Wyrażenia regularne

```
#include <regex>
```

Michał Stolorz
Łukasz Latusik

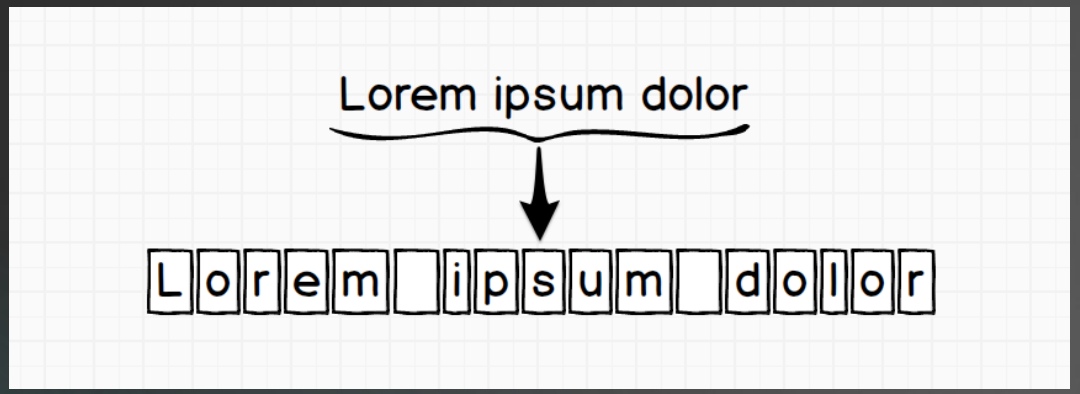
Wyrażenie regularne co to takiego?

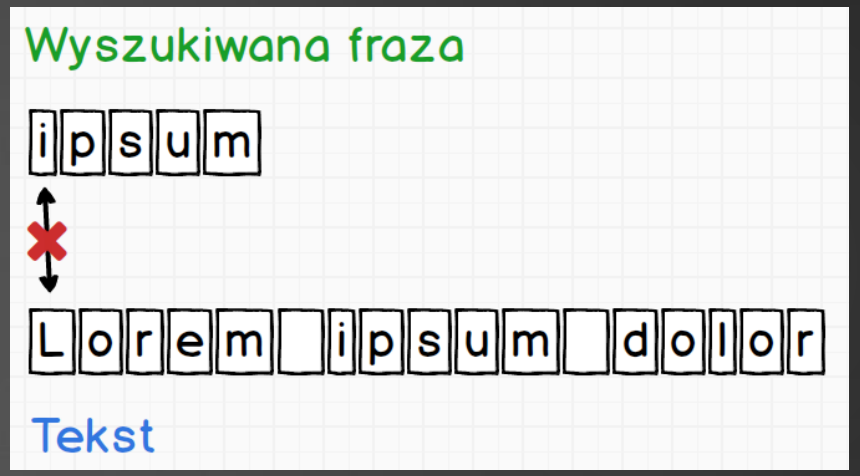
Wyrażenia regularne (ang. regular expressions) - wzorce, które opisują łańcuchy symboli.

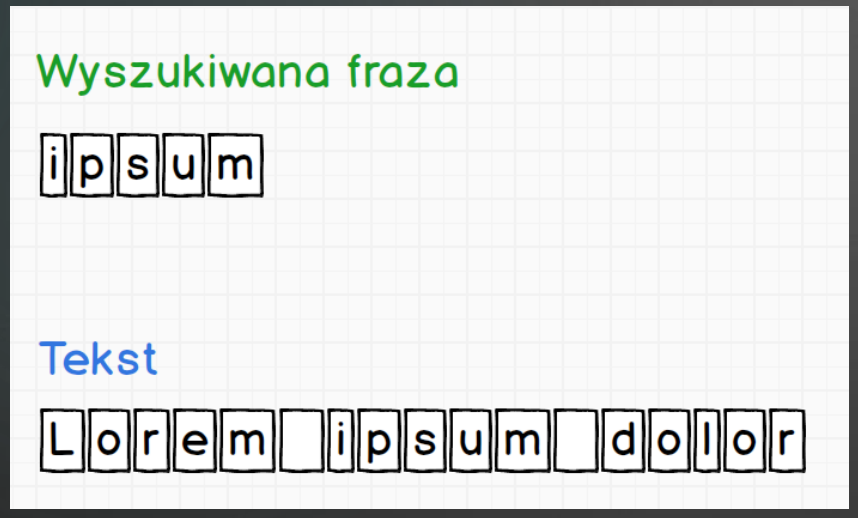
Wprowadzono je w standardzie C++11.

