Código Limpo

Clean Code

Lorhan Sohaky

18 de abril de 2018

Quem sou eu?

- Graduando em ciência da computação pela UFSCar
- Gerente de projetos na CATI Jr.
- Técnico em informática pelo IFSP

Biblioteca *Sohaky.h*

- Duas funções
- 172 linhas

Código 1: Funções para realizar operações matemáticas

```
1  void melhora(double t, char r[100]);
2
3  double resolve(char *n);
```

Função *melhora*

```
1  void melhora(double t, char r[100]){
2          sprintf(r,"%f",t);
3          int x, tamanho=0,i,j;
4            tamanho=strlen(r);
5            for (x=0; x<tamanho; x++){
6                  if (r[x]== ',' || r[x]=='.'){
7                          r[x]='.';
8                          for(i=x;i<tamanho;i++){
9                                  if(r[i]=='.'){
10                                         j=x;
11                                 }else if(r[i]!='0'){
12                                         j=i+1;
13                                 }
14                          }
15                 }
16           }
17           r[j]='\0';
18 }
```

Função *resolve*

```
1  double resolve(char *n){
2          char op='\0',a[100]={'\0'},b[100]={'\0'},op2='\0',xc[100]={'\0'},dpa[100]={'\0'};
3          int j,po=0,tam,l,v=0,po2=0,v2=0,co=0,k,cof=0,coa=0,poa=0,pof=0,dp=0,t=0;
4          double ai=0,bi=0,ri=0;
5          tam=strlen(n);
6          //...
7  }
```

Não sabe programar? Então por qual motivo escreve códigos ruins?

Qual a importância?

- Códigos confusos diminuem a produtividade
- Alto grau de complexidade
- Baixa reutilização
- Alto custo de manutenção e implementação de novos recursos
- Dificuldade em encontrar erros

Nomes não descritivos

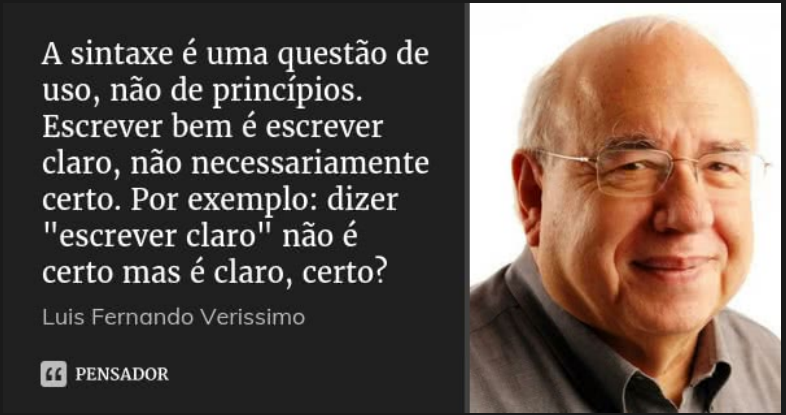


Figura: Retirada do site Pensador

Problemas

- Não saber o que a função faz de fato
- Dificuldade em encontrar erros
- Necessidade de olhar a implementação

Caso real

Código 2: Código em C ♯

```
1  private void tabPage1_Click(object sender, EventArgs e);
2
3  private void dateTimePicker2_ValueChanged(object sender, EventArgs e);
```

Caso real

Função *resolve*

```
1  double resolve(char *n){
2          char op='\0',a[100]={'\0'},b[100]={'\0'},op2='\0',xc[100]={'\0'},dpa[100]={'\0'};
3          int j,po=0,tam,l,v=0,po2=0,v2=0,co=0,k,cof=0,coa=0,poa=0,pof=0,dp=0,t=0;
4          double ai=0,bi=0,ri=0;
5          tam=strlen(n);
6          //...
7  }
```

Sabe o significado da variável *po*?

Sabe o significado da variável *po*?

Posição do primeiro operador

Exemplo

O que essa função faz?

