

# 可読性の良い プログラミングのススメ

 ぽぴ王子 @ わんくま同盟

## 今回話すこと

1. 読みやすいコードの書き方
2. ぽぴ王子がコーディングをする上で気をつけていること。
3. いわゆるオレ流プログラミングスタイル。  
(あくまで“オレ流”なのであまり参考にはなりません)

# Fool Proof を考えよう！

## e-Words より【フルプルーフとは】

工業製品や生産設備、ソフトウェアなどで、利用者が誤った操作をしても危険に晒されることがないように、設計の段階で安全対策を施しておくこと。正しい向きにしか入らない電池ボックス、ドアを閉めなければ加熱できない電子レンジ、ギアがパーキングに入っていないとエンジンが始動しない自動車、などがフルプルーフな設計の例である。

- 現場にはいろいろなレベルの人間が集まる
- トリッキーなコードは不慣れな人間には理解されにくい・または誤解を招きやすい
- あまちゃっこ(八戸弁)にも読めるコードを！

## コメントを書こう！

- コメントを書こう

なぜこういうソースになったのか、をわかるようにしよう

ソースの意図を伝えるためのコメント

- 間違ったコメントは逆効果

ソースを変更したら一緒にコメントも変更しよう

「索引のない本は残念だが、もっと残念なのは間違った索引がある本だ」

- “適切な”場所に“適切な”コメントを

コメントを書こう！

## 間違ったコメントの例

```
// 値が10以上の場合  
if (val > 10)  
{  
}
```

否定の“!”は使わない方が見やすい

## R・田中一郎さんを判断する処理

```
if (素敵だし爽やか)
{
    MessageBox.Show("あなたは、R・田中一郎さんですね (´▽`)9 ビシッ!");
}
```

## 処理を逆にしなければいけなくなったとき

```
if (!素敵だし爽やか)
{
    MessageBox.Show("あなたは、R・田中一郎さんじゃないですね");
}
```

違いはここ!

否定の“!”は使わない方が見やすい

“否定”→“trueではない”とするとわかりやすい

```
if (素敵だし爽やか != true)
{
    MessageBox.Show("あなたは、R・田中一郎さんじゃないですね");
}
```

でも最初からfalseと比較した方がもっとわかりやすい

```
if (素敵だし爽やか == false)
{
    MessageBox.Show("あなたは、R・田中一郎さんじゃないですね");
}
```

特に理由がなければ != より == を使った方がいい

## ブロックにはカッコをつけよう

```
if (ねんがんのアイスソードをてにいれた)
    MessageBox.Show("ころしてもうばいとる!");
else
    MessageBox.Show("ゆずってくれ たのむ!!");
```

## メッセージを追加してみると...

```
if (ねんがんのアイスソードをてにいれた)
    MessageBox.Show("ころしてもうばいとる!");
    MessageBox.Show("なにをきるきさまらー!");
else
    MessageBox.Show("ゆずってくれ たのむ!!");
```

ここがエラーになる ←

## ブロックにはカッコをつけよう

```
if (ねんがんのアイスソードをてにいれた)
```

```
{ ← ココから
```

```
    MessageBox.Show("ころしてもうばいとる!");
```

```
    MessageBox.Show("なにをやるきさまらー!");
```

```
} ← ココまでを実行するYO!
```

```
else
```

```
{
```

```
    MessageBox.Show("ゆずってくれ たのむ!!");
```

```
}
```

カッコで囲むことで、処理をしている場所を特定できるようにする！

## 最適化は人間に向けて行おう

- 人間が読みにくくなる最適化は“難読化”である
- イマドキのコンパイラさんは偉いので、多少コンピュータにわかりにくくても、頑張って最適化してくれる(はず)
- だから最適化はコンピュータよりも人間が読みやすい方で考えるべき
- (コンピュータへの) **小さな親切** (人間への) **大きなお世話**

最適化は人間に向けて行おう

最適化はまず人間のために！

次にコンピュータのために！

by ぽび王子

## 最適化は人間に向けて行おう

➤ あなたは友達以上

なおかつ

➤ あなたは恋人未満

という条件式があったとして、普通にコーディングすると

```
if (あなた >= 友達 && あなた < 恋人)
{
    MessageBox.Show("ただのキープ君じゃねーか!!");
}
```

こんな感じ

## 最適化は人間に向けて行おう

これを人間向けに最適化してみる

➤ あなたは友達～恋人の間である

という条件に変更した場合は以下のコードになる

```
if (友達 <= あなた && あなた < 恋人)
{
    MessageBox.Show("こう表現すると初々しい感じがしねえ?");
}
```

あえてあなた(自分)を中心に持ってくることで

➤ 友達 ≤ あなた < 恋人

と同じイメージを if に持たせることができる

# わざとポイントを強調してみる

## たとえばこんなソースコード

```
// お笑い同盟以外のメンバーを勧誘する
if (わんくまメンバー != ぽぴ王子 &&
    わんくまメンバー != R・田中一郎 &&
    わんくまメンバー != アクア)
{
    MessageBox.Show("わんくまお笑い同盟に入らないか?");
}
```

## わざとポイントを強調してみる

こんな風書き換えてみる。

```
// お笑い同盟以外のメンバーを勧誘する
switch (わんくまメンバー)
{
    case ぽぴ王子:
    case R・田中一郎:
    case アクア:
        /* 何もしない */

    default:
        MessageBox.Show("わんくまお笑い同盟に入らないか?");
}
```

あえてムダなところを入れておこう

普通のコメントだったら

// 鳩が出ますよー

で、かまわないのだけど...

大事なところは

/\* 中さんが出ますよー \*/

こんな風に普段使わない /\* ... \*/ を使うことであえて強調してみたりする

## まとめ

- 的確なコメントをたくさん書こう
- 素直なソースコードを書こう
- ポイントになるところは目立たせよう
- R・田中一郎さんは素敵だし爽やかなにをするきさらー！！

Questions?

