



THE ONE

Exercises in Programming Style
Capítulo 9

Grupo 1

Alice Lima
Carlos Eduardo
Giovana Pinho
João Francisco
João Gabriel

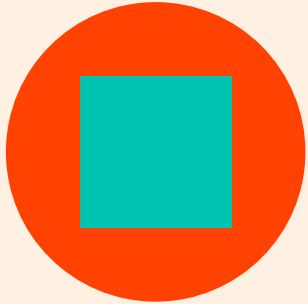
CARACTERÍSTICAS

- ❑ Uma **abstração** que permite conversão de valores;
- ❑ Operações da abstração:
 - ❑ “**embrulhar**” valores, tornando-os abstração;
 - ❑ Vincular os valores a funções (é estabelecida uma **sequência de funções**);
 - ❑ “**desembrulhar**” o valor, verificação de seu resultado final.

O PROCESSO



valor inicial



funções

valor final



Wordcount

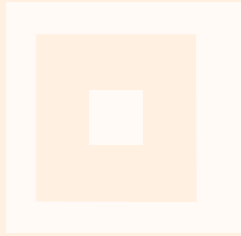
A classe **TFTheOne** possui um atributo do tipo `any` e três métodos.

any value, variável que guarda o valor

TFTheOne, um construtor

bind, método que invoca a sequência de funções que alteram o atributo

printme, mostra o resultado final



Wordcount

Um pipeline das funções ao lado é realizado



read_file
filter_char
normalize
scan
remove_stop_words
frequencies
sort_



read_file leitura do arquivo de texto

```
std::string read_file(std::string file){
    std::ifstream myfile(file);
    std::string aux_file;
    int cont = 0;

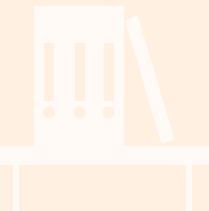
    while(!myfile.eof()){
        std::string aux;
        getline(myfile, aux);
        if(cont == 0) aux_file = aux;
        else aux_file = aux_file + " " + aux;
        cont++;
    }

    return aux_file;
}
```



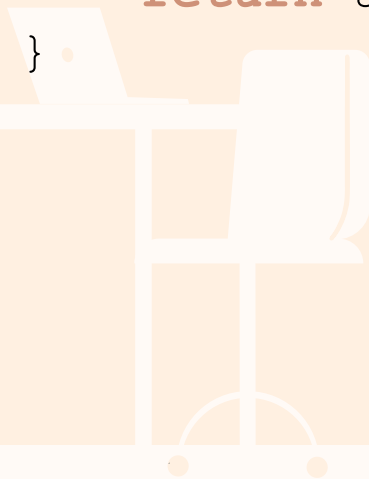
filter_char remoção de caracteres especiais

```
std::string filter_char(std::string aux_file){  
    for(int i = 0; i < aux_file.size(); i++){  
        if(!isalnum(aux_file[i])){  
            aux_file[i] = 32;  
        }  
    }  
    return aux_file;  
}
```



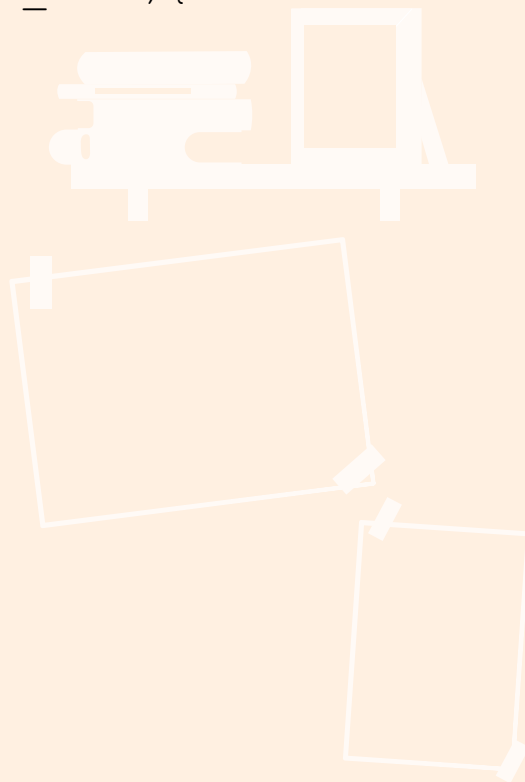
normalize todas as palavras em caracteres minúsculos

```
std::string normalize(std::string aux_file){  
    for(int i = 0; i < aux_file.size(); i++){  
        aux_file[i] = tolower(aux_file[i]);  
    }  
    return aux_file;  
}
```



scan fragmentação do texto em palavras

```
std::vector<std::string> scan(std::string aux_file){  
    std::vector<std::string> aux;  
    std::string aux2;  
    char * split;  
    char str[aux_file.size()];  
  
    strcpy(str, aux_file.c_str());  
    split = strtok(str, " ,");  
  
    while(split != NULL){  
        aux.push_back(split);  
        split = strtok(NULL, " ,");  
    }  
  
    return aux;  
}
```



remove_stop_words remoção de palavras vazias do dicionário

```
std::vector<std::string> remove_stop_words(std::vector<std::string>
list){
    std::vector<std::string> stop_words;
    std::string aux = "stop_words.txt", aux2, aux3;

    aux2 = read_file(aux);
    aux3 = normalize(aux2);
    stop_words = scan(aux3);
    for(int i = 0; i < list.size(); i++){
        for(int j = 0; j < stop_words.size(); j++){
            if(stop_words[j] == list[i]){
                list.erase(list.begin()+i);
                i--;
            }
        }
    }
    return list;
}
```

frequencies ocorrência de cada palavra

```
std::vector<std::pair<std::string, int>> frequencies(std::vector<std::string> aux){
    std::vector<std::pair<std::string, int>> freq;
    std::vector<bool> visited(aux.size(), false);

    for(int i = 0; i < aux.size(); i++){
        if(!visited[i]){
            int cont = 1;
            visited[i] = true;
            for(int j = 0; j < aux.size(); j++){
                if(aux[i] == aux[j] and !visited[j]){
                    visited[j] = true;
                    cont++;
                }
            }
            freq.push_back(make_pair(aux[i], cont));
        }
    }
    return freq;
}
```

sort_ ordenação por frequência

```
bool exe(std::pair<std::string, int> x, std::pair<std::string,  
int> y){  
    if(x.second == y.second){  
        return (x.first < y.first);  
    }  
    return (x.second > y.second);  
}
```

```
std::vector<std::pair<std::string, int>>  
sort_(std::vector<std::pair<std::string, int>> freq_words){  
    std::sort(freq_words.begin(), freq_words.end(), exe);  
    return freq_words;  
}
```



Obrigad@!

alice.lima@aluno.unb.br
carlos-oliveira.co@aluno.unb.br
giovana.pinho@aluno.unb.br
joao.targino@aluno.unb.br
joao.saraiva@aluno.unb.br

Este foi um trabalho realizado na disciplina Técnicas de Programação 2, Turma
A - Professor Rodrigo Bonifácio

CREDITS: This presentation template was created by
Slidesgo, including icons by **Flaticon**, infographics &
images by **Freepik**