

# **Декомпозиция программ**

Разработка ПО систем управления

Весна 2017 г.



```
C:\> spellcheck.exe
```

```
After time adrift amogn open stars
```

```
Along tides of lighth and through shoals of dust  
I will retur to where I began.
```

```
^Z
```

```
After time adrift amogn open stars
```

```
^^^^^
```

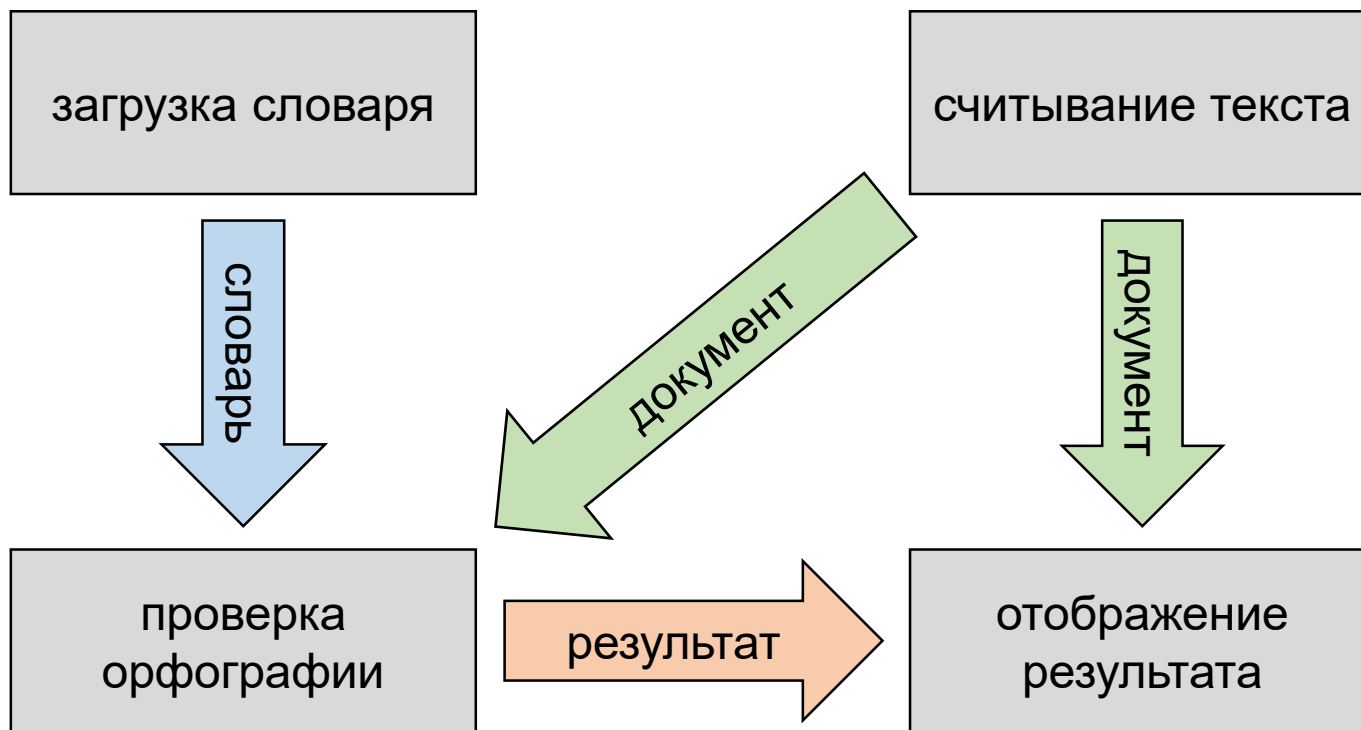
```
Along tides of lighth and through shoals of dust
```

```
^^^^^
```

```
I will retur to where I began.
```

```
^^^^^
```

# Data flow (потоки данных)



# Выбор типов

- Должен быть удобен для обработки.
- Вопросы:
  - Что нужно хранить?
  - Что нужно с данными делать?
  - Какие есть технические ограничения?
    - Объем памяти, скорость vs размер и т. п.

# Выбор типов: документ

- Что нужно хранить?
  - Текст без потерь.
- Что нужно с данными делать?
  - Искать неизвестные **слова** — нужно разбиение по словам.
  - **Подчеркивать** ошибки в строках:
    - нужно разбиение по строкам;
    - нужны позиции слов в строках.
  - ... в **оригинальном** тексте — нельзя терять часть ввода (например, разделители слов).

```
struct Document {  
    vector<string> lines;           // Строки текста.  
    vector<vector<Span>> words;    // Места слов в каждой строке.  
};
```

# Выбор типов: место в строке

- Что нужно хранить?
  - Позиции интервала (от какого до какого символа).
- Что нужно с данными делать?
  - Отображать линии **длиной** со слово, начиная с **позиции начала** слова.

```
struct Span {  
    size_t start;  
    size_t length;  
};
```

# Выбор типов: словарь

- Что нужно хранить?
  - Слова.
- Что нужно с данными делать?
  - **Искать** заданное слово.
- Упрощение: не будем оптимизировать поиск.
  - «Premature optimization is the root of all evil»

**using** Dictionary = vector<string>;

# Выбор типов: результат проверки

- Что нужно хранить?
  - **Места**, где встречены неизвестные слова — то же самое, что и место в строке.
- Что нужно с данными делать?
  - **В каждой строке** подчеркивать место ошибки линией длиной со слово в позиции слова:
    - нужна группировка по строкам;
    - по сути, это позиции слов, но не всех.

**using** Check = vector <vector <Span > >;



# Декомпозиция: загрузка словаря

- Что на входе?
  - Путь к файлу со словами — `string`.
    - `const string&`
- Что на выходе?
  - Словарь — `Dictionary`.
- Как работает?
  - Открыть файл.
  - Пока файл не закончился,
    - считывать очередное слово
    - и заносить в словарь.

