**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e Ingenierías**

**Taller de programación de sistemas**

**Reporte # 2**

**ALUMNO:**

Oscar Alberto Palomares López Sección D15

Quiñones Nunñez Karla Patricia Sección D07

**Descripción de las variables y funciones utilizadas**

Para esta Segunda práctica seguimos empleando las 2 clases

* **EnsambladorHC12Raw:**

En esta clase vienen las funciones principales que analizan el texto archivo P1ASM y crean los archivos de error en caso de errores de la práctica1.

* **EnsambladorHC12UI:**

En está clase se contiene las funciones donde se genera la UI y se manda a llamar a las funciones de la clase EnsambladorHC12Raw

**Variables nuevas de la clase EnsambladorHC12Raw:**

Como nuevas variables tenemos:

private String contenidoTABOPtxt;

private final String TABOP = "/TABOP.TXT";

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

* **Private String[] contenidoTABOP:**

Contiene el contenido en formato String del archivo TABOP.TXT

* **Private final String TABOP = “/TABOP.TXT”:**

Contiene el nombre del archivo por defecto del TABOP.TXT

**Métodos nuevos:**

**public void inicializarVariables(){ :** Ahora en este metodo se agrega el setContenidoTABOP() que lee el lugar en donde debe estar situado el TABOP usando la ruta del P2ASM.TXT

**String analizarLinea(int LINE\_NUMBER, String[] palabras) { Ahora en este método en el momento que se agrega un CODOP se agrega también los datos u errores que tenga el CODOP con el método buscarEnTABOP()**

* **public String buscarEnTABOP(String palabra, boolean hasOperando) :**  Analiza la el CODOP que viene en la variable palabra mandandose a llamar un ciclo que analiza todo el contenido de la variable contenidoTABOP y comparando si este codop existe en el archivo TABOP.TXT además lo compara de manera universal usando el metodo de java toUpperCase() y en caso que exista en la linea del TABOP.TXT se crea un arreglo de Strings llamado tokens en el que compara si el CODOP contiene la palabra SI o NO para en caso que deba o no contener un OPERANDO se genere el error correspondiente, en caso de no haber errores se regresará en pantalla la información relacionada al CODOP. Finalmente en caso que no se haya encontrado el CODOP en el TABOP se manda el error correspondiente

**Métodos getter y setter:**

* Son los getter y setter de las variables de esta clase.

public void setContenidoTABOPtxt(String contenidoTABOPtxt) {

this.contenidoTABOPtxt = contenidoTABOPtxt;

}

public void setContenidoTABOPtxt(String contenidoTABOPtxt) {

this.contenidoTABOPtxt = contenidoTABOPtxt;

}

public String getContenidoTABOPtxt() {

return this.contenidoTABOPtxt;

}

**Clase public class EnsambladorHC12UI extends javax.swing.JFrame :**

En esta clase se agregó una nueva ventana donde se muestra el TABOP.TXT.

**Nuevas Variables:**

|  |
| --- |
| private javax.swing.JLabel jLabel3; |
|  |
| private javax.swing.JScrollPane jScrollPane3; |
| ; |
| private javax.swing.JTextArea jTextArea3; |
|  |

.

**Métodos mejorados:**

* **Private void jButtonActionPerformed(Event evt)**  Ahora este método checa si existe el TABOP.TXT en la misma carpeta que el P2ASM.TXT seleccionado