第11章

この章の目的

◆コメントのしかたについて学ぶ

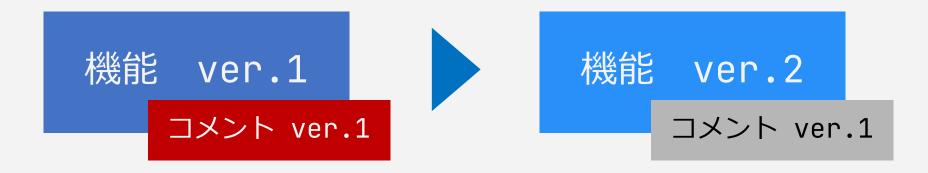
内容

内容とキーワード

- ◆退化コメント
- ◆コメントで命名をごまかす
- ◆コメントはいつ読まれるのか
- ◆ドキュメントコメント

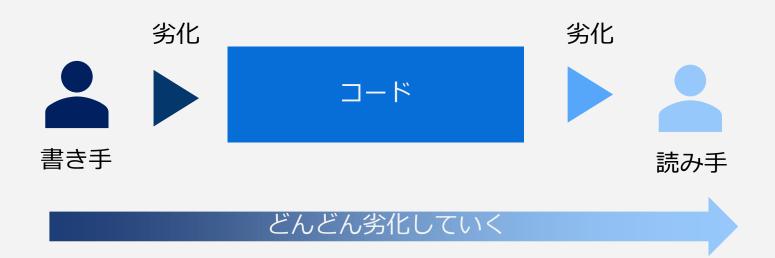
情報が古くなってしまったコメントを退化コメントと呼ぶ。

- ◆コメントはコードと比べてメンテナンスされにくい
 - ■機能が追加された際にコメントを更新し忘れる



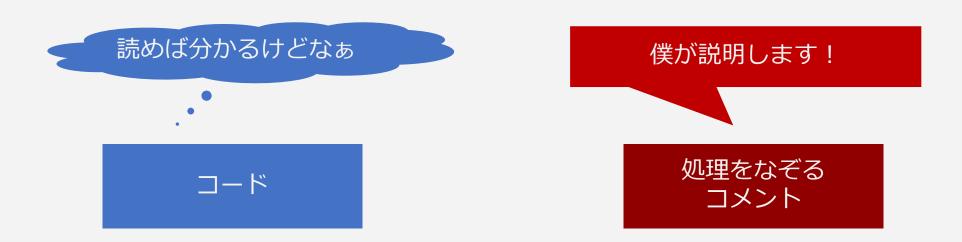
コメントが退化してしまっている Ver.2についての記載が無い

- ◆コメントは劣化コピーにすぎない
 - ■伝えたい内容は書き手の意思の劣化コピーにすぎない



可能な限り意思が伝わるコメントを心がける

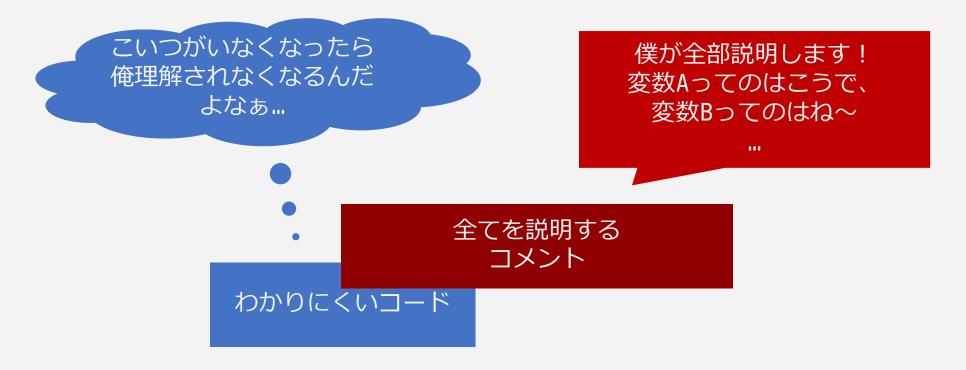
- ◆ロジックの挙動をなぞるだけのコメントは避ける
 - ■コードを更新するたびに更新する必要が出てくる
 - ■また、コードの理解にさほど貢献しない
 - ■伝言ゲームの性質を帯びてくる



コメントで命名をごまかす

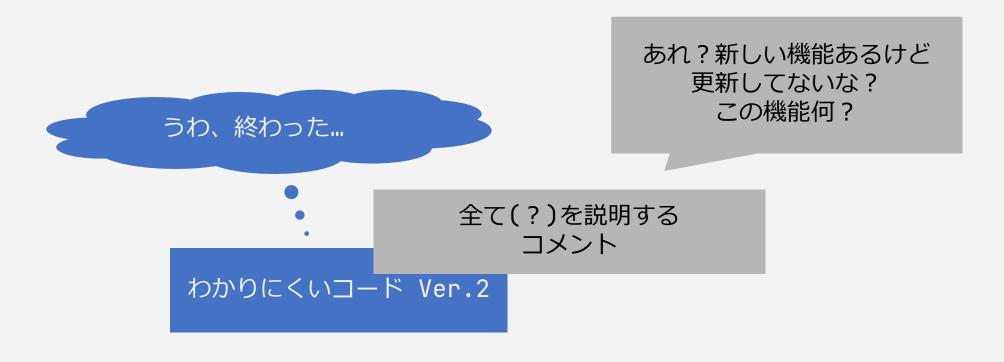
コメントで命名をごまかす

- ◆わかりにくい命名をコメントでカバーすると...
 - ■コメントありきのコードになってしまう



コメントで命名をごまかす

◆再説明をすると、コメントが退化しやすくなる



そもそもわかりやすい命名にすればこうはならない

コメントはいつ読まれるのか

コメントはいつ読まれるのか

- ◆機会としては保守と仕様変更時が多い
- ◆以下の二つを意識してコメントを書くようにする
 - ■コード保守時は
 - このロジックはどういう意図で動いているのかを気にする
 - 例:○○の状態の時にTrueを返す
 - ■仕様変更時は
 - 何に注意すれば安全に変更できるかをきにする
 - 例:○○の状態の仕様を変更する際はここのロジックを変更する

ドキュメントコメント

ドキュメントコメント

- ◆エディタ上で、メソッドなどにカーソルを合わせる と、情報が出てくることはありませんか?
- ◆あれはドキュメントコメントで作成されています

◆皆さんも独自のドキュメントコメントをかくことが

できます

```
net.URLConnection;

    java.net.URL

security.NoSuchAlgorithmException
                                        public java.net.URLConnection openConnection()
Main {
                                        object referred to by the URL.
tatic void main(String[] args) th
HttpsServer().start();
                                        URLStreamHandler.openConnection(URL) method of the protocol handler for this
ad Client = new HttpsClient();
nt.start():
                                        network connection on creation. This will happen only when calling URLConnection.
lient extends Thread {
 id run() {
                                        rii < corretto-11 >
URL url = new URL( spec: "https://w
URLConnection connection = url.openConnection()
```

次の章に向けて

◆コメントの適切な書き方を学んだ

◆次章ではメソッドの設計について学ぶ