

READING BY ABSTRACTION : PARSER

Constructor

Línea 31: inicializa el valor de la variable line.

Línea 32: se llama al constructor de la clase Entry para inicializar el valor de la variable en.

Línea 30-33: constructor de la clase.

insertEntry

Línea 36: pone el valor de la variable en a p.

Línea 37: se llama a la función createLine y el valor que devuelve esta se almacena en la variable line.

Línea 35-38: la función insertEntry tiene como parámetro a p de tipo Entry. Se encarga de insertar una entrada y de almacenar los datos de un usuario en la variable line, cuyo valor se puede obtener a través del método getLine().

insertLine

Línea 41: pone el valor de la variable line a l.

Línea 42: se llama a la función createEntry() y el valor de retorno se almacena en la variable llamada en.

Línea 40-43: la función insertLine() tiene como parámetro a la variable l de tipo string y crea un objeto de tipo Entry con los mismos datos que se pueden obtener a través del método getEntry().

createEntry

Línea 54: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada nombre.

Línea 55: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada apellido.

Línea 56: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada dirección.

Línea 57: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada población.

Línea 58: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada provincia.

Línea 59: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada código.

Línea 60: se llama al constructor String() para crear un string vacío y se almacena en la variable tipo string llamada teléfono.

Línea 61: se crea la variable tipo int llamada anno con el valor de 0.

Línea 62: se crean las variables tipo int llamadas posNom, posDir, posPob, posProv, posCod, posTfno, posAnno y longitud.

Línea 63: se crea la variable booleana llamada auxb con el valor de true.

Línea 64: se llama al constructor Entry() para inicializar la variable auxp de tipo Entry.

Línea 66: se llama a la función length() que devuelve la longitud de la variable llamada line y se almacena en la variable longitud.

Línea 67: se llama a la función indexOf con el parámetro NOM. Esta función devuelve el índice que indica la posición inicial del String NOM. El valor devuelto se almacena en posNom.

Línea 68: se llama a la función indexOf con el parámetro APE. Esta función devuelve el índice que indica la posición inicial del String APE. El valor devuelto se almacena en posApe.

Línea 69: se llama a la función indexOf con el parámetro DIR. Esta función devuelve el índice que indica la posición inicial del String DIR. El valor devuelto se almacena en posDir.

Línea 70: se llama a la función `indexOf` con el parámetro POB. Esta función devuelve el índice que indica la posición inicial del String POB. El valor devuelto se almacena en `posPob`.

Línea 71: se llama a la función `indexOf` con el parámetro PRO. Esta función devuelve el índice que indica la posición inicial del String PRO. El valor devuelto se almacena en `posProv`.

Línea 72: se llama a la función `indexOf` con el parámetro COD. Esta función devuelve el índice que indica la posición inicial del String COD. El valor devuelto se almacena en `posCod`.

Línea 73: se llama a la función `indexOf` con el parámetro TFN. Esta función devuelve el índice que indica la posición inicial del String TFN. El valor devuelto se almacena en `posTfno`.

Línea 74: se llama a la función `indexOf` con el parámetro ANNO. Esta función devuelve el índice que indica la posición inicial del String ANNO. El valor devuelto se almacena en `posAnno`.

Línea 77: se llama a la función `trim()` que elimina los espacios en blanco que se encuentren entre los índices indicados en la función `substring(posNom+7,posApe-2)` y se almacena en la variable string llamada `nombre`.//+7 porque el nombre está compuesto por 7 caracteres y -2 para señalar a la posición anterior a la inicial de la variable `apellido`.

Línea 78: se llama a la función `trim()` que elimina los espacios en blanco que se encuentren entre los índices indicados en la función `substring(posApe+11,posDir-2)` y se almacena en la variable string llamada `apellido`.//+11 porque el apellido está compuesto por 11 caracteres y -2 para señalar a la posición anterior a la inicial de la variable `direccion`.

Línea 79: se llama a la función `trim()` que elimina los espacios en blanco que se encuentren entre los índices indicados en la función `substring(posDir+11,posPob-2)` y se almacena en la variable string llamada `dirección`.//+11 porque la dirección está compuesta por 11 caracteres y -2 para señalar a la posición anterior a la inicial de la variable `población`.

Línea 80: se llama a la función `trim()` que elimina los espacios en blanco que se encuentren entre los índices indicados en la función `substring(posPob+11,posProv-2)` y se almacena en la variable string llamada `población`.//+11 porque la población está compuesta por 11 caracteres y -2 para señalar a la posición anterior a la inicial de la variable `provincia`.

