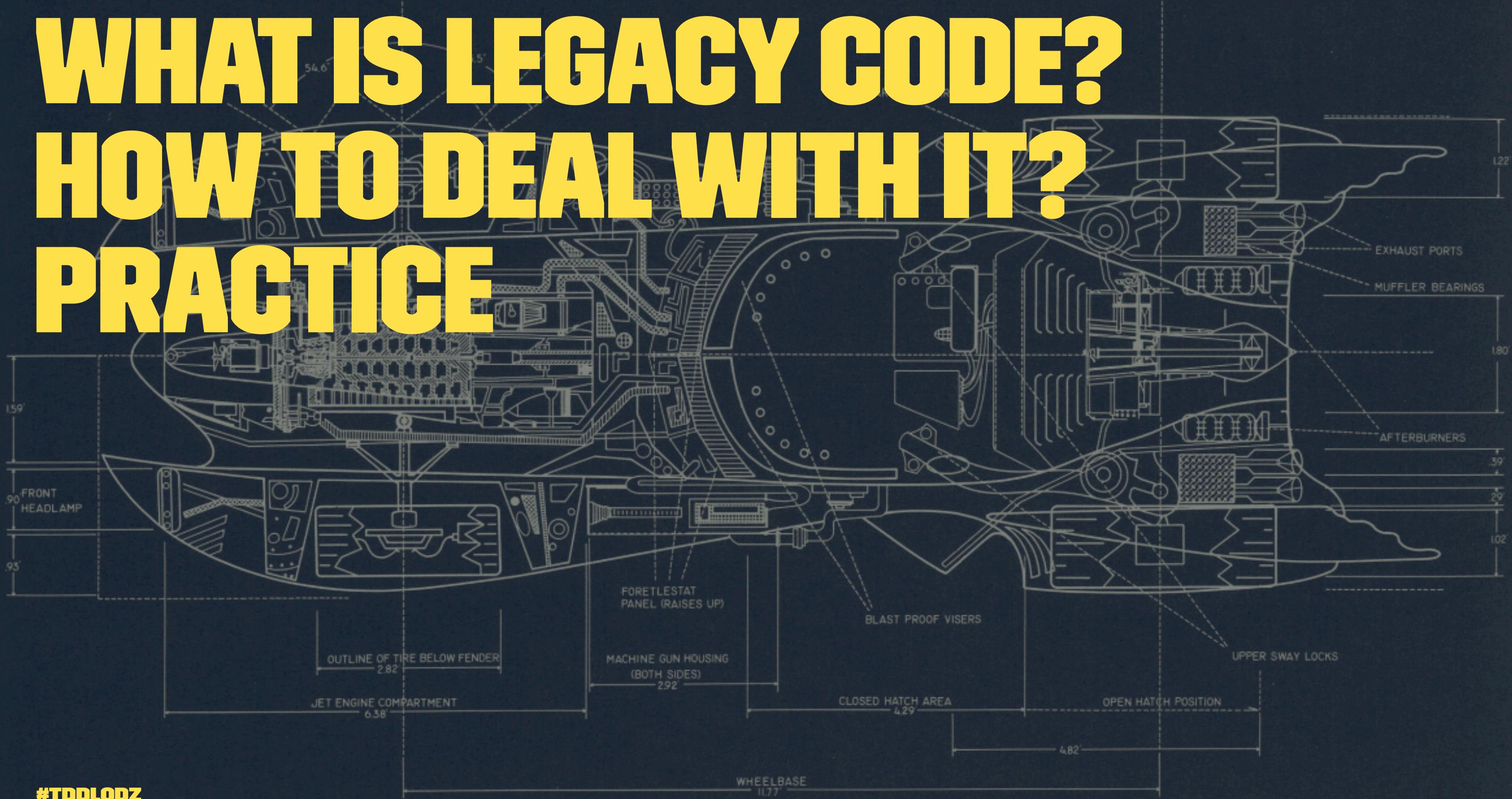


# WORKING WITH LEGACY CODE

@MACIEJOCZKO

# WHAT IS LEGACY CODE? HOW TO DEAL WITH IT? PRACTICE



WHAT?

