研究の目的(はじめに)

1. 多くの人が関わるOSS＝可読性が高いという前提が存在する
2. 多くの人が集まる場所＝暗黙値が存在するとかんがえられる
3. 識別子の命名に関する暗黙値を、品詞に注目しつつ明らかにする
4. 明らかになった暗黙値から、識別子命名のガイドラインを提案する 目的は暗黙値のガイドラインか

書かなくちゃいけないこと(背景)

可読性とは

可読性とは、人間にとっての、ソースコードの内容の理解しやすさのことである。 可読性は以下の要素によって左右される

可読性が求められる理由

ソフトウェアにはいくつかの工程が存在する。 この工程全体をソフトウェアライフサイクルという。 ソフトウェアライフサイクルは、「企画」、「要件定義」、「開発」、「運用」、「保守」という工程に分けることができる。 そして、ソフトウェアの保守作業は、ソフトウェアライフサイクルのコストの大半を占める。 例えば、XXは、調査によって、ソフトウェアの保守作業がソフトウェアライフサイクルのコストのうちおよそ70%を閉めることを明らかにした。

ソフトウェアの保守において、保守作業のしやすさのことを保守性という。 ソフトウェアの保守コストを左右する要因として、ソフトウェアのソースコードの品質が考えられる。 ソースコードの品質にはさまざまな指標が存在する その指標の一つとして、可読性があげられる。 可読性は、プログラムの実行結果に影響を及ぼさない。 しかし、可読性がソフトウェアの品質に影響を与えることがある。 例えば、XXXは、ソースコードの可読性がソフトウェアの品質に影響を与えることを明らかにした。

また、保守作業の多くはソースコードの読解に使われる。 それゆえ、ソースコードの可読性の向上は、ソフトウェアの保守性を保つために重要であると考えられる。 可読性を向上させるには、さまざまな方法が存在する。 例えば、〇〇の研究ではソースコード中のコメントは可読性の向上に貢献することを明らかにした。

また、〇〇の研究では適切な改行によって、ソースコードの可読性が向上することを明らかにした。

さらに、Johannes, Janet, Danielは、識別子名に単語を使用した場合、文字を使用した場合に比べてソースコードの理解速度が有意に速くなることを明らかにした。

識別子と可読性の関係

識別子とは

識別子とは「プログラム言語における変数名や関数名を表す名前」である。

Javaのプログラムは「Token」、「Comment」、「WhiteSpace」という3つの最小単位によって構成される。 そして、Tokenは「Identifier」、「Keyword」、「Separator」、「Operator」の4つに分類することができる。

category	description
Identifier	識別子
Keyword	ASCII文字から成る50種類の予約後
Separator	区切り文字 ; , ... @ ::
Operator	ASCII文字から成る38種類の演算子

これより、javaにおける識別子はプログラムの最小単位で、変数やメソッドを表す名前であると言える。

javaの識別子の制限

javaにおいて、識別子の命名には、いくつかの構文的制限が存在する。 以下はjavaの識別子名の構文的制限である。

```
Identifier:
IdentifierChars but not a Keyword or BooleanLiteral or NullLiteral
IdentifierChars:
JavaLetter {JavaLetterOrDigit}
JavaLetter:
any Unicode character that is a "Java letter"
JavaLetterOrDigit:
any Unicode character that is a "Java letter-or-digit"
```

これより、Javaの識別子名には大文字小文字のアルファベットが使用可能であることがわかる。 識別子名に全てのアルファベットを使用できることから、プログラムの書き手は自由に識別子を命名することができると言える。

javaの識別子の単語の連結

識別子名は制限を満たせばあらゆる文字列を使用可能である。 識別子名にはしばしば単語や略語が用いられることがある。 また、複数の単語の組み合わせによる識別子の命名がされることもある。 javaにおいて、複数の単語を組み合わせる際は、以下に挙げる記法によって単語が組み合わされる。

form	description
UpperCamelCase	単語の先頭を大文字とし、単語を連結する
LowerCamelCase	先頭単語の先頭の文字を小文字とし、以降は単語の先頭を大文字として、単語を連結する
UPPER_SNAKE_CASE	全ての単語を大文字として、アンダースコアで単語を連結する
lower_snake_case	全ての単語を小文字として、アンダースコアで単語を連結する