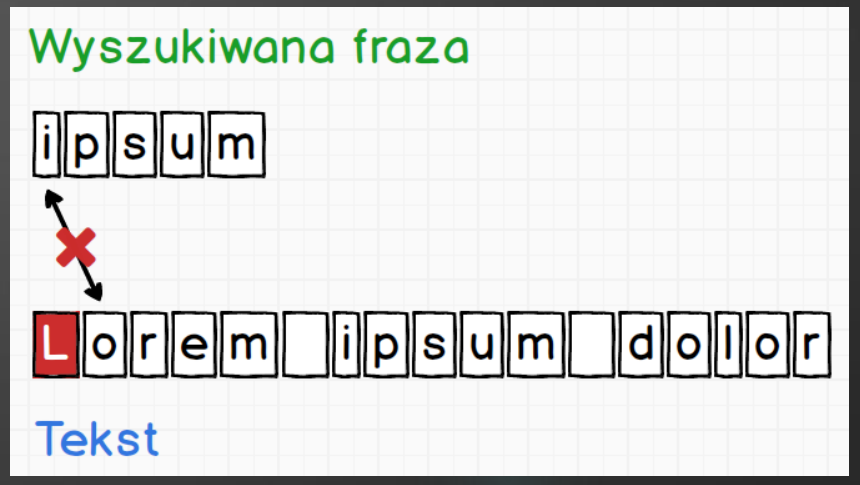
Wyrażenia regularne korzystają z gramatyk.

