

なぜ自動翻訳をするのか

手動翻訳と比較して、完全で正しい自動的な翻訳はコードをすぐに移行できる。(長い年月を掛けること無く)また、翻訳の結果初めに定義した要件を十分に満たしている。(ただし、手動翻訳の方がクオリティーは高い)

しかし、自動翻訳は読みやすさを保証できるわけではない。(自然でないコードの一部が導入されるかもしれない)しかし、読みにくい所は手動翻訳で書き直す事が出来る。(全部手動翻訳しなくても、一部だけ手動翻訳することもできる)

手動java翻訳と比べて自動翻訳の質はどうか？

コードの結果による保守性や読みやすさを一番初めに注目する。自動翻訳はjavaコードの自然に見えるような生成を最適化することに多く適用されている。demo節を見てください

C言語のどのバージョンがサポートされているか？

バージョン使用ではなくコンパイラ仕様に合わせている。(バージョンではなくコンパイラに依存している)

この方法では元のcコードの機能を満たしている事を保証する事が出来る。(元のコードが明確なコンパイラの動きをした時)

私たちはGCC(C言語コンパイラ)の全ての新しいバージョンをサポートする事が出来る。また、他のコンパイラの翻訳も出来る。そしてc言語プログラミングスタイルの違いは関係ないです。

```
if(i==0){i=5;}
```

```
if(i==0)
    i=5;
```

```
if(i==0){
    i=5;
}
```

なまりとはプログラミングスタイルの違い

どのようにソースコードが翻訳されるのか？

Cソースコード翻訳の通常のビルド(ほぼコンパイル)をするくらい簡単だ。編集中にシステムは(Cコードやリンクなど)必要な全てのものを集める。

ビルドが成功した場合、集められたデータは、直接javaに翻訳される。

そのような翻訳はCソースコードの特定の構造と一致する。

もし複数の構造がある時は,(例:異なるプラットフォームを目標にするとき)一回、それぞれの構造を翻訳し,全ての翻訳結果を結合してjavaに移植する。

本当に動くの？

この翻訳理論の基礎はETFZurichによって開発された。他のページに詳しい説明があるよ。