**“CODE YOU DIDN’T  
WRITE THIS MORNING  
(YET)”**

**“CODE IS LEGACY CODE  
AS SOON AS IT'S  
WRITTEN”**

Michael Feathers

# LEGACY CODE: CODE WITHOUT TESTS

# LEGACY CODE

- » inherited
- » poorly designed
- » too complicated
- » illegible

# HANDS ON #1

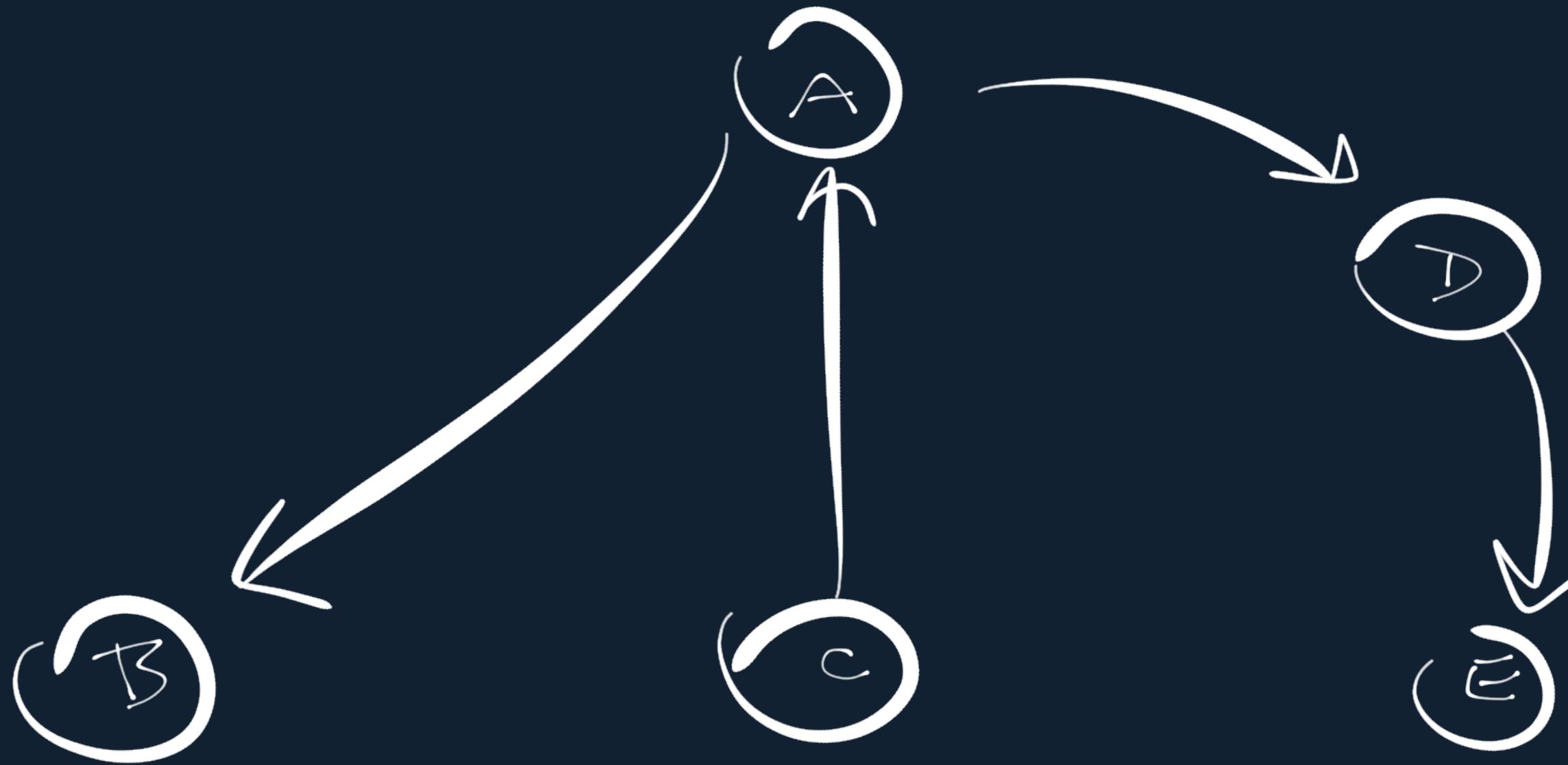
1. origin/master
2. Look at PollViewController viewWillAppear: method.
3. Test, if rightBarButtonItem is set correctly depending on PollManager isPollCompleted property value.