Sposób działania

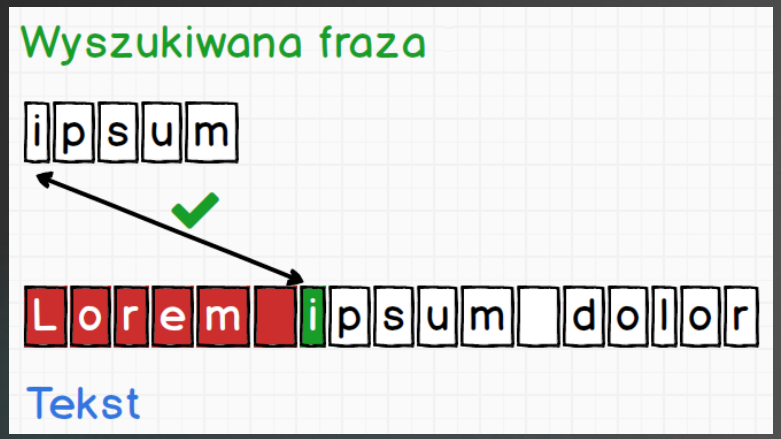


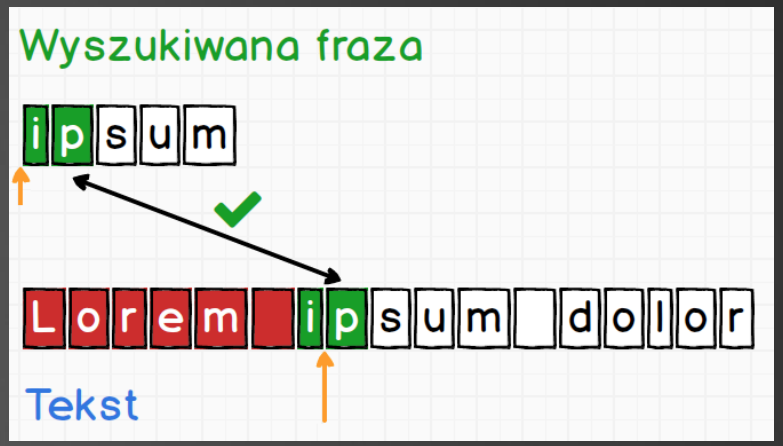


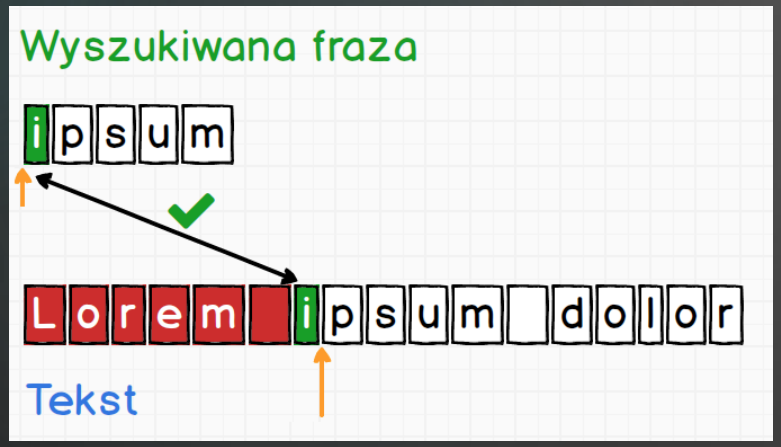


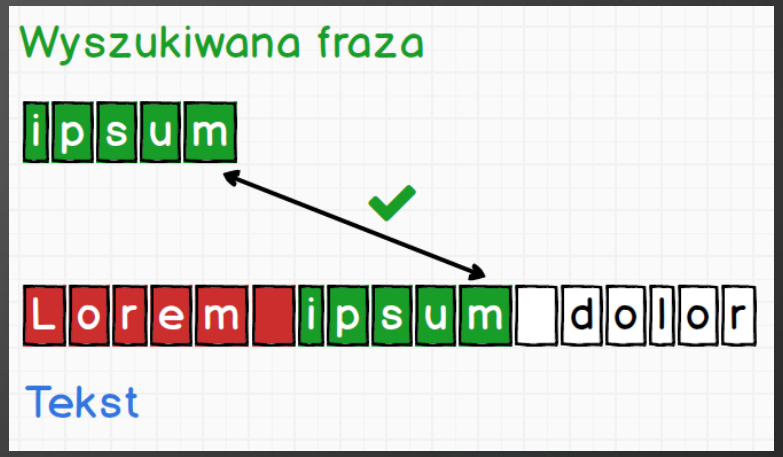


Sposób działania c.d.









ECMAScript

```
"[A-F][[:punct:]][[:digit:]]"
```

Domyślna gramatyka dla wyrażeń regularnych.

European Association for Standardizing Information and Communication Systems

```
"g{5}dc{3,7}"
```

ECMA-262

Składnia

Znaki specjalne	
\d	Cyfra dziesiętna
\D	Negacja powyższego
\w	Litera, cyfra lub „_”
\W	Negacja powyższego
Przy używaniu w kodzie: \\w	

Kwantyfikatory	
*	Poprzedni element powinien wystąpić zero lub więcej razy
+	Poprzedni element powinien wystąpić raz lub więcej razy
?	Poprzedni element powinien wystąpić zero lub raz
{n}	Poprzedni element powinien wystąpić n razy
{n, }	Poprzedni element powinien wystąpić co najmniej n razy
{n, m}	Poprzedni element powinien wystąpić co najmniej n ale nie więcej niż m razy

Klasy	
[:alnum:]	Znaki alfanumeryczne
[:alpha:]	Litery
[:digit:]	Cyfry
[:lower:]	Małe litery
[:punct:]	Znaki przestankowe
[:space:]	Spacja
[:upper:]	Duże litery
Przy używaniu w kodzie: [[:nazwa_klasy:]]	

