

課題

課題1

標準ライブラリコンテナ(`std::vector`、`std::list`、`std::deque`など)を使用し、その要素を出力する関数テンプレート「`print_elements`」を作成します。2種類以上のコンテナを使用して関数をテストしてください。

課題2

スタックの動作を模倣したクラステンプレート「`Stack`」を作成します。このクラスでは次のメンバ関数を使用してください。

- `void push(const T& value)`: スタックに値を入れます(プッシュします)。
- `void pop()`: スタックから一番上の値を取り出します(ポップします)。
- `T top() const`: スタックの一番上の値を返します。
- `bool is_empty() const`: スタックが空の場合に返します。

スタックの基本データ構造として、標準ライブラリコンテナ(`std::vector`、`std::list`、`std::deque`など)を使用します。2種類以上のコンテナを使用してクラスをテストしてください。

課題3

スタックの基本データ構造として`std::vector`、`std::list`、`std::deque`を使用した場合のパフォーマンスを比較します。スタックで一連のプッシュ操作とポップ操作を実行する小さなプログラムを作成し、コンテナの種類ごとに実行時間を計ります。このアプリケーションに最適なコンテナを判断し、その理由を説明してください。コンパイラがコードを最適化して無意味なベンチマークにならないように、必ず`-O0`を指定してコンパイルしてください。

課題4

名前と電話番号のリストを含むテキストファイルを読み取るプログラムを設計します。プログラムでは、`std::map`コンテナに名前と電話番号を格納してください。次の機能を実装します。

- 名前と電話番号をマップに追加します。
- 名前とそれに紐づく電話番号をマップから削除します。
- 特定の名前で電話番号を検索します。
- すべての名前と電話番号をアルファベット順に表示します。

この課題では、`std::map`とそれに関連する関数を調べて理解する必要があります。また、プログラムが想定どおりに動作するように、入力の検証とエラー処理を実装してください。