

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»**

**ПРОГРАММА ДЛЯ НАГЛЯДНОГО ОТОБРАЖЕНИЯ
СИНТАКСИСА АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ**

Автор работы:

Студент группы КЭ-406

АКИНБУЛИ О.Э.

Научный руководитель:

Доцент кафедры ЭВМ

Парасич В.А.

АКТУАЛЬНОСТЬ

Наглядное отображения синтаксиса алгоритмических языков особенно актуально при проведении занятий по программированию. Алгоритмические языки, такие как Java, Python или C++, имеют определенные правила и синтаксические структуры, которым необходимо следовать, чтобы код был корректным и удобочитаемым. Программа, которая визуально отображает синтаксис, может помочь обучающимся более эффективно понимать эти правила и придерживаться их.

ВВЕДЕНИЕ

Алгоритмические языки требуют точного и систематизированного способа представления своего синтаксиса. Синтаксис языка определяет правила и структуру для написания допустимых инструкций и выражений на этом языке. Он служит основой для построения алгоритмов и программ. Алгоритмические языки требуют точного и систематизированного способа представления своего синтаксиса. Синтаксис языка определяет правила и структуру для написания допустимых инструкций и выражений на этом языке. Он служит основой для построения алгоритмов и программ.

Представление синтаксиса алгоритмических языков является важнейшим шагом в языковом проектировании и программировании. Создавая формальную грамматику или систему обозначений, программисты могут гарантировать, что программы придерживаются правил языка, обеспечивая точную интерпретацию и выполнение компиляторами или интерпретаторами кода.

ВАЖНОСТЬ СИНТАКСИЧЕСКОГО ПРЕДСТАВЛЕНИЯ В АЛГОРИТМИЧЕСКИХ ЯЗЫКАХ

Представление синтаксиса в алгоритмических языках имеет первостепенное значение по нескольким причинам.

1. **Правильная интерпретация.** Синтаксис обеспечивает правила и структуру, которые определяют правильную форму программ на языке. Придерживаясь правил синтаксиса, программисты гарантируют, что их код поддается интерпретации и может быть выполнен корректно. Синтаксическое представление помогает обнаруживать и помечать любые нарушения или ошибки в коде, позволяя разработчикам выявлять и исправлять их.
2. **Согласованность и стандартизация.** Синтаксическое представление устанавливает стандарт для написания программ на том или ином языке. Это гарантирует, что все программы, написанные на этом языке, имеют согласованную и единообразную структуру, что облегчает программистам чтение, понимание и поддержку кода, написанного другими пользователями. Согласованный синтаксис также облегчает совместную работу и повторное использование кода в командах разработчиков или более широком сообществе программистов.
3. **Ясность и удобочитаемость.** Четко определенный синтаксис делает код более читабельным и понятным. Синтаксическое представление определяет правила организации инструкций, выражений и программных структур, облегчая понимание потока и логики работы программы. Это позволяет программистам выражать свои намерения ясно и недвусмысленно, повышая читаемость кода и снижая вероятность неправильного толкования.

ВАЖНОСТЬ СИНТАКСИЧЕСКОГО ПРЕДСТАВЛЕНИЯ В АЛГОРИТМИЧЕСКИХ ЯЗЫКАХ (продолжение)

Представление синтаксиса в алгоритмических языках имеет первостепенное значение по нескольким причинам:

4. Реализация компилятора и интерпретатора: Синтаксическое представление имеет важное значение для разработки компиляторов и интерпретаторов, которые отвечают за перевод и выполнение кода, написанного на языке программирования. Реализации компилятора и интерпретатора полагаются на синтаксическое представление для разбора и анализа кода, выполнения проверки синтаксиса и генерации исполняемого кода или промежуточных представлений для выполнения.
5. Обнаружение ошибок и отчетность: Синтаксическое представление позволяет обнаруживать синтаксические ошибки и сообщать о них в процессе компиляции или интерпретации. Когда программа нарушает правила синтаксиса, например, использует неправильное ключевое слово или размещает символы в неправильном порядке, синтаксическое представление может идентифицировать эти ошибки и предоставлять значимые сообщения об ошибках, чтобы помочь программистам в их исправлении.
6. Эволюция языка и расширения: Синтаксическое представление играет решающую роль в эволюции и расширении языков программирования. Разработчики языка могут изменять или расширять синтаксическое представление, чтобы внедрять новые языковые функции, улучшать выразительность или устранять ограничения. Представление служит справочным материалом для разработчиков языка и предоставляет четкую дорожную карту для включения нового синтаксиса в компиляторы и интерпретаторы.

