```
#include "_restartableThread.h"
#include "_dataCopy.h"
#include <map>
using namespace std;

//! \structure  _speculativeBranch
//! \brief  struct used to encapsulate the information necessary for a speculativeBranch, mostly it's copied data.
//!
//! Limitations:
//!

struct _speculativeBranch: _restartableThread {


        //red-black trees used to keep per thread data copies, each node
        //has as key the reference to a data element in _shared_data,
        //the other variable holds a void* representing the value
        //written by the branch.
        //These per/thread copies help to prevent antidependence violations
        //and output violations (WAR, WAW)
        map <void*, _data_copy*> _copied_data;

        pthread_mutex_t _copied_data_mutex;

        _speculativeBranch(){
                pthread_mutex_init (&_copied_data_mutex, NULL);
        }

};
```