Código 3: Declaração da função

```
1                    float calc (float b, float c);
```

Que significado a função *calc* tem?

Agora você sabe? Mas o que o nome dela te diz?

Código 4: Implementação da função

```
1  float calc (float b, float c){
2          float a = b * c;
3          return a;
4  }
```

Que significado a função *calc* tem?

- É uma função para multiplicar dois valores?
- É uma função para calcular a área de um retângulo?

O que essa função faz?

Código 5: Declaração da função de multiplicar

```
1  float multiplicar (float primeiroValor, float segundoValor);
```

Versão melhorada da função *calc*

Código 6: Código em C

```
1  float multiplicar (float primeiroValor, float segundoValor){
2          float resultado = primeiroValor * segundoValor;
3          return resultado;
4  }
```

Exemplo

O que essa função faz? O que t representa?

```
1  def getT():
2          return t / 60;
```

O que essa função faz? O que tempo representa?

```
1  def getTempo():
2          return tempo / 60;
```

Ainda tem dúvida sobre o que a função faz?

```
1  def getTempoEmMinutos():
2          return tempoEmSegundos / 60;
```

Exemplo

O que essa função faz?

Código 7: Código para logar no sistema SIGA

```
1  def loginSiga(username, password):
2                  #...
```

Código 8: Código para logar no sistema SIGA

```
1  def loginSiga(username, password):
2          driver = webdriver.PhantomJS()
3          driver.set_window_size(1024, 768) # optional
4          driver.get('https://sistemas.ufscar.br/siga/')
5
6          formularioLogin = driver.find_element_by_id("login")
7          campoUsuario = driver.find_element_by_name("login:usuario")
8          campoSenha = driver.find_element_by_name("login:password")
9          botaoLogin = driver.find_element_by_name("login:j_idt39")
10
11         #...
12
13         return toJSON(driver.page_source.encode('ascii', 'ignore'))
```

Solução

- Escolha um nome que descreva o que a função faz
- Os nomes das variáveis devem expressar significado
- Evite usar siglas

Módulos grandes (não especializados)

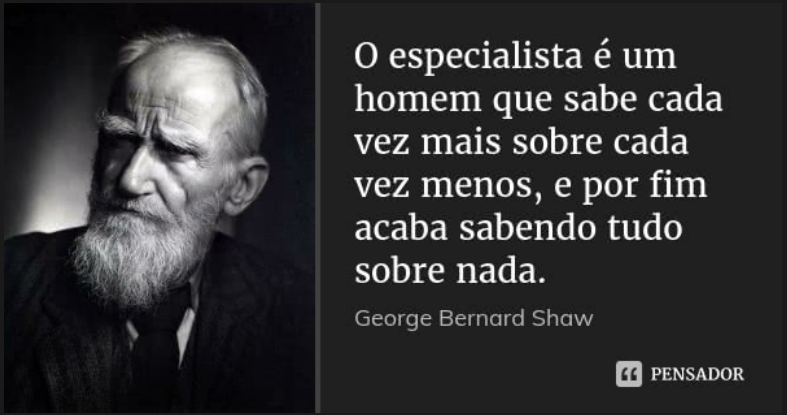


Figura: Retirada do site Pensador

Problemas

- Fazer diversas coisas (resolver diversos problemas)
- Alto grau de complexidade
- Baixa reutilização
- Alto custo de manutenção
- Podem não ser tão eficientes
- Podem esconder problemas
- Modificar um trecho que não deveria

Caso real

Onde está o problema?

```
1  [...]
2  FILE *pArquivo;
3  int nLinhas = 0;
4  char caracter;
5  char palavra[100];
6
7  if((pArquivo=fopen("meuArquivo.txt", "r")) != NULL){
8          while(fscanf(pArquivo, "%c", &caracter) != EOF){
9                  if(caracter == '\n'){
10                   nLinhas+=1;
11                 }
12         }
13         for(i=0; i<rand()%nLinhas+1; i++){
14                 fscanf(pArquivo, "%s\n", &palavra);
15         }
16 fclose(pArquivo);
17 [...]
```

