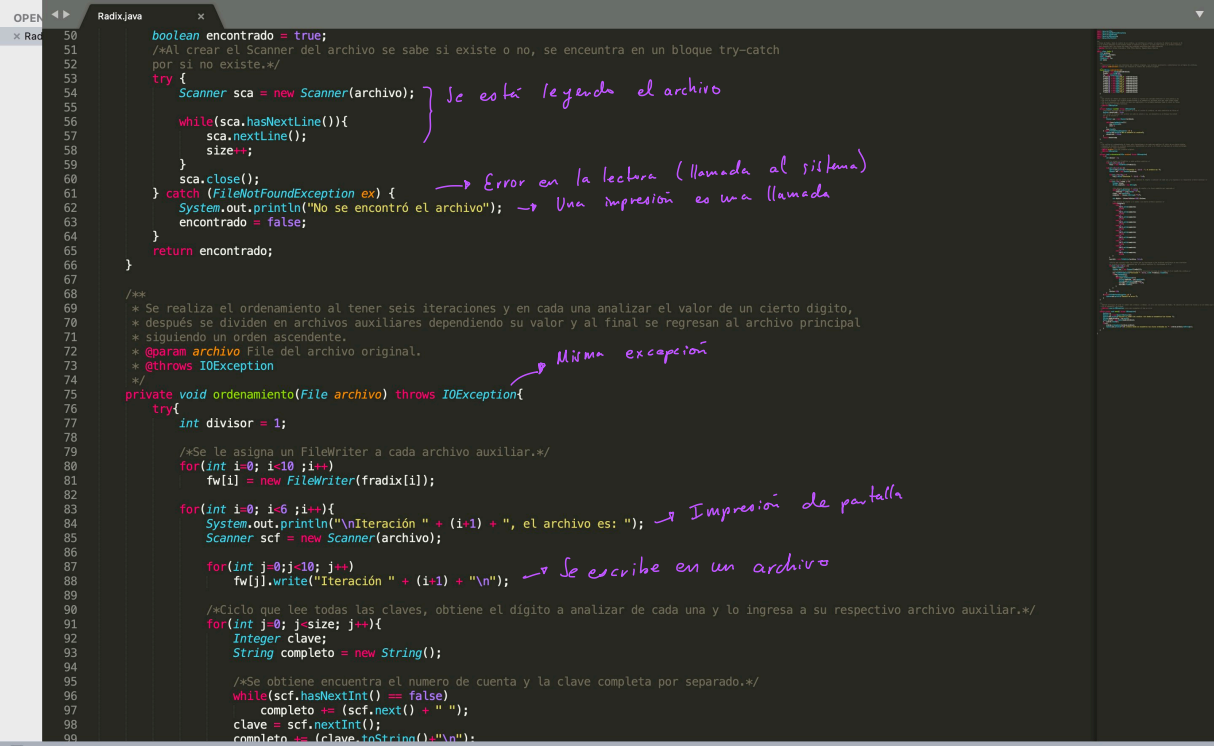
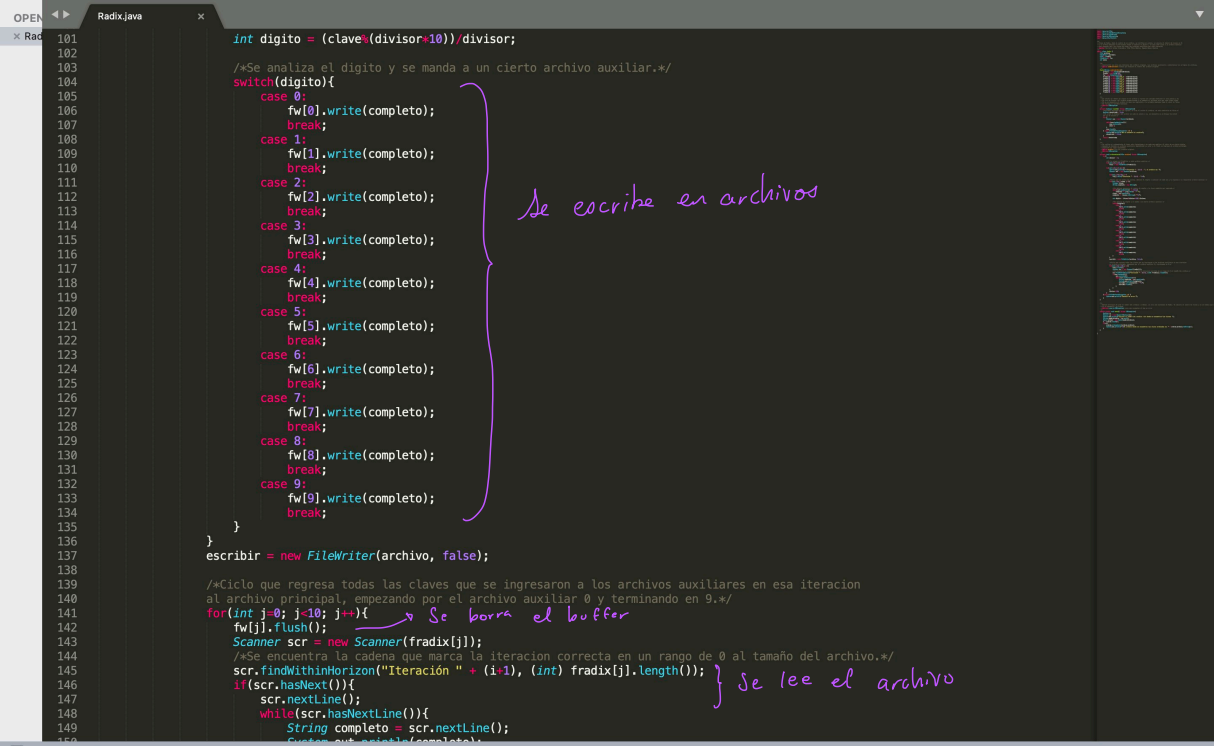
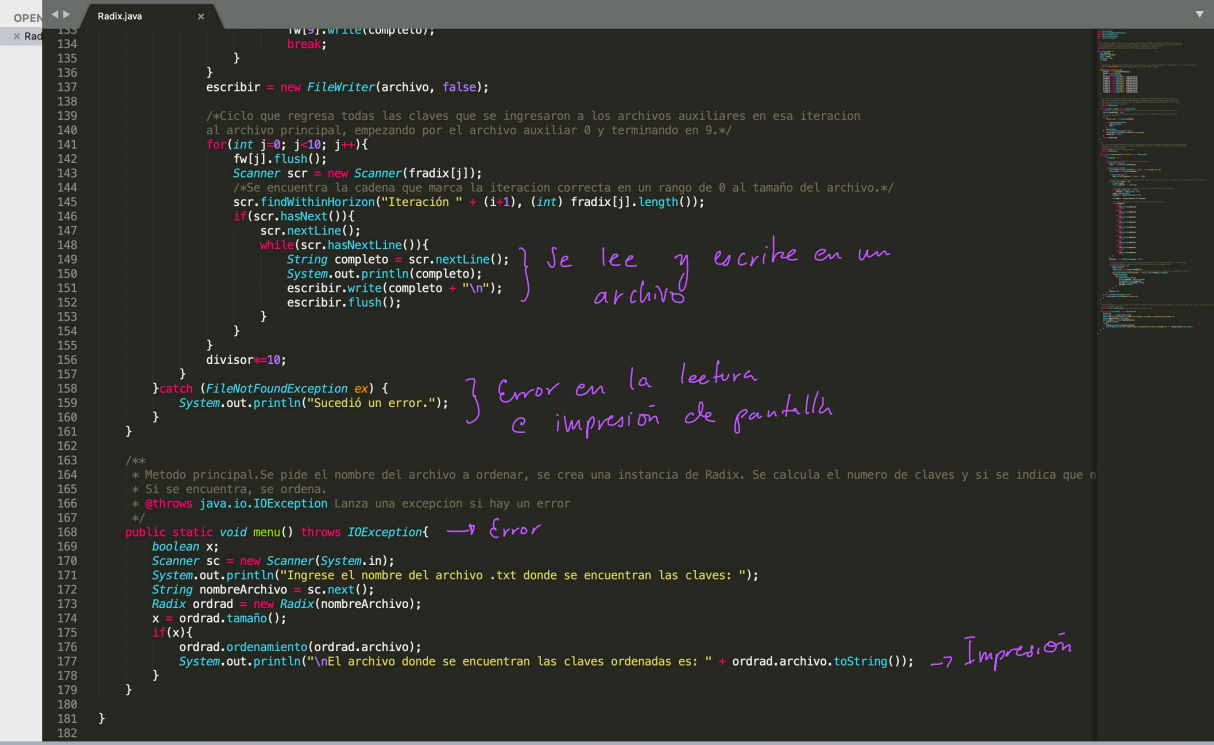


Line 11, Column 85 Spaces: 4 Java


Radix.java
boolean encontrado = true;
/*Al crear el Scanner del archivo se sabe si existe o no, se enceuntra en un bloque try-catch
por si no existe.*/
try {
Scanner sca = new Scanner(archivo);
while(sca.hasNextLine()){
sca.nextLine();
size++;
}
sca.close();
} catch (FileNotFoundException ex) {
System.out.println("No se encontró el archivo");
encontrado = false;
}
return encontrado;
}
/**
* Se realiza el ordenamiento al tener seis iteraciones y en cada una analizar el valor de un cierto digito,
* después se dividen en archivos auxiliares dependiendo su valor y al final se regresan al archivo principal
* siguiendo un orden ascendente.