# Код: загрузка словаря

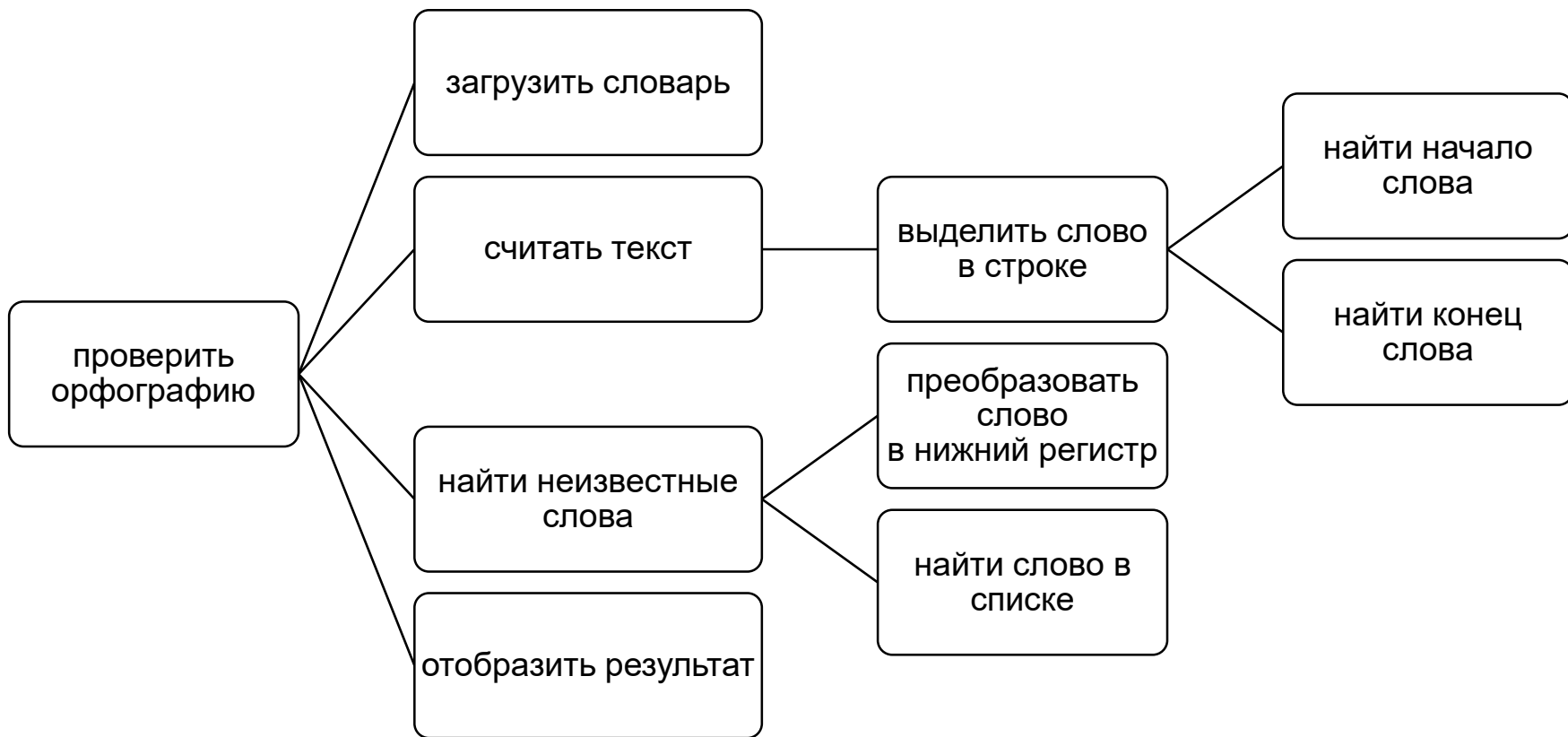
Dictionary

```
load_dictionary(  
    const string& path) {  
    Dictionary dictionary;  
    ifstream input{ path };  
    while ( !input.eof() ) {  
        string word;  
        input >> word;  
        dictionary.push_back( word );  
    }  
    return dictionary;  
}
```

- Возвращает Dictionary.
- Называется load\_dictionary.
- Принимает ссылку на неизменяемую строку.
- Открывает файл.
- Пока файл не закончится,
  - считывает слово
  - и добавляет его в словарь.

# Декомпозиция: считывание текста

- Что на выходе?
  - Документ — `Document`.
- Что на входе?
  - Если читать из `cin`, ничего не нужно.
- Как работает?
  - Считать строку.
  - Добавить к строкам документа.
  - Разбить строку на слова.
  - Повторять, пока не достигнут конец `cin`.



# Декомпозиция: разбиение строки на слова

- Есть **позиция, до которой строка разобрана**, изначально 0.
- Найти первый не-разделитель после этой позиции.
  - Если не найден, больше слов в строке нет,
  - иначе найдена **позиция начала слова**.
- Найти первый разделитель с начала слова.
  - Если не найден, слово продолжается до конца строки.
  - Это будет **позиция конца слова**.
- Сохранить интервал из позиции **начала** и **конца** слова.
- Считать строку **разобранной до позиции конца слова**.
- Если строка **разобрана до конца**, выйти.

# Hints: разбиение на слова

- Слова разделены символами из набора:  
`char separators[] = " ,.!?;()";`
- `string line;`
  - `line.find_first_of(separators, position)` —  
позиция первого символа из `separators` в `line`,  
начиная с позиции `position`.
  - `line.find_first_not_of`
  - `string::npos` возвращается, если символ не найден.