

# 卒業研究の方針の見直しについて

## 自然言語に基づく変数の命名方法の提案

70711041 鳥居克哉

# 研究の目的

命名方法を提案することで品質の良いコードを描けるようにする

↓

ソフトウェア保守のコスト削減に貢献する

# 従来の方針

## プログラムの書き方を自然言語に近づける

```
int result = plus(a).and(b);
```

こうすれば自然言語っぽくプログラムを読み流可能性がある

# 従来の方針の問題1

## プログラムの記述量が必要以上に増える

プログラミングに最適化された書き方ではないので必然的に記述量が増える

↓

研究の目的はソフトウェア保守のコスト削減

記述量が増えると保守コストが上昇してしまう

## 従来の方針の問題2

### プログラミング言語と自然言語の構文のギャップが大きい

構文のギャップを埋めようとするとうプログラミング言語の開発のような話になってしまう

↓

ソースコードの読み手はプログラマであり、ある程度のスキルを習得している

自然言語に近づけることが保守性を向上させるとは必ずしも言えない

# 一般的に命名について言われていること

より詳細・的確に対象を表現することが重視されている

- 明確な単語
- カラフルな単語
- 汎用的な名前を避ける
- 抽象的な名前よりも具体的な名前を使う
- 不要な単語の削除
- 名前の長さ
- 場合によって単位を含める

※参考

Trevor Foucher 「リーダブルコード」 オライリージャパン(2012)

# ソースコードと自然言語にの関係について言われていたこと

自然言語由来の記号を識別子の命名に用いる

↓

プログラマは自然言語由来の記号から意味するものを連想し、ソースコードを「読む」

# 識別子の命名について言えそうなこと

- 複数の単語の組み合わせによって命名される
- 識別子は以下のようになることがほとんど

**[メインとなる単語] + [メインを修飾する単語]\***

ex. `isStandardType` => `[is] + ([standard] + [type])`

↓

メインを修飾する単語にはなんらかの傾向が存在するのではないか

ex.

- 生存期間の短い変数なら修飾する単語が省略されるのではないか
- 数値を扱う変数なら単位(`sec`, `byte`...)が使われるのではないか
- メソッド名であれば「何が」に関する単語が抜けるのではないか  
...etc



# 新しい方針

識別子の種類と識別子に付けられる単語との関係を明らかにし、  
識別子名の命名手法を提案する

# 研究の進め方

1. オープンソースソフトウェアのコードを識別子を解析する
2. 解析結果から命名に関する相関を見つける
3. 相関で得られた結果を元に命名手法の提案をおこなう