Javaでは識別子が指す対象によって、推奨される記法が存在する。

identifier category	form
クラス名	UpperCamelCase

identifier category	form
型パラメータ名	一文字の大文字
メソッド名	lowerCamelCase
フィールド名(変数名)	lowerCamelCase
定数名（変数名)	UPPER_SNAKE_CASE
ローカル変数名(変数名)	小文字による短い単語

識別子の命名が可読性の向上に大きな影響を持っている

ソースコードは、その大半が識別子によって構成されている。 Florian, PizkaはEclipse(version 3.1.1)のソースコードを分析し、識別子トークンの数が全トークンに対して33%を占めており、 文字数では全文字数に対し識別子トークンの文字数が72%を占めていることを明らかにした。

これより、識別子はソースコードの品質に対し、大きな影響をもっていると考えられる。

命名規則と可読性

先に述べたように、プログラムの書き手は自由に識別子を命名する行うことができる。 自由に識別子を命名できることは一見いいことのように思われるが、その自由さ故に問題を引き起こすことがある。 〇〇は、識別子の命名は、プログラムの書き手によって恣意的に命名されることを指摘している。

恣意的な識別子の命名は、可読性の低下を招く。 それゆえ、ソフトウェア開発のプロジェクトによっては識別子の命名の方針を定めた、「命名規則」が導入されることがある。

命名規則と品詞

命名規則ではさまざまな項目が定められる。 しばしば、命名規則には品詞にかかわる規則が定められることがある。 javaでは、クラス名は名詞・名詞句、メソッド名は動詞・動詞句、変数名は名詞・名詞句になるよう識別子を命名することが推奨されている。

推奨される識別子の命名と品詞の対応

identifier	pos tag
class	noun
method	verbs
variable	noun

このように命名規則の中に品詞との関係をみることができる。

OSSと可読性

オープンソース・ソフトウェア(Open Source ・ Software)とは、誰もが検査、修正、拡張できるソースコードを持つソフトウェアのことである。

一般に、オープンソース・ソフトウェアの開発では、不特定多数のソフトウェア開発者が関わることになる。 これは、オープンソース・ソフトウェアのソースコードは不特定多数のソフトウェア開発者に共有されることを意味している。 不特定多数のソフトウェア開発者間でソースコードが共有されることから、オープンソース・ソフトウェアではソースコードの可読性が重要であると考えられる。 オープンソース・ソフトウェアの開発では、ソフトウェアの品質を維持するため、ソースコードの変更を統合する前に、変更内容を検証する「レビュー」という開発プロセスが存在する。 これよりオープンソース・ソフトウェアではレビューという開発プロセスによってソースコードの可読性が維持されていると考えられる。

自然言語と可読性

自然言語の構文が守られているほど可読性が高くなるという関係を指摘する もし関係を示すことができなければ、識別子内で自然言語の構文が乱れると可読性が低くなることを指摘する 自然言語に近いコード＝可読性が高いor識別子を自然言語の文法的に正しい＝可読性が高いを示す

AST解析

java環境で、javaオブジェクト表現としてjavaソースコードと対話することを可能にしたオブジェクト表現を、抽象構文木(AST: Abstract Syntax Tree)という。 ASTはソースコードを木構造として表現している。 この木構造にはファイル全体を表すルートを持っている。 このルートから木構造をたどることで、任意のノードにアクセスすることができる。

javaparserによるAST生成の例 ソースコード

```
package com.github.javaparser;

import java.time.LOcalDateTime;

public class TimePrinter {

  public static void main(String args[]){
    System.out.println(LocalDateTime.now());
  }
}
```

javaparserによってソースコードから生成されたAST(一部省略)

```
salt
{
  {T
  + CompilationUnit
  ++ PackageDeclaration
  +++ QualifiedNameExpr
  ++++"com.github.javaparser"
  ++ SingleTypeImportDeclaration
  +++ ClassOrInterfaceType
  ++++"java.time.LocalDateTime"
  ++ ClassOrInterfaceDeclaration
  +++ NameExpr
```

```
    ++++ "TimePrinter"
    +++ MethodDeclaration
    ++++ "void main String args"
    }
  }
```