Línea 81: se llama a la función `trim()` que elimina los espacios en blanco que se encuentren entre los índices indicados en la función `substring(posProv+11,posCod-2)` y se almacena en la variable string llamada `provincia`.//+11 porque la provincia está

compuesta por 11 caracteres y -2 para señalar a la posición anterior a la inicial de la variable código.

Línea 82: se llama a la función trim() que elimina los espacios en blanco que se encuentren entre los índices indicados en la función substring(posCod+7,posTfno-2) y se almacena en la variable string llamada codigo.//+7 porque el código está compuesto por 7 caracteres y -2 para señalar a la posición anterior a la inicial de la variable telefono.

Línea 83: se llama a la función trim() que elimina los espacios en blanco que se encuentren entre los índices indicados en la función substring(posTfno+9,posAnno-2) y se almacena en la variable string llamada telefono.//+9 porque el telefono está compuesto por 9 caracteres y -2 para señalar a la posición anterior a la inicial de la variable anno.

Línea 84: se llama a la función trim() que elimina los espacios en blanco que se encuentren entre los índices indicados en la función substring(posAnno+10,longitud). Se llama también a la función parseInt() de la clase Integer que convierte el string en una variable tipo int y se almacena en la variable int llamada anno.//+10 porque el anno está compuesto por 10 caracteres y longitud para señalar a la posición final de la variable line.

Línea 85: en la variable auxb se almacena el valor de true.

Línea 76-86: si las variables pos tienen un valor distinto de -1 se almacenan los datos del usuario en todas las variables y auxb adquiere el valor de true.

Línea 87: si alguna de las variables pos tiene el valor de -1, la variable auxb toma el valor de false.

Línea 89: se llama a la función setName que tiene de parámetro la variable nombre, que es el que obtiene la variable de tipo Entry auxp.

Línea 90: se llama a la función setSurname que tiene de parámetro la variable apellido, que es el que obtiene la variable de tipo Entry auxp.

Línea 91: se llama a la función setAddress que tiene de parámetro la variable direccion, que es la que obtiene la variable de tipo Entry auxp.

Línea 92: se llama a la función setCity que tiene de parámetro la variable poblacion, que es la que obtiene la variable de tipo Entry auxp.

Línea 93: se llama a la función setCountry que tiene de parámetro la variable provincia, que es la que obtiene la variable de tipo Entry auxp.

Línea 94: se llama a la función setZip que tiene de parámetro la variable codigo, que es la que obtiene la variable de tipo Entry auxp.

Línea 95: se llama a la función setTelephone que tiene de parámetro la variable telefono, que es la que obtiene la variable de tipo Entry auxp.

Línea 96: se llama a la función setBirthYear que tiene de parámetro la variable anno, que es la que obtiene la variable de tipo Entry auxp.

Línea 88-97: si la variable booleana auxb tiene el valor de true, se obtienen todos los datos del usuario, que formarán los datos de la variable tipo Entry auxp.

Línea 98: se devuelve el valor de la variable tipo Entry auxp.

Línea 53-99: la función createEntry() crea una entrada y la devuelve en una variable tipo Entry.

createLine

Línea 102: se llama a la función getName() y el nombre que se obtiene se almacena en la variable string llamada nombre.

Línea 103: se llama a la función getSurname() y el apellido que se obtiene se almacena en la variable string llamada apellido.

Línea 104: se llama a la función getAddress() y la dirección que se obtiene se almacena en la variable string llamada direccion.

Línea 105: se llama a la función getCity() y la poblacion que se obtiene se almacena en la variable string llamada poblacion.

Línea 106: se llama a la función getCountry() y la provincia que se obtiene se almacena en la variable string llamada provincia.

Línea 107: se llama a la función getZip() y el codigo que se obtiene se almacena en la variable string llamada codigo.

Línea 108: se llama a la función getTelephone() y el telefono que se obtiene se almacena en la variable string llamada telefono.

Línea 109: se llama a la función getBirthYear() y el año de nacimiento que se obtiene se almacena en la variable int llamada anno.

Línea 110: se crea una variable tipo string y se inicializa.

Línea 113: se almacena en aux2 el string formado por NOM, APE Y DIR con sus respectivos valores y un '.' de por medio.

Línea 114:se almacena el valor de aux2 concatenado con el string formado por POB,PRO,COD con sus respectivos valores y un '.' de por medio.

Línea 115: se almacena el valor de aux2 concatenado con el string formado por TFN y ANNO con sus respectivos valores y un '.' de por medio.

Línea 112-116: se llama a la función hasData() para obtener los datos de la variable tipo Entry llamada en. Si tiene datos, se almacenan en aux2.

Línea 117: se devuelve el valor de la variable aux2.

Línea 101-118: se crea una línea de entrada.