Onde está o problema?

```
1  [...]
2  FILE *pArquivo;
3  int nLinhas = 0;
4  char caracter;
5  char palavra[100];
6
7  if((pArquivo=fopen("meuArquivo.txt", "r")) != NULL){
8          while(fscanf(pArquivo, "%c", &caracter) != EOF){
9                  if(caracter == '\n'){
10                   nLinhas+=1;
11                 }
12         }
13         //Aqui está o problema, pois o ponteiro pArquivo não está mais na posição inicial
14         for(i=0; i<rand()%nLinhas+1; i++){
15                 fscanf(pArquivo, "%s\n", &palavra);
16         }
17 fclose(pArquivo);
18 [...]
```

Solução porca

```
1  [...]
2  FILE *pArquivo;
3  int nLinhas = 0;
4  char caracter;
5  char palavra[100];
6
7  if((pArquivo=fopen("meuArquivo.txt", "r")) != NULL){
8          while(fscanf(pArquivo, "%c", &caracter) != EOF){
9                  if(caracter == '\n'){
10                         nLinhas+=1;
11                 }
12         }
13
14         rewind(pArquivo);
15         for(i=0; i<rand()%nLinhas+1; i++){
16                 fscanf(pArquivo, "%s\n", &palavra);
17         }
18 fclose(pArquivo);
19 [...]
```

Solução ideal

```
1  void lerLinhaSorteada(char *nomeDoArquivo){
2          int quantidadeDeLinhas = contarNumeroDeLinhas(nomeDoArquivo);
3
4          if(quantidadeDeLinhas == -1)
5                  return;
6
7          int linha = sortear(quantidadeDeLinhas);
8
9          FILE *arquivo = fopen(nomeDoArquivo, "r");
10
11         moverPonteiroArquivo(&arquivo, linha);
12
13         char palavra[100];
14         fscanf(arquivo,"%s\n",palavra);
15         fclose(arquivo);
16
17         printf("%s\n",palavra);
18 }
```

Solução ideal

```c
int contarNumeroDeLinhas(char *nomeDoArquivo){
    int quantidadeDeLinhas = 0;
    char caracter;
    FILE *arquivo = fopen(nomeDoArquivo, "r");

    if(arquivo ==NULL){
        quantidadeDeLinhas = -1;
    }else{
        while(fscanf(arquivo, "%c", &caracter) != EOF){
            if(caracter == '\n'){
                quantidadeDeLinhas++;
            }
        }

        fclose(arquivo);
    }

    return quantidadeDeLinhas;
}
```

Solução ideal

```
1  int sortear(int limite){
2          return rand()%limite+1;
3  }
```

Solução ideal

```
1  void moverPonteiroArquivo(FILE **arquivo, int linha){
2      char palavra[100];
3
4      for(int i=0; i<linha-1; i++){
5          fscanf(*arquivo, "%s\n",&palavra);
6      }
7  }
```

Solução

- Criar funções especializadas

Comentários

Figura: Retirada do site Pensador

Problemas

- Comentários tendem a mentir
- Você não sabe pra que serve um código comentado
- Comentários tentam explicar o que não está claro em seu código
- * Comentários podem ser lixos no código

Exemplo

O que essa função deve fazer?

Código 9: Comentário para explicar o que a função faz

```
1  /*
2          Função para contar o tamanho da string
3          char *string String a ter seu tamanho contado
4          Retorno tamanho da string
5  */
6  int contarTamanho(char *string);
```

O que a função faz de fato?