研究同期

以上を踏まえてまとめあげる こういう理由で、識別子の研究をし他みたいなことをいう

識別子の収集

調査を行うにあたって識別子を収集した。 識別子収集の対象としたのはGithub上に存在する177のjavaプロジェクトである。

初めに、Github APIを用いてリポジトリ名一覧を取得した。 そして、各プロジェクトに対してJavaparserによるAST解析を行うことで、クラス名、メソッド名、変数名を抽出した。 ここでは、ローカル変数とフィールド変数を合わせて変数名としている。 抽出した際のフォーマットは以下の通りである。

identifier	format
クラス名	Githubユーザ名/Githubリポジトリ名, ファイル名, クラス名, public修飾子の有無
メソッド名	Githubユーザ名/Githubリポジトリ名, ファイル名, メソッド名, public修飾子の有無, メソッドの戻り値の型
変数名	Githubユーザ名/Githubリポジトリ名, ファイル名, 変数名, public修飾子の有無, 変数の型, フィールド変数かどうか

識別子の品詞タグ付け

本研究では識別子中の品詞に焦点を当てている。 それゆえ、識別子名を分ち書きをし、各単語の品詞を取得する必要がある。 そこで本研究では、javaの識別子中の単語の連結がキャメルケースによって行われていることに着目し、以下の正規表現によって単語を分ち書きを行った。

```
([a-z]+)([A-Z][a-z]+)|([A-Z][a-z]+)
```

正規表現による分ち書きの例

identifier	result
video	video
Video	Video
videoGame	video, Game
VideoGame	Video, Game
VideoGamePlayer']	Video, Game, Player

分ち書きを行った後、各単語の品詞の取得を行った。 品詞の取得にはpythonのNatural Language Toolkit(NLTK)というパッケージを用いた。 このパッケージは、Pythonで自然言語を処理する際ためのさまざまな機能が用意されたものである。

NLTKで品詞を取得する際はタグセットを指定できるが、今回はデフォルトで指定されている「PennTreebank tagset」タグセットを用いた。

PennTreebank tagsetの品詞対応表

tag	description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol

tag	description
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

各品詞を取得し、

調査結果分析

クラス、メソッド、変数に分けて調査を行う

- 単語の場所と品詞の出現度の傾向
- クラス、メソッド、変数による品詞の出現頻度の傾向
- マルコフ連鎖による自然言語との構文的な比較

構文解析

自然言語の品詞同士の関係について 自然言語で品詞がどのように関係しているのかを紹介する

識別子中の品詞動詞の関係について調査する

議論

可読性と自然原語の関係を述べる 自然言語＝可読性が高いが言えないとまずい

結論

自然言語宙の品詞の何らかの関係と識別子中の品詞の何らかの関係から、有意なデータを示す

調査対象としたリポジトリー覧

- Azure/azure-sdk-for-java
- aws/aws-sdk-java
- aliyun/aliyun-openapi-java-sdk

- bytedeco/javacpp-presets
- quarkusio/quarkus
- bcgit/bc-java
- google/guava
- SonarSource/sonar-java
- radcortez/javaee7-angular
- alibaba/fastjson
- bjmashibing/java
- kubernetes-client/java
- ramram43210/Java
- apache/servicecomb-java-chassis
- libgdx/libgdx
- tensorflow/java
- querydsl/querydsl
- grpc/grpc-java
- java-native-access/jna
- mongodb/mongo-java-driver
- javaparser/javaparser
- tronprotocol/java-tron
- skylot/jadx
- watson-developer-cloud/java-sdk
- iluwatar/java-design-patterns
- mybatis/mybatis-3
- shopizer-ecommerce/shopizer
- fabric8io/kubernetes-client
- javaee-samples/javaee7-samples
- jboss-javassist/javassist
- GoogleCloudPlatform/java-docs-samples
- graphql-java/graphql-java
- docker-java/docker-java
- realm/realm-java
- DuGuQiuBai/Java
- gitblit/gitblit
- aliyun/aliyun-oss-java-sdk
- brianway/java-learning
- GoogleContainerTools/jib
- thaycacac/java
- square/retrofit
- zeromq/jeromq
- alibaba/arthas
- msgpack/msgpack-java
- agoncal/agoncal-book-javaee7
- mthli/Java
- OpenFeign/feign
- dcm4che/dcm4che
- javamelody/javamelody

