Prosta instrukcja if czy switch. No przecież tu nie da się zrobić niczego lepiej. No, niekoniecznie. Da radę i zaraz to pokażę.

Przykłady będą w PHP, ale odnosi się do większości języków.

Instrukcja if

Myślę, że nie potrzebuje zbyt długo tłumaczyć jak działa if. Ale chciałbym pokazać, co można zrobić, aby był czytelniejszy. Pamiętaj - kod pisze się raz, a czyta wiele razy. Warto zatem odpowiednio go napisać.

{codecitation class='brush: php'}

```
// jest to juz trzecia proba polaczenia  // i juz bylo blednie podane haslo   if ($a == 3 && $b > 0) { $file = fopen('file', 'w+'); flock($file, LOCK_EX); $n = fwrite($file, $b); flock($file, LOCK_UN); $a += n; }
```

{/codecitation}

No i co tu jest nie tak? Przeciez komentarz ladnie opisuje wszystko, mamy wciecia, mamy wszystko czego potrzebujemy... No właśnie, ale czy nie da się lepiej?

Krok 1 - wyrzuć magiczne liczby

{codecitation class='brush: php'}

```
// jest to juz trzecia proba polaczenia  // i juz bylo blednie podane haslo   if ($connection_attemps == MAX_CONNECTION_ATTEMPTS && $wrong_passwords > 0)  { ... }
```

{/codecitation}

Prawda, że już czytelniej. W sumie można teraz usunąć komentarz. Nie podobają mi się jeszcze 2 rzeczy.

Krok 2 - kolejność porównań i jednoznaczność

Ile razy zdarzył ci się taki błąd:

{codecitation class='brush: php'}if ($tmp = true) {...} {/codecitation}

i przez godzinę szukałeś błędu. To jest bezsensowne, bo jeśli do `$tmp' przypiszesz 3 to warunek będzie spełniony. Zawsze. Dlatego warto zamienić kolejność: najpierw stała, potem zmienna. Tak jak tu:

{codecitation class='brush: php'}

if (MAX_CONNECTION_ATTEMPTS === $connection_attemps && $wrong_passwords > 0) {
... }

{/codecitation}

Dodatkowo w językach skryptowych często występują dwa operatory porównania: "jest równe" (==) i "jest identyczne" (===). Jeśli coś piszesz, to powinieneś wiedzieć, jakiego typu coś będzie i jaka jest oczekiwana wartość. W takich sytuacjach powinieneś porównywać właśnie z taką wartością i wykorzystywać operator "jest identyczny".

Krok 3 - Osobna funkcja (metoda)

Bardzo często się zdarza, że taki warunek staje się niejako "stanem". I występuje w kilku miejscach. Jest niewiele rzeczy gorszych od powtarzającego się kodu. Z takim kodem najpiej jest... wrzucić go do funkcji (metody) i wywoływać.

O ile w obiekcie ma to większe uzasadnienie (szczególnie, jeśli warunek opiera się na właściwościach obiektu), o tyle w kodzie strukturalnym jest to czasem utrudnienie - jeśli do funkcji przekazujesz 5 parametrów, to przestaje to być czytelne.

{codecitation class='brush: php'}

```
define('MAX_CONNECTION_ATTEMPTS', 3);    function maxConnectionAttemptsAndWrongPass($connection_attemps, $wrong_passwords) {      return MAX_CONNECTION_ATTEMPTS === $connection_attemps && $wrong_passwords > 0;  } function logger($content) {      $file = fopen('file', 'w+');      flock($file, LOCK_EX);      $n = fwrite($file, $content);      flock($file, LOCK_UN);      return n;  }    if (maxConnectionAttemptsAndWrongPass($connection_attemps, $wrong_passwords)) { logger($wrong_passwords);  }
```

{/codecitation}

Tu trochę na siłę jest wprowadzona jest funkcja `maxConnectionAttemptsAndWrongPass()'. Ale chciałem pokazać zasadę.