POW?

# APPROACH

- » Identify change points
- » Find an inflection point
- » Cover the inflection point
- » Make changes
- » Refactor the covered code

# INFLECTION POINT



# SINGLE RESPONSIBILITY PRINCIPLE

# OPEN-CLOSED PRINCIPLE

# COVERING INFLECTION POINT

- » Break external dependencies
- » Break internal dependencies
- » Write tests

**BREAKING  
EXTERNAL  
DEPENDENCIES IS  
ABOUT MOVING...**

# FROM THIS

```
- (NSUInteger)calculatePrice {  
    // ...  
    StockAnalyzer *analyzer = [StockAnalyzer new];  
    // ...  
}
```

# TO THIS

```
- (instancetype)initWithAnalyzer:(StockAnalyzer *)analyzer {
    self = [super init];
    if (self) {
        self.analyzer = analyzer;
    }
    return self;
}

- (NSUInteger)calculatePrice {
    // ...
    id analysis = [self.analyzer analyze];
    // ...
}
```

**BREAKING  
INTERNAL  
DEPENDENCIES IS  
ABOUT MOVING...**

# FROM THIS

```
- (CGSize)calculateSize {  
    // ...  
    UIScreen *screen = [UIScreen mainScreen];  
    // ...  
}
```

# TO THIS

```
- (CGSize)calculateSize {  
    // ...  
    UIScreen *screen = [self screen];  
    // ...  
}  
  
- (UIScreen *)screen {  
    return [UIScreen mainScreen];  
}
```

# WRITE TESTS

**MAKE**

changes

# HANDS ON #2

1. origin/poll\_singleton
2. Look at PollViewController text field delegate methods.
3. Extract text validation logic, test it and test whether view controller validates text fields' inputs correctly.

# REFACTOR TIPS



WATCH  
YOUR  
STEP

# TOO WIDE CLASS RESPONSIBILITY

```
@interface MyClass : NSObject  
@property(nonatomic, readonly) MessagesComposer *composer;  
@property(nonatomic, readonly) AlbumReader *reader;  
@property(nonatomic, readonly) FeedParser *parser;  
// ...  
@end
```

