ADRIAN MAXIMILIANO MUNOZ MARTINEZ

Laboratorio POO  Sabado 7 am – 9am

PRactica 4: STL(String)

**Introducción**

Se desarrollará una aplicación que mediante una clase concreta SimpleString realizará manipulaciones y comparaciones en cadenas de caracteres como comparar, ordenar, contar, cortar y unir, asi como una clase Numbers que determinara cual numero es el mas grande de los ingresados y que también calcule el iva basado en el precio de un producto y la cantidad de productos comprados.

**Objetivo**

El objetivo de la practica es reforzar nuestro manejo de la sintaxis en Java así como de las funcionalidades orientadas a objetos que posee.

**Practica**

Test.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package practica4;

/\*\*

\*

\* @author FCFM

\*/

public class Practica4 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

SimpleString cadena = new SimpleString("the quick brown fox jumps over the lazy dog");

System.out.println(cadena.compare("sa"));

System.out.println(cadena.order());

System.out.println(cadena.count());

System.out.println(cadena.split(7));

System.out.println(cadena.join("123"));

System.out.println(Numbers.compare(1,2));

System.out.println(Numbers.compare(-1, 2, 3));

System.out.println(Numbers.iva(50.0,2));

System.out.println(Numbers.iva(2, 50.0));

}

}

SimpleString.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package practica4;

/\*\*

\*

\* @author FCFM

\*/

public class SimpleString {

private String str;

public SimpleString(){

this("");

}

public SimpleString(String cadena){

setStr(cadena);

}

public int count(){

int x = 0;

for(char a:str.toCharArray()){

x++;

}

return x;

}

public boolean compare(String other){

return str==other;

}

public char[] order(){

char[] word\_array= str.toCharArray();

return quickSort(word\_array, 0, this.count()-1);

}

public char[] quickSort(char[] arr, int low, int high){

int i = low;

int j = high;

int pivot =(arr[i]+arr[j])/2;

while (i<j){

while(arr[i]<pivot){

i++;

}

while(arr[j]>pivot){

j--;

}

if(j>=i){

char x = arr[i];

arr[i] = arr[j];

arr[j] = x;

i++;

j--;

}

}

if(low<j){

quickSort(arr, low, j);

}

if(high>i){

quickSort(arr, i, high);

}

return arr;

}

public char[] split(int limit) {

char[] change = new char[limit];

int count = 0;

for(char a:str.toCharArray()) {

if(count<limit) {

change[count] = a;

}

count++;

}

return change;

}

public String join(String str2) {

return str+str2;

}

/\*\*

\* @return the str

\*/

public String getStr() {

return str;

}

/\*\*

\* @param str the str to set

\*/

public void setStr(String str) {

this.str = str;

}

}

Numbers.java

package practica4;

public class Numbers {

public static int compare(int a, int b) {

if(a>=b) {

return a;

}else {

return b;

}

}

public static int compare(int a, int b, int c) {

return compare(compare(a,b),c);

}

public static double iva(double precio, int cantidad) {

double tax = (precio\*cantidad\*0.14);

return tax;

}

public static double iva(int cantidad, double precio) {

return iva(precio, cantidad);

}

}

**Screenshots**

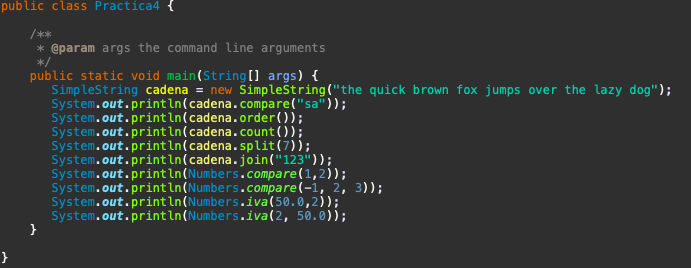


Imagen que contiene captura de pantalla

Descripción generada automáticamente

**Explicación**

Se creo una aplicación con una clase SimpleString que recibe una cadena en su método constructor, tiene la capacidad de comparar esta cadena con otra, ordenar la cadena que tiene como atributo principal, calcular su longitud, hacer una partición y concatenarse con otra cadena devolviendo el resultado de la concatenación.

La cadena es ordenada convirtiéndola primeramente en un arreglo de caracteres y después tratar estos caracteres como números en un algoritmo llamado quickSort para ordenarlos de acuerdo con su valor numérico ASCII.

También existe una clase Numbers que realiza comparaciones entre números y calcula el iva de una compra basada en el numero de objetos comprados y el precio individual de estos. Numbers no necesita tener atributos por lo que todos sus métodos son estáticos.

**Conclusión**

Java es un lenguaje que ofrece mucha funcionalidad y versatilidad ya que permite la creación de clases y la sobre carga de métodos lo que permite que el código sea mas flexible, fácil de modificar y se pueda reusar en otros aplicaciones.