RBtree - UML - Design - Cosimo Gonnelli

CLL where each node has a concept with 2 arrays (one for c++ and one for Java) for syntax comparison

CLL protected c_node rear; protected int addInfo(syntaxMap toAdd); protected int displayCLL(c_node current); has a public CLL(); public int displayCLL();

protected d_node next; protected information[] array1; protected information[] array2; protected int index; protected int. addInfo(syntaxMap toAdd); protected void display();

c_node

public c_node(); public c_node(int size); public c_node goNext(); public void setNext(c_node next);

protected I_node head;

toAdd);

protected String topicName;

abstract class information protected String name; protected String type;

protected String topicName; protected int compare(information toCompare); protected int compare(String toCompare); protected int compareType(String to Compare);

has a

has a

. _ _ _ _ _ _ _ _ _ _

public information(); public information(String name, String type, String topicName)

l_node

public information(information toAdd); abstract public int display();

has a

protected I_node next; protected infomation myInfomration;

public I_node(); public I_node goNext(); public void setNext(I_node next); public void display();

linuxTool

is a

is a

is a

protected String address; protected String evaluation;

public website();

public int display();

public String getAddress();

protected String desctiption; protected String command;

public linuxTool(); public linuxTool(String name, String type, String topicName, String description, String command); public linuxTool(information toAdd); public linuxTool(linuxTool toAdd); public int display();

website

public wedsite(String name, String type, String topicName, String address, String evaluation);

public website(information toAdd); public website (website toAdd);

RBtree

RBtree of nodes where each node is a topic name that holds a LLL of websites of the same topic

protected rb_node root; protected rb_node destroyAll(rb_node root); protected rb_node insertTopic(rb_node parent, rb node root, topic toAdd); protected int rblnsertFixup(rb_node newNode); protected in leftRotate(rb_node node); protected in rightRotate(rb_node node); protected rb_node findMatch(rb_node root, String topiciname); protected int rbDelete*rb_node toDelete); protected int rbTransplant(rb_node x. rb_node y); protected int rbDeleteFixup(rb_node node); protected rb node tMinimum(rb node node);

protected rb_ addInfo(rb_node root, information toAdd,

public RBtree(); public int destroyAll(); public rb_node findMatch(String topicName) public int rbDelete(String topicName); public int addInfo(information toAdd, String topicName); public int displayAll(); public int displayRB();

protected int displayRB(rb_node root);

string topicName);

has a

protected rb_node left; protected rb_node right; protected rb_node parent; protected topic myTopic; protected char color; has a rb_node (); ---public void setColor(char color); public void setLeft(rb node left); public void setRight(rb_node. right); public void setParent(rb_node parent); public rb_node goLeft(); public rb_node goRight(); public rb_node getParent(); public char getColor(); public int compare(rb_node toCompare); public void display();

rb_node

protected int compareTopic(topic toCompare); protected int compareTopic(String topicN); protected int compareTopicName(information toCompare); public topic(); public topic(String topicName); public int destroyLLL(); public int insertInfo(information toAdd); public in IDelete(String infoName); public int display();

category (topic name)

protected I_node insertInfo(I_node head, information

protected IDelete(I_node head, String infoName);

protected I_node destroyLLL(I_node head);

protected int display(I_node head);

synataxMap

protected String cPlusPlus; protected String java;

public syntaxMap(); public syntaxMap(String name, String type, String topicName, String cPlusPlus, String java); public syntaxMap(information toAdd); public syntaxMap(syntaxMap toAdd); public int display();