Código 10: Comentário para explicar o que a função faz

```c
/*
        Funcao para contar o tamanho da string
        char *string String a ter seu tamanho contado
        Retorno tamanho da string
*/
int contarTamanho(char *string){
        int tamanho=0;
        int i=0;

        while(string[i]!='\0' && i < strlen(string)){
                if(string[i]==' '){
                        string[i]='\0';
                }
                tamanho++;
                i++;
        }

        return tamanho;
}
```

Caso real

Código 11: Comentários em excesso

```java
/**
* Metodo construtor de Ocorrencia.
*
* @param dataHora Data e hora da ocorrência.
* @param transitavelVeiculo Transitável por meio de Veiculo.
* @param transitavelAPe Transitável a pé.
* @param descricao Descrição da ocorrência.
* @param qtdExistente Quantidade de existente.
* @param qtdInexistente Quantidade de inexistente.
* @param qtdCasoEncerrado Quantidade de caso encerrado.
* @param pkOcorrencia Número de identificação da ocorrência
* @param tipo Tipo da ocorrência.
* @param user Objeto Usuário.
*/
public Ocorrencia(Date dataHora, boolean transitavelVeiculo, boolean transitavelAPe,
                        String descricao, long qtdExistente, long qtdInexistente,
                        long qtdCasoEncerrado, long pkOcorrencia, Tipo tipo, Usuario user) {
        this.dataHora = (Date) dataHora.clone();
          //...
}
```

Exemplo

Código 12: Comentário para explicar cada trecho de código

```
1  /*
2          Função para calcular a área de um retângulo
3          float b um lado do retângulo
4          float c outro lado do retângulo
5          Retorno área do retângulo
6  */
7  float calc (float b, float c){
8          float a = b * c; // a recebe b * c
9          return a; // retorna a
10 }
```

Exemplo

O que fazer com código comentado?

Código 13: Código comentado

```
1  module inicial ( giro, entrada, saida, metais, ledVerde, ledVermelho, display, clock );
2          input giro, entrada, saida, metais, clock;
3          output [1:0] ledVerde, ledVermelho;
4          output [6:0] display;
5
6          reg [3:0] tmp;
7          [...]
8          tmp = { giro, entrada, saida, metais };
9          /*
10                 tmp[3] = giro;
11                 tmp[2] = entrada;
12                 tmp[1] = saida;
13                 tmp[0] = metais;
14         */
15
16         [...]
17 endmodule
```

Código 14: Código comentado

```
1  module inicial ( giro, entrada, saida, metais, ledVerde, ledVermelho, display, clock );
2          input giro, entrada, saida, metais, clock;
3          output [1:0] ledVerde, ledVermelho;
4          output [6:0] display;
5
6          reg [3:0] tmp;
7          [...]
8          tmp = { giro, entrada, saida, metais };// Equivalente ao código comentado
9          /*
10                 tmp[3] = giro;
11                 tmp[2] = entrada;
12                 tmp[1] = saida;
13                 tmp[0] = metais;
14         */
15
16         [...]
17 endmodule
```

Código repetido

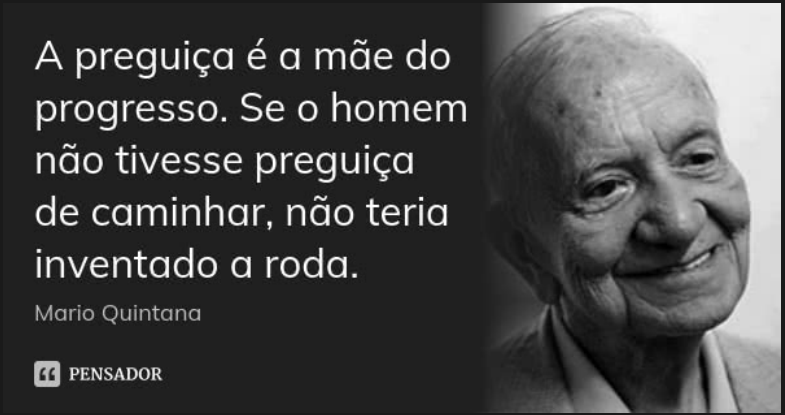


Figura: Retirada do site Pensador

Problemas

- Dificulta a manutenção
- As funções podem ser diferentes
- Dificuldade de localizar o problema
- As vezes não é tão eficiênte

