

Opis głównej struktury Bazy Danych

Opis ogólny działania:

- Struktura danych składa się z klasy „lista<T>” oraz struktury „Node<T>” gdzie T – typ danych przechowywanych przez listę
- Cała mechanika polega na liście dwukierunkowej struktur Node<T> posiadających wskaźniki na poprzedni i następny element
- Classa lista zawiera funkcje first() i last() zwracające kolejno wskaźniki do pierwszego i ostatniego elementu listy pozwala nam to w prosty sposób pętla while przejść przez wszystkie elementy listy

Opis głównych funkcji klasy lista z biblioteki list.h:

- Przykład wywołania Konstruktora Klasy dla typu zmiennej int:
 - `lista<int> L1;`
- Metoda `void lista<T>::push_back(T rekord)`
 - Działanie: wstawia nowy element na koniec listy dwukierunkowej
 - Parametry: funkcja za parametr przyjmuje nowy element dowolnego typu wcześniej zdefiniowanego podczas wywoływania konstruktora
- Metoda `void lista<T>::drop(int index)`
 - Działanie: usuwa element o podanym indexie z listy
 - Parametry: jak parametr przyjmuje zmienna index typu int
 - Throw: w przypadku wyjścia poza zakres listy funkcja throwuje wyjątek typu string
- Metoda `void lista<T>::sort(bool asc, bool (*con)(T, T, bool))`
 - Działanie: sortuje wszystkie elementy tablicy w oparciu o zmienną asc typu bool wskazującą na to czy sortowanie ma się odbyć rosnąco czy malejąco a także o funkcję con zwracającą typ bool służącą do porównania dwóch danych typu T
 - Parametry: funkcja przyjmuje dwa parametry : asc typu bool wskazujący na to czy sortowanie ma się odbyć rosnąco czy malejąco i wskaźnik do funkcji typu bool przyjmującej trzy argumenty (T,T,bool) ten argument posiada wartość domyślną:
`[] (T a,T b,bool asc){ return (asc)? a>b : a<b;}`
 - Throw: w przypadku gdy lista jest pusta wywołany zostaje wyjątek typu string
- Nadpisanie operatora `[] T operator[](const int& index);`
 - Działanie: wyszukuje komórkę Node posiadającą index podany w klamrach ([])
 - Parametry: jaka parametr przyjmuje zmienna index przez referencje typu const int
 - Return: zwraca zmienną typu T

Kod pobocznych funkcji klasy lista z biblioteki list.h:

- `template <typename T> int lista<T>::size() { return _size; }`
- `template <typename T> Node<T>* lista<T>::first() { return _first; }`
- `template <typename T> Node<T>* lista<T>::last() { return _last; }`