* @param archivo File del archivo original.
* @throws IOException
*/
private void ordenamiento(File archivo) throws IOException{
try{
int divisor = 1;
/*Se le asigna un FileWriter a cada archivo auxiliar.*/
for(int i=0; i<10 ;i++)
fw[i] = new FileWriter(fradix[i]);
for(int i=0; i<6 ;i++){
System.out.println("\nIteración " + (i+1) + ", el archivo es: ");
Scanner scf = new Scanner(archivo);
for(int j=0;j<10; j++)
fw[j].write("Iteración " + (i+1) + "\n");
/*Ciclo que lee todas las claves, obtiene el dígito a analizar de cada una y lo ingresa a su respectivo archivo auxiliar.*/
for(int j=0; j<size; j++){
Integer clave;
String completo = new String();
/*Se obtiene encuentra el numero de cuenta y la clave completa por separado.*/
while(scf.hasNextInt() == false)
completo += (scf.next() + " ");
clave = scf.nextInt();
completo += (clave.toString()+"\n");
Se está leyendo el archivo
Error en la lectura (llamada al sistema)
Una impresión es una llamada
Misma excepción
Impresión de pantalla
Se escribe en un archivo
Line 11, Column 85
Spaces: 4
Java


Radix.java
int digito = (clave%(divisor*10))/divisor;
/*Se analiza el digito y se manda a un cierto archivo auxiliar.*/
switch(digito){
case 0:
fw[0].write(completo);
break;
case 1:
fw[1].write(completo);
break;
case 2:
fw[2].write(completo);
break;
case 3:
fw[3].write(completo);
break;
case 4:
fw[4].write(completo);
break;
case 5:
fw[5].write(completo);
break;
case 6:
fw[6].write(completo);
break;
case 7:
fw[7].write(completo);
break;
case 8:
fw[8].write(completo);
break;
case 9:
fw[9].write(completo);
break;
}
}
escribir = new FileWriter(archivo, false);
/*Ciclo que regresa todas las claves que se ingresaron a los archivos auxiliares en esa iteracion
al archivo principal, empezando por el archivo auxiliar 0 y terminando en 9.*/
for(int j=0; j<10; j++){
fw[j].flush();
Scanner scr = new Scanner(fradix[j]);
/*Se encuentra la cadena que marca la iteracion correcta en un rango de 0 al tamaño del archivo.*/
scr.findWithinHorizon("Iteración " + (i+1), (int) fradix[j].length());
if(scr.hasNext()){
scr.nextLine();
while(scr.hasNextLine()){
String completo = scr.nextLine();
Se escribe en archivos
Se borra el buffer
Se lee el archivo
Line 11, Column 85
Spaces: 4
Java


Radix.java
fw[9].write(completo);
break;
}
}
escribir = new FileWriter(archivo, false);
/*Ciclo que regresa todas las claves que se ingresaron a los archivos auxiliares en esa iteracion
al archivo principal, empezando por el archivo auxiliar 0 y terminando en 9.*/
for(int j=0; j<10; j++){
fw[j].flush();
Scanner scr = new Scanner(fradix[j]);
/*Se encuentra la cadena que marca la iteracion correcta en un rango de 0 al tamaño del archivo.*/
scr.findWithinHorizon("Iteración " + (i+1), (int) fradix[j].length());
if(scr.hasNext()){
scr.nextLine();
while(scr.hasNextLine()){
String completo = scr.nextLine();
System.out.println(completo);
escribir.write(completo + "\n");
escribir.flush();
}
}
}
divisor*=10;
}
}catch (FileNotFoundException ex) {
System.out.println("Sucedió un error.");
}
}
/**
* Metodo principal.Se pide el nombre del archivo a ordenar, se crea una instancia de Radix. Se calcula el numero de claves y si se indica que n
* Si se encuentra, se ordena.
* @throws java.io.IOException Lanza una excepcion si hay un error
*/
public static void menu() throws IOException{
boolean x;
Scanner sc = new Scanner(System.in);
System.out.println("Ingrese el nombre del archivo .txt donde se encuentran las claves: ");
String nombreArchivo = sc.next();
Radix ordrad = new Radix(nombreArchivo);
x = ordrad.tamaño();
if(x){
ordrad.ordenamiento(ordrad.archivo);
System.out.println("\nEl archivo donde se encuentran las claves ordenadas es: " + ordrad.archivo.toString());
}
}
}
Se lee y escribe en un archivo
Error en la lectura e impresión de pantalla
Error
Impresión
Line 11, Column 85
Spaces: 4
Java