Inne przydatne…	
(<podwzorzec>)	Podwyrażenie, które może być później używane (jako grupa przechwytywania)
|	Alternatywa
.	Dowolny znak
[ ]	Zakres symboli

http://www.cplusplus.com/reference/regex/ECMAScript/

typedef basic_regex<char> regex

- Obiekt klasy regex jest instancją klasy basic_regex.
- Posługujemy się nimi przy wyszukiwaniu wzorca w tekście.

typedef match_results<string::const_iterator> smatch

- Instancja klasy match_result.
- Przechowuje dopasowanie wyszukane w sekwencji
- Wypełniana przez: regex_match, regex_serach, regex_iterator

smatch

smatch s

s.str() – zwraca znalezioną sekwencję jako string

s.suffix() – zwraca resztę stringa po wykrytej sekwencji

s.prefix() – zwraca resztę stringa przed wykrytą sekwencją

W następnych slajdach będą nam potrzebne dane obiekty:

```
regex rx;
string napis;
string zmiana;
smatch wynik;
```

regex_match

Metoda porównuje i zwraca true jeżeli wykryto sekwencję znaków, w przeciwnym wypadku zwraca false.

regex_match(napis, rx);

regex_match(napis, wynik, rx);

Przykład użycia

```
regex rx("\\w* jest \\w*");
string s1("Trawa jest zielona");
string s2("Niebo jest niebieskie");
string s3("Darth Vader jest człowiekiem");
                                        //Wynik
regex_match(s1, rx);                    //True
regex_match(s2, rx);                    //True
regex_match(s3, rx);                    //False
```

std::regex_match

```
regex rx("[[:digit:]]* to [[:lower:]]*");
string s1("5123 to liczba");
string s2("Czy 5123 to liczba?");
                                        //Wynik
regex_match(s1, rx);                    //True
regex_match(s2, rx);                    //False
```

regex_search

Metoda porównuje i zwraca true jeżeli wykryto sekwencję znaków, w przeciwnym wypadku zwraca false. W przeciwieństwie do regex_match sekwencja nie musi być pełna.

```
regex_search(napis, rx);
```

```
regex_search(napis, wynik, rx);
```

Przykład użycia

std::regex_search

```
regex rx1("a+b");
regex rx2("k.+n");
regex rx3("[[:space:]]");
regex rx4("[[:digit:]]");
string s("Ala ma kabanosa");

                                        //Wyniki
regex_search(s, rx1);                   // 1
regex_search(s, rx2);                   // 1
regex_search(s, rx3);                   // 1
regex_search(s, rx4);                   // 0
```

regex_replace

Metoda zamienia wykrytą sekwencję znaków na podaną w argumentach funkcji.

regex_replace(napis, rx, zmiana);

Przykład użycia

std::regex_replace

```
regex rx("zlego");
string s("Imperium zrobilo duzo zlego w galaktyce");
string element_for_replacement("dobrego");
cout << s << endl;
cout << regex_replace(s, rx, element_for_replacement) << endl;
```

```
Imperium zrobilo duzo zlego w galaktyce
Imperium zrobilo duzo dobrego w galaktyce
Press any key to continue . . .
```

regex_iterator

- Iterator umożliwiający przemieszczanie się po tych samych dopasowaniach wzorca w ciągu znakowym.
- Tworzy wektor dopasowań, umożliwiająca dostęp do wszystkich wystąpień sekwencji w podanym ciągu.
- Pierwszy obiekt może zostać odczytany przy tworzeniu.
- Po skonstruowaniu wskazuje na ostatnie dopasowanie w ciągu.

sregex_iterator(napis.begin(), napis.end(), rx); -specjalizacja dla typu string

Dziękujemy za Uwagę!

Źródła:

www.cplusplus.com/reference/regex/
cplusplus.com/reference/regex/ECMAScript/
en.cppreference.com/w/cpp/regex
pl.wikipedia.org/wiki/ECMAScript
Wykłady z przedmiotu PK3- S.Deorowicz