COMPILE TIME DATA STRUCTURE FOR OEXPL

Class Table

OExpL compilation requires only one additional data structure - the class table. The class table stores information pertaining to all the classes declared in an OExpL program. For a class it stores member fields, member functions, name of the class and parent class pointer.

Structure

The structure of Class Table(CT) is as follows:

```
struct Classtable {
        char *Name;                             //name of the class
        struct Fieldlist *Memberfield;          //pointer to Fieldlist
        struct Memberfunclist *Vfuncptr;        //pointer to Memberfunclis
        struct Classtable *Parentptr;           //pointer to the parent's 
        int Class_index;                        //position of the class in
        int Fieldcount;                         //count of fields
        int Methodcount;                        //count of methods
        struct Classtable *Next;                //pointer to next class ta
};
```

classtable.h hosted with ❤ by **GitHub** **view raw**

✣NOTE:

Memberfield list is used to store the information regarding the type, name, fieldindex and type of class of all the member fields of that class.

```
struct Fieldlist{
        char *Name;                     //name of the field
        int Fieldindex;                 //position of the field
        struct Typetable *Type;         //pointer to typetable
        struct Classtable *Ctype;       //pointer to the class containing
        struct Fieldlist *Next;         //pointer to next fieldlist entry
};
```

MemberFieldlist.h hosted with ❤ by **GitHub** **view raw**

Memberfunc list is used to store the information regarding the type, name of the function, argument list, it's flabel and it's position.

```
struct Memberfunclist {
        char *Name;                      //name of the member function in
        struct Typetable *Type;          //pointer to typetable
        struct Paramstruct *paramlist;   //pointer to the head of the for
        int Funcposition;                //position of the function in th
        int Flabel;                      //A label for identifying the st
        struct Memberfunclist *Next;     //pointer to next Memberfunclist
};
```

MemberFunclist.h hosted with ❤ by **GitHub** **view raw**

Associated Methods

- struct Classtable* CInstall(char *name,char *parent_class_name) : Creates a class table entry of given 'name' and extends the fields and the methods of parent class.
- struct Classtable* CLookup(char *name) : Search for a class table entry with the given 'name', if exists, return pointer to class table entry else return NULL.
- void Class_Finstall(struct Classtable *cptr, char *typename, char *name) : Installs the field into the given class table entry which is given as an argument.
- void Class_Minstall(struct Classtable *cptr, char *name, struct Typetable *type, struct Paramstruct *Paramlist) : Installs the method into the given class table entry which is given as an argument.

Illustration

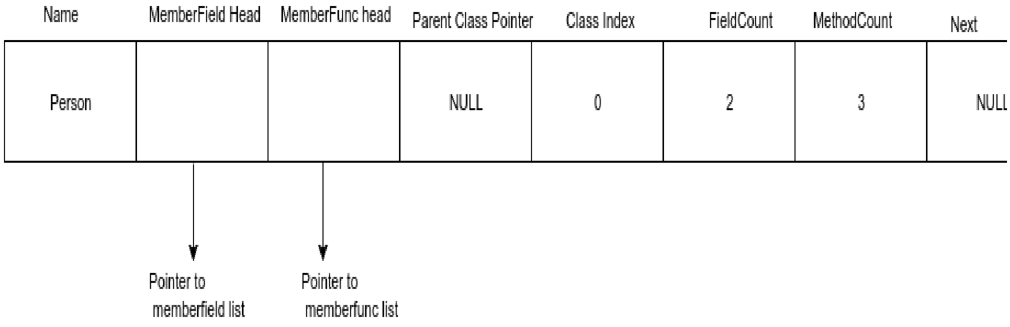
Here is an example illustrating it.

```
class
Person{
        decl
                str name;
                int age;
                int printDetails();
                str findName();
                int createPerson(str name, int age);
        enddecl
        int printDetails(){
                decl
                enddecl
                begin
                        write(self.name);
                        write(self.age);
                        return 1;
                end
        }
        str findName(){
                decl
                enddecl
                begin
                        return self.name;
                end
        }
        int createPerson(str name, int age){
                decl
                enddecl
                begin
                        self.name=name;
                        self.age=age;
                        return 1;
                end
        }
}     /*end of Person class */
Student extends Person{

        decl
                int rollnumber;               /*  The members name and a
                str dept;
                int printDetails();
                int createPerson(str name, int age,int rollNo, str dept)
        enddecl
        int createPerson(str name, int age,int rollNo, str dept){  /*  c
                decl
                enddecl
                begin
                        self.name =name;
                        self.age = age;
                        self.rollNo = rollNo;
                        self.dept = dept;
                        return 1;
                end
        }
        int printDetails(){  /* This function is also overridden in the
                decl
                enddecl
                begin
                        write(self.name);
                        write(self.age);
                        write(self.rollnumber);
                        write(dept);
                        return 1;
                end
        }         /**  The derived class inherits the findName() functio
}  /* end of student class */
endclass
```

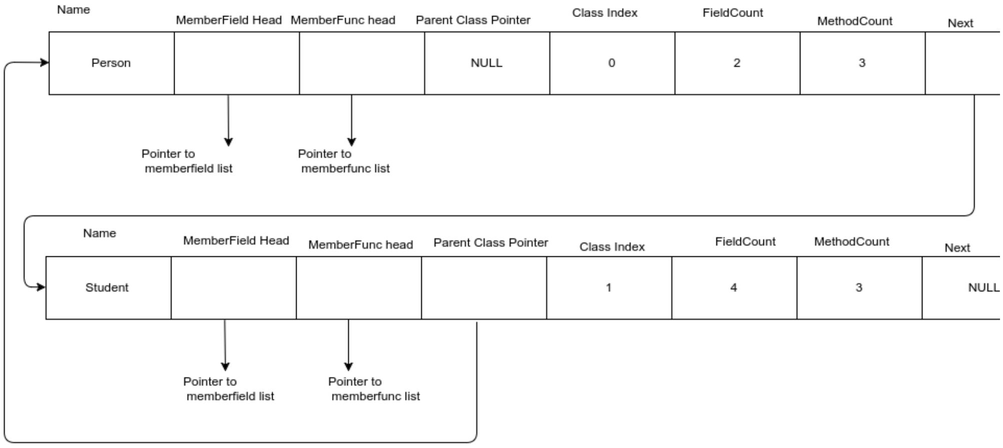
Inheritance.txt hosted with ❤ by **GitHub** **view raw**

1. As soon as the compiler encounters the class name, it installs the class name and the parent class name if present into the class table. Subsequently, If there is an extension to the parent class, all the member fields and methods of parent class are inherited. Following is how class table looks when class *Person* is installed.



Name	MemberField Head	MemberFunc head	Parent Class Pointer	Class Index	FieldCount	MethodCount	Next
Person			NULL	0	2	3	NULL

2. Following is how class table looks when class *Student* is installed.



Github

Contributed By : J.Ritesh
J.Phani
Koushik
M.Jaya
Prakash

Home | About