Caso real

Classe *Form_GerenciarAcomodacoes*

```
1  //Função para validar as strings (não permitir caracteres especiais)
2  public bool validaStr(string str)
3  {
4          Regex r = new Regex("^[a-zA-ZáéíóúàèìòùâêîôûãõçÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÃÕÇ0-9 -]*$");
5      return r.IsMatch(str);
6  }
```

Classe *Form_GerenciarFuncionario*

```
1  //Função para validar as strings (nao permitir caracteres especiais)
2  public bool validaStr(string str)
3  {
4          Regex r = new Regex("^[a-zA-ZáéíóúàèìòùâêîôûãõçÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÃÕÇ -]*$");
5      return r.IsMatch(str);
6  }
```

Classe *Form_GerenciarTarifas*

```
1  //Função para validar as strings (nao permitir caracteres especiais)
2  public bool validaStr(string str)
3  {
4          Regex r = new Regex("^[a-zA-ZáéíóúàèìòùâêîôûãõçÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÃÕÇ0-9 -]*$");
5      return r.IsMatch(str);
6  }
```

Elas eram iguais?

Exemplo

Função para substituir vogais por espaço

```
1  bool substituirVogaisPorEspaco(char *string){
2          char vogais[]="aeiouAEIOU";
3          bool substituiu = false;
4
5          for( int i=0; i<strlen(string); i++){
6                  for( int j=0; j<strlen(vogais); j++){
7                          if(string[i] == vogais[j]){
8                                  string[i] = ' ';
9                                  substituiu = true;
10                         }
11                 }
12         }
13
14         return substituiu;
15 }
```

Função para substituir números pela letra A

```
1  bool substituirNumerosPorLetras(char *string){
2          char numeros[]="0123456789";
3          bool substituiu = false;
4
5          for( int i=0; i<strlen(string); i++){
6                  for( int j=0; j<strlen(numeros); j++){
7                          if(string[i] == numeros[j]){
8                                  string[i] = 'A';
9                                  substituiu = true;
10                         }
11                 }
12         }
13
14         return substituiu;
15 }
```

NOVA Função para substituir vogais por espaço

```
1  bool substituirVogaisPorEspaco(char *string){
2          char vogais[] = "aeiouAEIOU";
3
4          bool substituiu = false, tmp;
5
6          for( int i=0; i<strlen(vogais); i++){
7                  tmp = substituirCaractere(string, vogais[i], ' ');
8                  if(tmp){
9                          substituiu = tmp;
10                 }
11         }
12
13         return substituiu;
14 }
```

NOVA Função para substituir números pela letra A

```
1  bool substituirNumerosPorLetras(char *string){
2          char numeros[]="0123456789";
3
4          bool substituiu = false, tmp;
5
6          for( int i=0; i<strlen(numeros); i++){
7                  tmp = substituirCaractere(string, numeros[i], 'A');
8                  if(tmp){
9                          substituiu = tmp;
10                 }
11         }
12
13         return substituiu;
14 }
```

NOVA Função para substituir caracteres

```
1  bool substituirCaractere(char *string, char antigoCaractere, char novoCaractere){
2          bool substituiu = false;
3
4          for( int i=0; i<strlen(string); i++){
5                  if(string[i] == antigoCaractere){
6                          string[i] = novoCaractere;
7                          substituiu = true;
8                  }
9          }
10
11         return substituiu;
12 }
```