- testcontainers/testcontainers-java
- jfinal/jfinal
- zxing/zxing
- google/gson
- searchbox-io/Jest
- rabbitmq/rabbitmq-java-client
- dromara/soul
- ScottOaks/JavaPerformanceTuning
- scribejava/scribejava
- pubnub/java
- json-iterator/java
- EleTeam/Shop-for-JavaWeb
- phishman3579/java-algorithms-implementation
- appium/java-client
- exercism/java
- DreamCats/JavaBooks
- eclipse/paho.mqtt.java
- java-decompiler/jd-gui
- hub4j/github-api
- JustinSDK/JavaSE6Tutorial
- chenhaoxiang/Java
- googleapis/google-cloud-java
- TheAlgorithms/Java
- Vedenin/useful-java-links
- chanjarster/weixin-java-tools
- byhieg/JavaTutorial
- code4craft/webmagic
- qos-ch/slf4j
- json-path/JsonPath
- apereo/java-cas-client
- RichardWarburton/java-8-lambdas-exercises
- redis/jedis
- in28minutes/JavaInterviewQuestionsAndAnswers
- DiUS/java-faker
- lenve/JavaEETest
- hamcrest/JavaHamcrest
- mrdear/JavaWEB
- structurizr/java
- googlemaps/google-maps-services-java
- TooTallNate/Java-WebSocket
- JeffLi1993/java-core-learning-example
- awangdev/LintCode
- ronmamo/reflections
- java8/Java8InAction
- crossbario/autobahn-java
- prometheus/client_java

- objectbox/objectbox-java
- bytedeco/javacv
- rybalkinsd/atom
- influxdata/influxdb-java
- tobato/FastDFS_Client
- moquette-io/moquette
- oldmanpushcart/greys-anatomy
- echoTheLiar/JavaCodeAcc
- bytedeco/javacpp
- biezhi/learn-java8
- 0opslab/opslabJutil
- egzosn/pay-java-parent
- qiyuangong/leetcode
- ChinaLHR/JavaQuarkBBS
- crossoverJie/JCSprout
- jwtk/jjwt
- natural/java2python
- micromasterandroid/java
- HelloWorld521/Java
- mrniko/netty-socketio
- NanoHttpd/nanohttpd
- google/google-java-format
- shyiko/mysql-binlog-connector-java
- qiyaTech/javaCrawling
- javaee-samples/javaee8-samples
- android-async-http/android-async-http
- onblog/JavaMonitor
- alibaba/p3c
- notnoop/java-apns
- tipsy/javalin
- bottleleung/Java
- winterbe/java8-tutorial
- square/javapoet
- akshitagit/JAVA
- AllAlgorithms/java
- nanchen2251/RxJava2Examples
- yasserg/crawler4j
- auth0/java-jwt
- Blankj/awesome-java-leetcode
- otale/tale
- stleary/JSON-java
- amitshekhariitbhu/RxJava2-Android-Samples
- hmkcode/Java
- kaushikgopal/RxJava-Android-Samples
- AllenDowney/ThinkJavaCode
- happyfish100/fastdfs-client-java

- csxiaoyaojianxian/JavaScriptStudy
- socketio/socket.io-client-java
- Beerkay/JavaMultiThreading
- XU-ZHOU/Java
- shekhargulati/java8-the-missing-tutorial
- gaopu/Java
- WritingMinds/ffmpeg-android-java
- binarywang/weixin-java-mp-demo
- yangfuhai/ASimpleCache
- ipinfo/java
- vdurmont/emoji-java
- KikiLetGo/VirusBroadcast
- joeyajames/Java
- kevinsawicki/http-request
- javagrowing/JGrowing
- VerbalExpressions/JavaVerbalExpressions
- biezhi/30-seconds-of-java8
- in28minutes/JavaWebApplicationStepByStep
- enhorse/java-interview
- shakeelstha/java
- SquareSquash/java
- sendbird/SendBird-JavaScript
- Snailclimb/JavaGuide
- Nirman-Rathod/Java
- CarpenterLee/JavaLambdaInternals
- shekhargulati/strman-java
- nramnad/java
- jenkins-docs/simple-java-maven-app
- MindorksOpenSource/from-java-to-kotlin
- MicrosoftDocs/pipelines-java
- h2pl/Java-Tutorial
- SuperMap/iClient-JavaScript
- 17mon/java
- mercyblitz/jsr
- udacity/Just-Java