# DUPPLICATION



# NOT READABLE CODE

```

char*_ = "/*";
#include <stdio.h>
#define m 21
#define o(l, k) for(l=0; l<k; l++)
#define n(k) o(T, k)

int E,L,O,R,G[42][m],h[2][42][m],g[3][8],c
[42][42][2],f[42]; char d[42]; void v( int
b,int a,int j){ printf("\33[%d;%df\33[4%d"
"m ",a,b,j); } void u(){ int T,e; n(42)o(
e,m)if(h[0][T][e]-h[1][T][e]) { v(e+4+e,T+2
,h[0][T][e]+1?h[0][T][e]:0); h[1][T][e]=h[
0][T][e]; } fflush(stdout); } void q(int l
,int k,int p){
    int T,e,a; L=0
    ; O=1; while(O
    ){ n(4&&L){ e=
    k+c[1] [T][0];
    h[0][L-1+c[1][
    T][1]] [p?20-e:
    e]=-1; } n(4){
    1]+1; if(a==42
    ; } } n(4){ e=
    T][1]][p?20-e:
    u(); } n(42) {
    o(a, m&&e==m){
    ]=h[0][L-1] [a
    }int main(){ int T,e,t,r,i,s
    1][T]=7-T; R--; n(42) o(e,m)
    R; n(17){ e=d[T]-48; d[T]=0;
    } n(8)if(g[0][7-T]){ t=g[i
    g[2][g[i][T]]=T; n(R+i)o(e,m
    )o(t,2){ f[T+t+T]=(T["%#,"4"
    [e][t]=(5'<$=$8)Ih$=h918'9"
    s=T>9?m:(T&3)-3?15:36;o(e,s)o(t,2)c[T+19][e][t]="6*6,8*6.608.6264826668
    865:+(0(6+6-6/8,61638065678469.;88))()3(6,8*6.608.6264826668865:+;4)*-6-/61365,
    -6715690.5,;89,81+,0(23096/.40(8-7751)2)65;695(855(+*8)+,*++4(((6.608.626482666886\5:+;4+4)0(8/6/61638065678469.;88)-4,*4*8+4(((60/(6264826668865:+;4-61365676993-9:54\+/-14).:/347.+18*:1:-*0-975/)936.+4*,80987(887(0(*)4.*"*/4,*4*8+4(((6264826668865:\+/4/4-4+8-4)0(8)6365678469.;88)1/(6*6,6.60626466686:8)8-*818.8582/9863(+;"/**6,6.6\0626466686:4(8)8-*818.8582/9863(+;/.6.60626466686:8-818.8582/9864*4+4{0()}+;/.6062\6466686:8/8380/7844,4-4*4+4{0()}69+;/0626466686:818582/9864.4/4,4-4*4+4{0()}+;"[e+E+e+t]-40; E+=s+s; } n(45){ if(T>i) { v(2,T,7); v(46,T,7); } v(2+T,44,7); } T=0; o(e,
42)o(t,m)h[T][e][t]--; while(R+i) { s = D=0; if (r-R) { n(19) if (G[R+i][T]+i) V=T/2
; else if(G[R][T]+i) s++; if(s) { if(V>4){ V=9-V; D++; } V+=29; n(20) q(c[V][T][0],c
[V][T][i],D); } } n(19) if((L=G[R][T])+i) { O=T-L; e=O>9; t=e?18-O :0; o(K,((t&3)-3?
16:37)){ if(K){ L=c[t+19][K-i][0]; O=c[t+19][K-i][i] ; } q(L,O,K && e); } } if(s) q(
c[V][20][0], c[V][20][i], D); R--; } printf("\33[47;1f\33[?25h\33[40m"); return 0; }

```

# TOO MANY METHOD ARGUMENTS

```
- (id)showPresentation:(id)presentation  
          shuffled:(BOOL)shuffled  
enablingInteraction:(BOOL)enablingInteraction  
           type:(NSInteger)type  
      options:(NSDictionary *)options;
```

**COMPOSITION OVER INHERITANCE  
SINGLETONS AVOIDANCE  
DEPENDENCIES ISOLATION**

# THANKS!

@MACIEJOCZKO

# IMAGES ATTRIBUTION

Günter Hentschel <https://flic.kr/p/o6BJWK>

David Bleasdale <https://flic.kr/p/nNZRV>

Tony Unruh <https://flic.kr/p/akplcxlibber>

<https://flic.kr/p/eYXePj>

Peter Dutton <https://flic.kr/p/EraKt>

Rafa Garcés <https://flic.kr/p/78KsJo>  
cornflakegirl\_ <https://flic.kr/p/zynrD>

Jeff Eaton <https://flic.kr/p/aLN51T>

Procsilas Moscas <https://flic.kr/p/26eSr>