Solução

- Remover a repetição e reorganizar

Quantidade de parâmetros

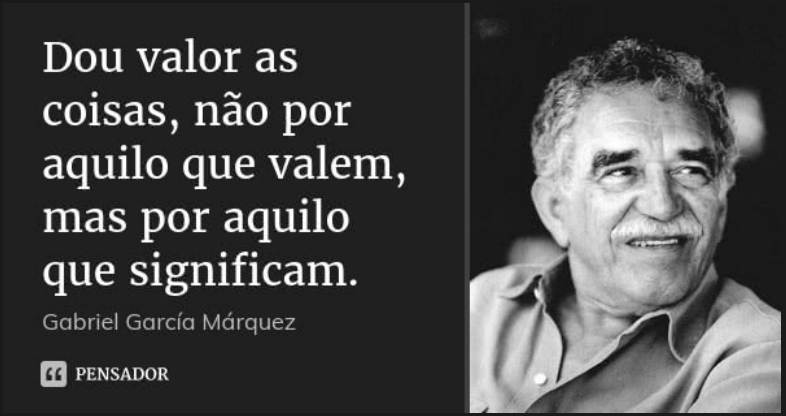


Figura: Retirada do site Pensador

Problemas

- Necessidade de criar objetos
- Não saber o que é realmente necessário

Caso real

Código 15: Trabalho para reserva de poltrona em avião

```
int mod2(int rg,int p, int f, int np, int nf, int opt){
        FILE *arq;
        FILE *arq2;
        int i=0,j=0,tam;
        char ch;
        struct cli{
                int rg;
                char nome[30];
                int p,f,d,m,a;
        };
        struct cli cad;
        arq=fopen("cli.dat","rb");
        if(arq==NULL){
                return 100;
        }
        [...]

}
```

Exemplo

```
class Produto{
        private:
                int id;
                string nome;
                string descricao;
                float preco;

        public:
                Produto(int id, string nome, string descricao, float preco);
                Produto(string nome, string descricao, float preco);
};
```

```
class ProdutoDAO : protected ConexaoBancoDeDados{
        public:
                bool cadastrar(string nome, string descricao, float valor);
};
```

```
produtoDAO.cadastrar(  "Livro - Clean Code" , "100 paginas", 50  );
```

```
class ProdutoDAO : protected ConexaoBancoDeDados{
        public:
                bool cadastrar(Produto produto);
};
```

```
Produto produto( "Livro - Clean Code" , "100 paginas", 50 );
produtoDAO.cadastrar( produto );
```

```
class ProdutoDAO : protected ConexaoBancoDeDados{
        public:
                bool cadastrar(Produto produto);
                Produto pegarInformacoes(Produto produto);
};
```

Quero pegar as informações do produto, mas quais dados devo fornecer?

```
1  Produto produto = produtoDAO.pegarInformacoes( ? );
```

Suposição dos campos necessários

```
1  int idProduto = 10;
2  string nomeProduto = "Clean Code";
3  produtoAPesquisar( idProduto, nomeProduto, "", 0 );
4  produto = produtoDAO.pegarInformacoes( produtoAPesquisar );
```

Solução

```
class ProdutoDAO : protected ConexaoBancoDeDados{
        public:
                bool cadastrar(Produto produto);
                Produto pegarInformacoes(int id);
};
```

```
Produto produto = produtoDAO.pegarInformacoes( idProduto );
```

Exemplo

```c
typedef struct {
        int x,y;
} Ponto;

void desenhar_triangulo(Ponto ponto1, Ponto ponto2, Ponto ponto3, char *corLinha,
                                                    char *corPreenchimento){
        [...]
}
```

Solução

```
1  typedef struct {
2      int x,y;
3  } Ponto;
4
5  typedef struct {
6      char corPreenchimento[100];
7      char corLinha[100];
8      Ponto vertices[3];
9  } Triangulo;
10
11 void desenhar(Triangulo triangulo){
12     [...]
13 }
```

