



Chapitre 3 – Implémentation d'un LIFO
Implementierung eines LIFO

Exercice Algo 3.2

Dans cet exercice nous voulons implémenter une liste chaînée avec un comportement de LIFO (Last-In—First-Out). Cette liste doit stocker des éléments qui sont décrit par la structure suivante :

In dieser Übung wollen wir eine verkettete Liste implementieren, welche das Verhalten eines LIFO-Buffers (Last-In—First-Out) besitzt. Diese Liste soll Elemente speichern, welche von der folgenden Struktur beschrieben werden:

```
typedef struct Data_s{  
    int    sensorNumber;  
    double sensorValue;  
    long   time;  
    struct Data_s* pNext;  
} Data_t;
```

Pour pouvoir travailler avec la liste, 3 fonctions avec les 3 prototypes suivants sont à disposition.

Um mit der Liste arbeiten zu können, stehen 3 Funktionen zur Verfügung, welche die folgenden Prototypen besitzen:

```
/// @brief With this function a new list is created  
/// @param -  
/// @return pointer to the first element of the list  
Data_t* createLIFO(void);  
  
/// @brief Add a new element to list  
/// @param ppList is the pointer to the list pointer  
/// @param newElement is element to be added to the list  
/// @return true for success, false if new element could not be added to the list  
bool put(Data_t** ppList, Data_t newElement );  
  
/// @brief Retrieve an element from the list  
/// @param ppList is the pointer to the list pointer  
/// @param pElement is the pointer to store the retrieved element  
/// @return true for success, false if there was no element in the list.  
bool get(Data_t** ppList, Data_t* pElement);
```

- Adaptez ces 3 fonctions en se basant sur l'exercice de FIFO
Adaptieren Sie diese 3 Funktionen basierend auf der FIFO-Übung
- Écrivez un petit programme, qui utilise ces fonctions. Effectuez aussi des tests des situations d'exception (p.e. pop appliqué à une liste vide).
Schreiben Sie ein kleines Programm, welches diese Funktionen verwendet. Führen Sie auch Tests von Ausnahmesituationen (z.B. pop auf eine Liste, die leer ist durch).
- Adaptez **put()** pour passer un pointer pour le paramètre newElement
Adaptieren Sie put() um einen Zeiger für den Parameter newElement zu übergeben