

Chapter 3 C++による大規模開発 ～C++標準テンプレートライブラリの利用(中級)～

3.1 検索

双方向リストは、可変長配列と同様に、std::find()関数を利用することで、リストに記録されている要素を検索することができます。次のコードはfind()関数の利用例です。

```
std::list< int > hoge;
hoge.push_back(30);
hoge.push_back(40);
hoge.push_back(50);

std::list<int>::iterator it;
// find関数は第一引数と第二引数にイテレーターを指定して、
// 検索範囲を指定する。第三引数は検索する値。
// 今回だと、hogeの先頭から終端までの範囲で、40という値を検索している。
it = std::find(
    hoge.begin(),
    hoge.end(),
    40
);

if ( it == hoge.end() ) {
    // 検索対象の値が見つからなかった場合は終端のイテレーターを返してくる。
    std::cout << "見つからなかった" << "\n";
}else{
    std::cout << "見つかった" << *it << "\n";
}
```

3.2 要素の削除

listは可変長配列と同様にerase()関数を利用することで、要素を削除することができます。erase()関数に削除したいイテレーターを渡すことで削除することができます。次のコードはerase()関数の利用例です。

```
std::list< int > hoge;
hoge.push_back(30);
hoge.push_back(40);
hoge.push_back(50);

std::list<int>::iterator it;
// hogeから40が記録されているイテレータを検索する。
it = std::find(
    hoge.begin(),
    hoge.end(),
    40
);
```

```
if ( it != hoge.end() ) {  
    // 見つけたので削除  
    hoge.erase( it );  
}
```