Solução

- Reavaliar para encontrar o que se encaixa melhor

Falta de padronização

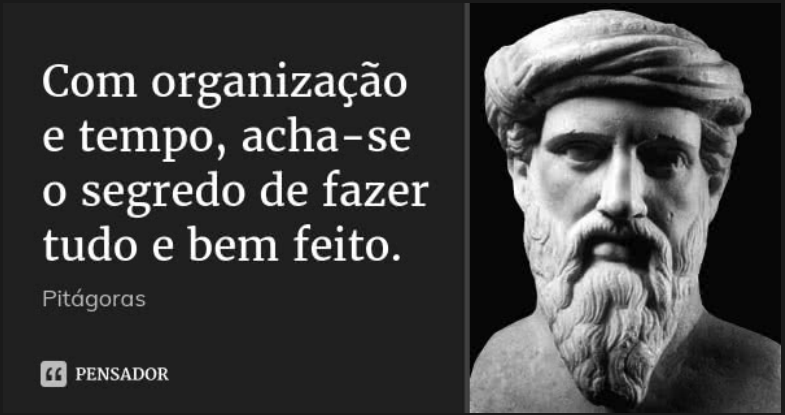


Figura: Retirada do site Pensador

Problemas

- Necessidade de olhar a implementação

Exemplo

```
1           bool trocarVogaisPorEspaco(char *string);
2
3           bool substituirNumerosPorLetras(char *string);
```

```
1  bool trocarVogaisPorEspaco(char *string){
2          char vogais[]="aeiouAEIOU";
3
4          bool substituiu, tmp;
5
6          for( int i=0; i<strlen(vogais); i++){
7                  tmp = substituirCaractere(string, vogais[i], ' ');
8
9                  if(tmp){
10                         substituiu = tmp;
11                 }
12         }
13
14         return substituiu;
15 }
```

```
1  bool substituirNumerosPorLetras(char *string){
2          char numeros[]="0123456789";
3
4          bool substituiu, tmp;
5
6          for( int i=0; i<strlen(numeros); i++){
7                  tmp = substituirCaractere(string, numeros[i], 'A');
8
9                  if(tmp){
10                         substituiu = tmp;
11                 }
12         }
13
14         return substituiu;
15 }
```

```
class Usuario{
        private:
                int id;
        public:
                int getId() const{
                        return id;
                }
};
```

```
class Cliente{
        private:
                ID id;
        public:
                ID getId() const{
                        return id;
                }
};
```

Problema

```
1  int idCliente = cliente.getId();
2  int idUsuario = usuario.getId().getValue();
```

Solução

```
1  class Cliente{
2          private:
3                  ID id;
4          public:
5                  int getId() const{
6                          return id.getValue();
7                  }
8  };
```

Solução

- Padronizar

Não possuir testes

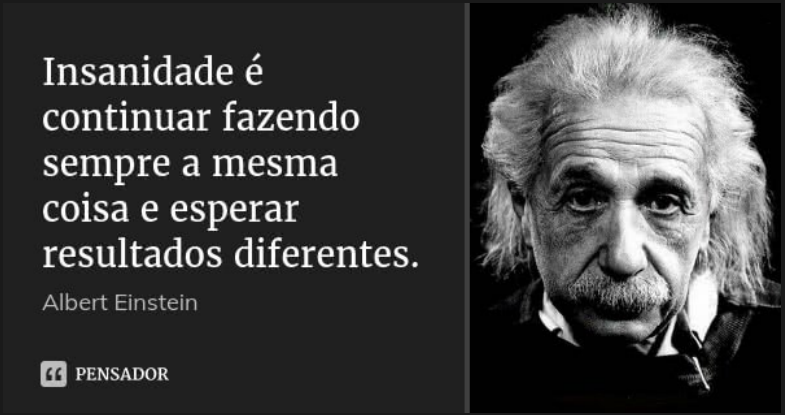


Figura: Retirada do site Pensador

Problemas

- Necessidade de testar manualmente
- Durante as atualizações pode aparecer um problema e não ser identificado de imediato

```c
bool testSubstituirVogaisPorEspaco(void){
    char *string ="abcdef";
    char *resultadoEsperado = " bcd f";

    substituirVogaisPorEspaco(string); // retorna string = "AbcdEf"

    if( !strcmp( resultadoEsperado, string) ){
        printf("Erro no teste da função substituirVogaisPorEspaco\n");
        return false;
    }
    return true;
}
```

