污点池类结构初步设计

2016.5.8

欧锦荣

1. 链表节点结构TaintedNode

char name[20]; //用于存储变量名

int lineno; //用于存储该变量被污染时所在的行号

int field; //用于存储变量作用域，变量类型有待商讨

TaintedNode \*next; //指向下一个节点的指针

作为污点池中存储的元素的最小单位，计划使用开散列的形式存储被污染变量。

1. 污点池类TaintedPool

类成员

TaintedNode \*table[HASHSIZE]; //private类型变量，用于存储被污染变量的开散列表，大小待定

成员函数

TaintedNode \*checkid(char str[]); //查找名字为str的变量是否在污染池中，如果存在则返回//该变量在污染池中的指针，否则返回空，成员属性为public

TaintedNode \*insertnode(char str[]); //将名字为str的变量插入到污染池中，插入成功后返回指//向该新节点的指针，否则返回NULL，成员属性为public

//调用该函数时会自动调用checkid()函数

void deletenode(char str[]); //将名字为str的节点从污染池中删除，成员属性为public

void printtable( ); //打印污染池中的内容，成员属性为public

污点池类，该类为程序中存储被污染的变量的数据结构，考虑到某个变量被污染后再被未被污染的变量赋值的情况，除了查找插入打印操作外，引入了删除操作。