

UPPA - année universitaire 2013/2014

Département informatique

Licence informatique 3^e année

Franck.Barbier@FranckBarbier.com

Examen de 2^e session – TOO – juin 2014

Soit une classe C++ simulant une collection LIFO (*Last-In First-Out*) c'est-à-dire que toute suppression supprime le dernier entré.

```
using namespace std;

template <typename T> class In_Out {
public:
    virtual bool empty() const = 0;
    virtual void in(const T&) = 0;
    virtual T out() = 0; // retour par copie, attention...
};

template <typename T> class LIFO : public In_Out<T> {
private:
    list<T> _representation;
public:
    bool empty() const;
    void in(const T&);
    T out();
};

template <typename T> bool LIFO<T>::empty() const {
    return _representation.empty();
}

template <typename T> T LIFO<T>::out() {
    if(empty()) throw "empty..."; // on lance une exception si vide pour signaler le problème...
    T t = _representation.back();
    _representation.pop_back(); // suppression effective
    return t;
}

template <typename T> void LIFO<T>::in(const T& t) {
    _representation.push_back(t);
}
```

Question 1 : renommer la classe *LIFO* par *FIFO* puis modifier ce code C++ pour obtenir un protocole FIFO (*First-In First-Out*).

Question 2 : écrire les classes *In_Out* et *LIFO* en Java. Dans ce contexte, on veillera à trouver quelle est la meilleure traduction de la classe C++ *In_Out* en Java.