Solução

- Criar testes

Não separação em arquivos

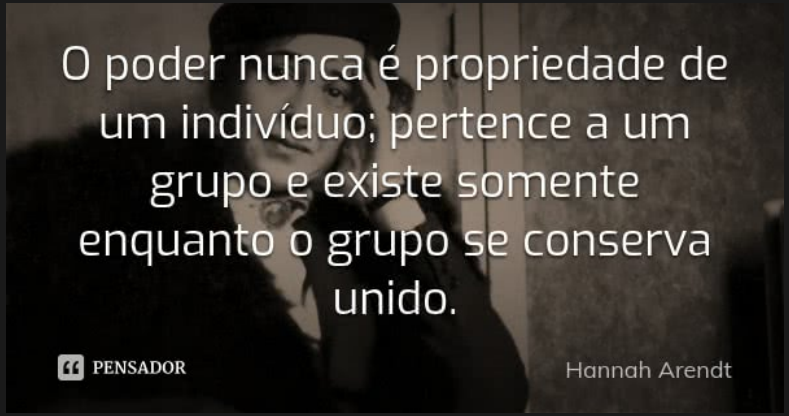


Figura: Retirada do site Pensador

Problemas

- Códigos "misturados"

Exemplo

Código 16: Classe lidando com "escopos diferentes"

```
class ConexaoBancoDeDados{
        public:
                bool cadastrarProduto( Produto produto );
                bool cadastrarCliente( Cliente cliente );
                bool cadastrarUsuario( Usuario );

                Produto pegarInformacoesProduto( int idProduto );
                [...]
};
```

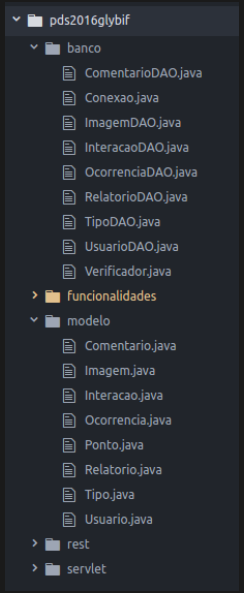
```
conexaoBancoDeDados.cadastrarUsuario( usuario );
conexaoBancoDeDados.cadastrarCliente( cliente );
conexaoBancoDeDados.cadastrarProduto( produto );
```

Solução

```
1  class ProdutoDAO : protected ConexaoBancoDeDados{
2          public:
3                  bool cadastrar( Produto produto );
4                  Produto pegarInformacoes( id );
5  };
6
7  class ClienteDAO : protected ConexaoBancoDeDados{
8          public:
9                  bool cadastrar( Cliente cliente );
10                 Cliente pegarInformacoes( int id );
11 };
12
13 class UsuarioDAO : protected ConexaoBancoDeDados{
14         public:
15                 bool cadastrar( Usuario usuario );
16                 Usuario pegarInformacoes( int id );
17 };
```

```
1  Produto produto = produtoDAO.pegarInformacoes( idProduto );
```

Exemplo de projeto em Java organizado de acordo com "proximidade"



Solução

- Agrupar as coisas (funções, classes, etc) de modo que as mesmas estejam juntas de acordo com sua proximidade

Recompensa

- Códigos expressivos
- Baixa complexidade
- Reaproveitamento de código
- Baixo custo de manutenção
- Maior eficiência
- Códigos testáveis

Seus códigos devem funcionar como uma caixa preta.

Bons códigos não dependem da linguagem de programação.

Só pelo fato do programa funcionar não quer dizer que ele é bom.

Deixe o código mais limpo do que estava antes de você mexer.

Gambiarras são seu fracasso profissional e do projeto.

Não reinvente a roda.


Steve McConnell.
Code Complete.
Vol. 2, 2004.


Robert C Martin.
Clean Code.
Vol. 1, 2008.