

TDA

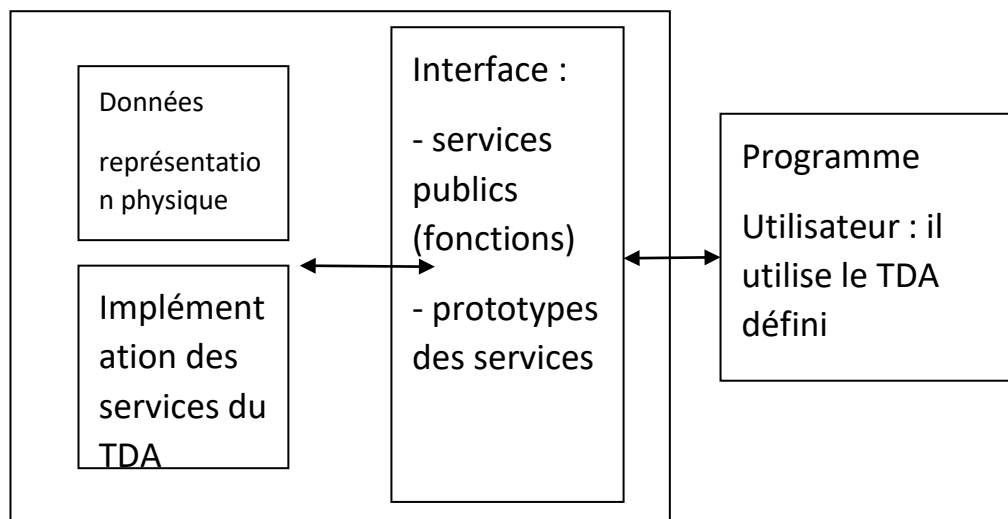
Type de données abstrait :

Un TDA est la définition d'un nouveau type incluant des données et des opérations. Les opérations publiques sont exposées à l'extérieure (elles peuvent être utilisées par des entités externes). Une entité externe n'a pas le droit d'accéder aux données qu'à travers les opérations publiques.

Encapsulation : Une entité externe n'a pas le droit d'accéder aux données qu'à travers les opérations publiques.

Mécanismes en C pour implémenter TDA (représentation physique, Module : interface et implémentation, assertion)

Module



Données ==> comment la représenter dans la mémoire?

==> **représentation physique**

Interface en C ==> fichier .h

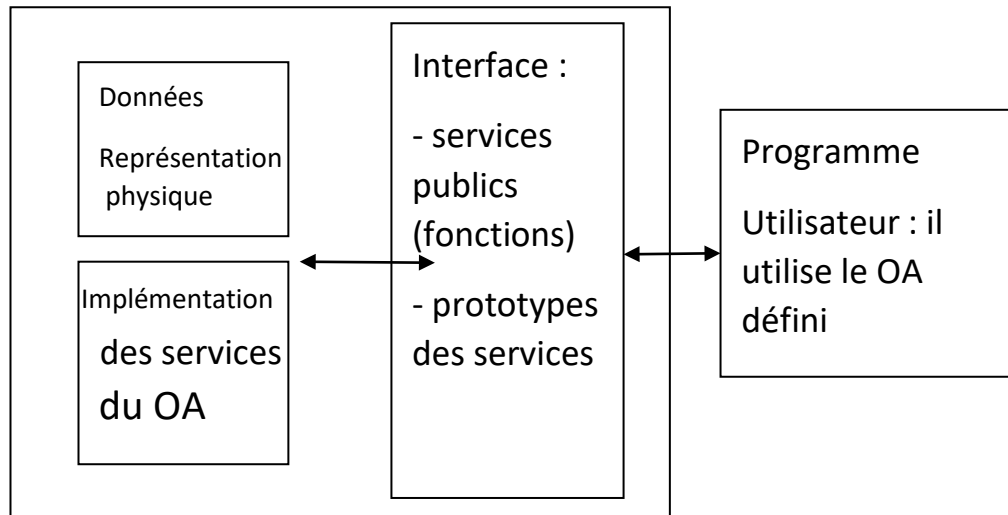
Implémentation ==> fichier .c

**Données ==> représentation physique ==> (TDA) ==> doit être
dans l'interface**

OA

Objet Abstrait:

Module



Données ==> comment la représenter dans la mémoire?

==> **représentation physique**

Prototypes ==> fichier.h

Interface en C ==> fichier .c

Implémentation ==> fichier .c

Données ==> représentation physique ==> (OA) ==> doit être

Dans l'implémentation

Remarque :

TDA ==> générateur d'exemplaires ==> manipuler plusieurs exemplaires
==> **représentation physique dans l'interface .h**

OA ==> Objet Abstrait ==> manipuler un seul exemplaire ==>
représentation physique dans le fichier .c

Remarque :

Le langage C **n'est pas le langage adéquat** pour l'implémentation des TDA.

Exemple d'un TDA déclaré en fichier.h :

Point2D.h :

```
#ifndef Point2D_H_INCLUDED
#define Point2D_H_INCLUDED

/*représentation physique*/
typedef struct Point2D
{
    double      abs;
    double ord;
}Point2D;

/*services à exporter*/

Point2D creer_point(double,double); /*service de création*/

double getAbs(Point2D); /*service de consultation*/

#endif
```

Structures de données SD

une structure de données est une manière pour organiser/stocker/mémoriser/sauvegarder des données. La manipulation des données suit des règles/propriétés à respecter.

SD et TDA

relation entre SD et TDA?

Les SDs sont une concrétisation du TDA.

**Les SDs vont être implémentés en utilisant la philosophie du TDA.
==> en utilisant les mécanismes offerts par le Langage C.**