Czysty kod

Dlaczego?

Czym jest czysty kod?

Czym jest czysty kod?

Czysty kod jest prosty i bezpośredni. Czysty kod czyta się jak dobrze napisaną prozę. Czysty kod nigdy nie zaciemnia zamiarów projektanta.

Grady Booch

Czym jest czysty kod?

Czysty kod jest prosty i bezpośredni. Czysty kod czyta się jak dobrze napisaną prozę. Czysty kod nigdy nie zaciemnia zamiarów projektanta.

Grady Booch

Kod elegancki i efektywny. Logika kodu powinna być prosta, aby nie mogły się w niej kryć błędy, [...] a wydajność zbliżona do optymalnej, aby nikogo nie kusiło psucie kodu w celu wprowadzenia niepotrzebnych optymalizacji. Czysty kod wykonuje dobrze jedną operację.

Bjarne Stroustrup

MI

Nazwy

Używaj nazw przedstawiających intencje

```
d <- 10 # Czas trwania w dniach

elapsed_time_in_days <- 10
days_since_creation <- 10
days_since_modification <- 10
file_age_in_days <- 10
```

Używaj nazw przedstawiających intencje

```
get_them <- function(the_list){
  list1 <- list()
  for (i in 1:length(the_list))
    if (the_list[[i]][1] == 4)
      list1 <- append(list1, the_list[i])
  return(list1)
}
```

Używaj nazw przedstawiających intencje

```
get_them <- function(the_list){
  list1 <- list()
  for (i in 1:length(the_list))
    if (the_list[[i]][1] == 4)
      list1 <- append(list1, the_list[i])
  return(list1)
}

get_flagged_cells <- function(game_board){
  flagged_cells <- list()
  flagged <- 4
  for (cell in 1:length(game_board))
    if (game_board[[cell]]['status_value'] == flagged)
      flagged_cells <- append(flagged_cells, game_board[cell])
  return(flagged_cells)
}
```

Unikaj dezinformacji

```
account_list

account_group
bunch_of_accounts
accounts

xyz_controller_for_efficient_handing_of_strings
xyz_controller_for_efficient_storage_of_strings

for (i in 1:n)
  if (0 == 1)
    a = 1
```

Funkcje

```
function(){
  dodaj(maslo)
  while (nie_jest_stopione(maslo)) {
    for (atom in atomy_w_probce_masla) {
      dostarcz_energii(atom)
    }
  }

  while (jest_cala(szynka)) {
    oddzialuj_nozem_na_siec_krystaliczna_szynki()
  }
  dodaj(szynka)


  while (jest_caly(szczypiorek)) {
    oddzialuj_nozem_na_siec_krystaliczna_szczypiorku()
  }
  dodaj(szczypiorek)

  dodaj_enerigii_potencjalnej(noz)
  zamien_energie_potencjalna_na_kinetyczna(noz)
  uderz_w(jajko, noz)
  dodaj(zawartosc_jajka)

  for (krysztalek_soli in szczypta_soli) {
    dodaj(krysztalek_soli)
  }

  for (ziarnko_pieprzu in szczypta_pipeprzu) {
    dodaj(ziarnko_pieprzu)
  }

  while (!jajka_sciete()) {
    mieszaj()
  }
}
```

http://pawelwlodarski.blogspot.com/2011/02/poziomy-abstrakcji.html

Małe funkcje

```
zrob_jajecznice<- function(){
  roztop(maslo)
  dodaj(szynka, "krojona")
  dodaj(szczypiorek, "krojony")
  dodaj(jajko, "rozbite")
  dodaj(sol)
  dodaj(pieprz)
  mieszaj_do_momentu_sciecia_sie_jajek()
}
```

Wykonuj jedną czynność

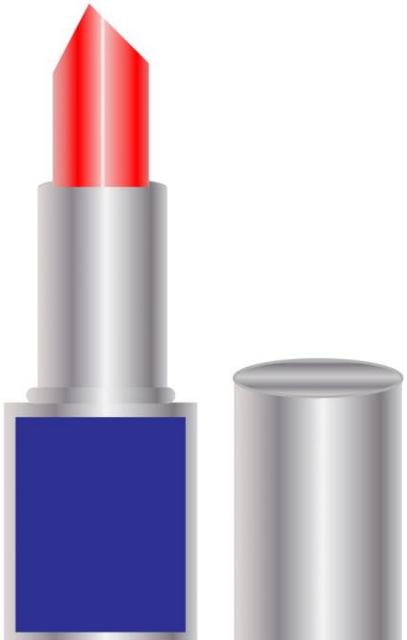
Funkcje powinny wykonywać jedną operację.
Powinny robić to dobrze. Powinny robić tylko to.

Zasada Zstępująca:
Kod powinien być tak skonstruowany, aby można go było czytać jako serię akapitów ABY.

Jak pisać takie funkcje?

Komentarze

Komentarze nie są szminką dla złego kodu



Nie komentuj złego kodu - popraw go.

