

La Structure de Données Lofig

La structure de données `{{Lofig}}` est conçue pour représenter les *netlists* en mémoire. Les attributs des différents objets constituant cette structure ont été présentés en cours. Nous allons maintenant compléter avec les déclarations des fonctions membres.

Fonctions membres de la class Locon

Les constructeurs :

- Un `Locon` peut appartenir, soit à un modèle (`Lofig`), soit à une instance (`Loins`). Il y aura donc deux constructeurs correspondant à chacune de ces possibilités. On fournit en outre son nom et sa direction. Le type sera déduit à partir du constructeur appelé.
 - ◆ `Locon ( Lofig*, const std::string& name, unsigned int dir );`
 - ◆ `Locon ( Loins*, const std::string& name, unsigned int dir );`

Les accesseurs :

- `std::string getName ();`
- `Losig* getSignal ();`
- `Lofig* getModel ();`
- `Loins* getInstance ();`
- `unsigned int getDirection();`

Les modificateurs (*mutators*, en VO) :

- Un `Locon` peut (doit) être associé à un signal. On pourra le faire de deux façons différentes, soit en donnant explicitement un pointeur sur le `Losig`, soit en indiquant simplement le nom du signal.
 - ◆ `void setSignal ( Losig* );`
 - ◆ `void setSignal ( const std::string& );`
- Positionnement de la direction :
 - ◆ `void setDirection ( unsigned int );`

De plus, le code de trois fonctions membres vous sont fournies pour pouvoir afficher l'objet dans un flux. Dans `Locon.h` :

```
class Locon {
public:
    static std::string    typeToString ( unsigned int );
    static std::string    dirToString  ( unsigned int );
public:
    void                  xmlDrive     ( std::ostream& );
};
```

Dans `Locon.cpp` :

```
#include "Indentation.h"

/// ...

std::string Locon::typeToString ( unsigned int type )
{
    switch ( type ) {
        case Internal: return "Internal";
        case External: return "External";
    }
}
```

```

    }
    return "Unknown";
}

```

```

std::string Locon::dirToString ( unsigned int direction )
{
    switch ( direction ) {
        case In:      return "In";
        case Out:     return "Out";
        case Inout:   return "Inout";
        case Tristate: return "Tristate";
        case Transcv: return "Transcv";
    }
    return "Unknown";
}

```

```

void Locon::xmlDrive ( std::ostream& o )
{
    o << indent << "<Locon name=\"" << _name << "\"";
    o << " signal=\"" << ((_signal != NULL) ? _signal->getName() : "NULL") << "\"";
    if ( _type == External ) o << " model=\"" << getModel    ()->getName() << "\"";
    else                    o << " instance=\"" << getInstance()->getName() << "\"";
    o << " dir=\"" << dirToString(_direction) << "\"";
    o << " type=\"" << typeToString(_type) << "\"/>\n";
}

```

□ □ □