Brian W. Kernighan i P. J. Plaugher

MI

```
# Check if employee is egible for full benefits
if ((employee$flags & hourly_flag) & (employee$age > 65))


if (is_egible_for_full_benefits(employee))
```

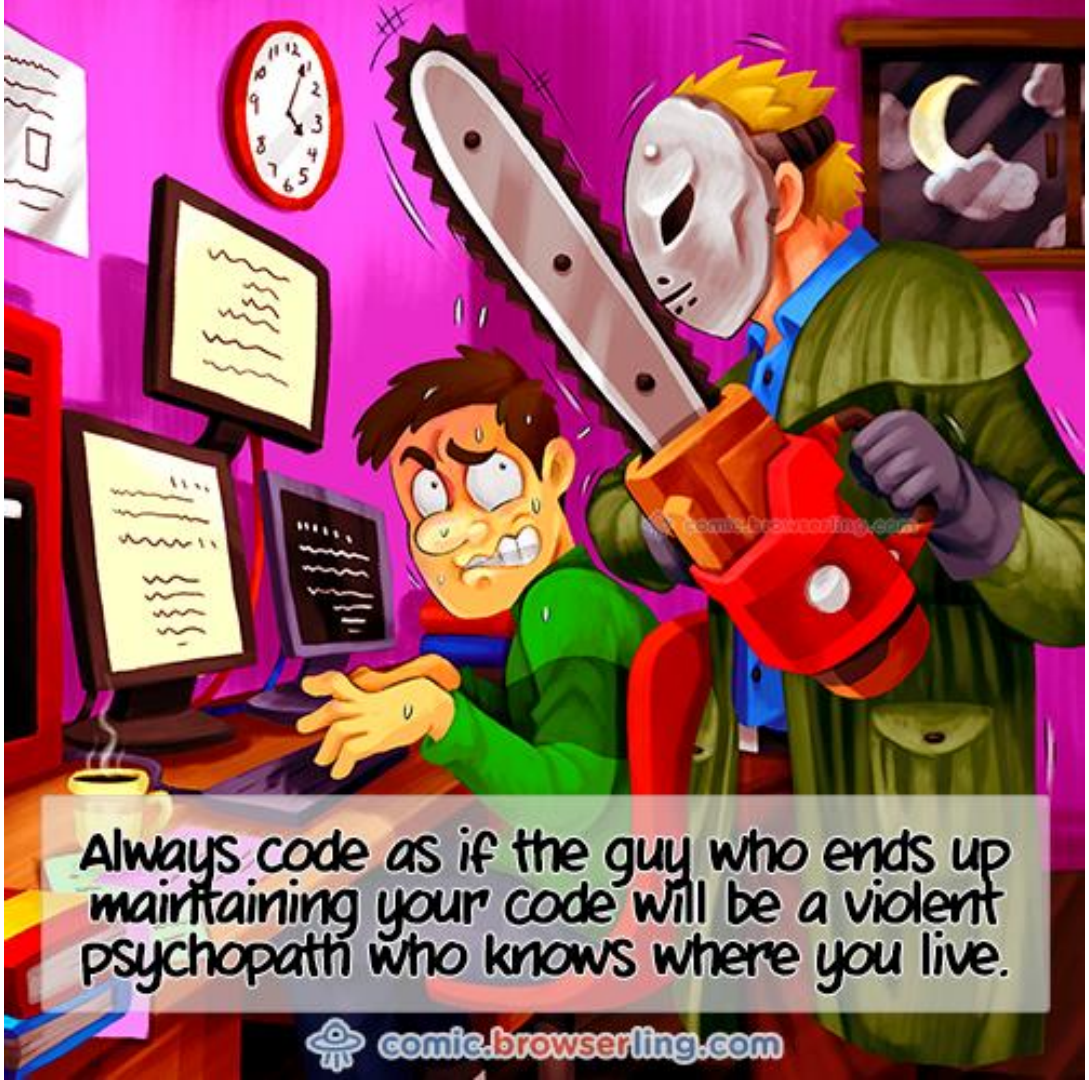
```
; UGH!!! Horrible bug here that should be fixed at some point:
; If the name we are scanning is longer than CX, we keep on reading!


                                        CALL    RESTXADDR       ; If you change this into a jmp don't come
                                        return                  ; crying to me when things don't work ARR

 ;Profiler for MS-DOS 1.25 2.00
 ;
 ; Lots of stuff stolen from debug.
                                    egomes: db      "Chris Peters helped with the new dos!",cr,lf
                                            db      "Microsoft rules ok$"

 ; Outputs:
 ;       DS:BX = Points to FAT ID byte (IBM only)
 ;               GOD help anyone who tries to do ANYTHING except
 ;               READ this ONE byte.
```

Podsumowanie

Nazwy powinny przedstawiać intencje,

Funkcje powinny być krótkie oraz wykonywać jedną czynność, instrukcje w funkcji powinny być na tym samym poziomie abstrakcji

Komentarze **nie** są szminką dla złego kodu!


Always code as if the guy who ends up mairtaining your code will be a violent psychopath who knows where you live.
comic.browserling.com