ПОДСВЕТКА СИНТАКСИСА

Подсветка синтаксиса - это функция, обычно встречающаяся в редакторах кода и интегрированных средах разработки (IDE). Она применяет цвета и форматирование к различным элементам кода на основе их синтаксиса или языковых правил.

Роль подсветки синтаксиса и ее преимущества.

1. Визуальная дифференциация. Подсветка синтаксиса визуально выделяет различные элементы кода, такие как ключевые слова, переменные, функции и комментарии. Каждому элементу присваивается определенный цвет или форматирование, что облегчает разработчикам визуальную идентификацию и различие между ними.

2. Удобочитаемость и понимание. Подсветка синтаксиса улучшает удобочитаемость и понимание кода. Использование цвета и форматирования помогает быстро выявлять структуру кода, выделять комментарии и литералы, сопоставлять открывающие и закрывающие скобки и т.д..

ПОДСВЕТКА СИНТАКСИСА (продолжение)

3. Навигация по коду. Подсветка синтаксиса помогает в навигации по коду. Выделение различных элементов позволяет разработчикам быстро находить определенные части кода и переходить к ним, таким как определения функций, ссылки на переменные или циклы.
4. Поддержка в зависимости от языка. Подсветка синтаксиса обеспечивает поддержку в зависимости от языка путем адаптации к синтаксическим правилам различных языков программирования. Это гарантирует, что код визуально представлен в соответствии с синтаксисом конкретного языка, что облегчает написание и понимание кода на этом языке.

Подсветка синтаксиса - это ценная функция, которая улучшает процесс кодирования за счет улучшения читаемости, помогает обнаруживать ошибки и поддерживает эффективную навигацию по коду. Это играет важную роль в оказании помощи разработчикам в эффективном написании, понимании и обслуживании кода.

ПРОГРАММА ПОДСВЕТКИ СИНТАКСИСА ДЛЯ C++, РЕАЛИЗОВАННАЯ НА САМОМ C++:

```
main.cpp | syntaxc++.cpp ⋮
106
107 - int main() {
108 -     std::string code = R"(
109         #include <iostream>
110
111         int main() {
112             int num = 5;
113             std::string message = "Hello, world!";
114
115             if (num > 0) {
116                 std::cout << message << std::endl;
117             } else {
                std::cout << "Number is negative." << std::endl;
            }

            return 0;
        }
    }
}
```

input

ПРОГРАММА ПОДСВЕТКИ СИНТАКСИСА ДЛЯ C++, РЕАЛИЗОВАННАЯ НА САМОМ C++:

```
main.cpp SYNTAXFILE.cpp
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <string>
5 #include <unordered_set>
6
7 std::unordered_set<std::string> keywords = {
8     "auto", "break", "case", "char", "class", "const", "continue", "default",
9     "delete", "do", "double", "else", "enum", "extern", "float", "for",
10    "goto", "if", "int", "long", "new", "operator", "private", "protected",
11    "public", "return", "short", "sizeof", "static", "struct", "switch",
12    "template", "this", "typedef", "union", "unsigned", "void", "volatile", "while"
};

std::unordered_set<std::string> datatypes = {
    "bool", "char", "int", "float", "double", "void"
};

std::unordered_set<char> symbols = {
    '(', ')', '{', '}', '[', ']', ';', ',', '+', '-', '*', '/', '%', '=', '<', '>',
    '!', '&', '|', '^', '~', '.', ':', '?'
};

std::unordered_set<char> operators = {
```

input

```
Enter the filename: SYNTAXFILE.cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <unordered_set>

std::unordered_set<std::string> keywords = {
    "auto", "break", "case", "char", "class", "const", "continue", "default",
    "delete", "do", "double", "else", "enum", "extern", "float", "for",
    "goto", "if", "int", "long", "new", "operator", "private", "protected",
    "public", "return", "short", "sizeof", "static", "struct", "switch",
    "template", "this", "typedef", "union", "unsigned", "void", "volatile", "while"
};

std::unordered_set<std::string> datatypes = {
    "bool", "char", "int", "float", "double", "void"
};

std::unordered_set<char> symbols = {
    '(', ')', '{', '}', '[', ']', ';', ',', '+', '-', '*', '/', '%', '=', '<', '>',
    '!', '&', '|', '^', '~', '.', ':', '?'
};

std::unordered_set<char> operators = {
```

ПРОГРАММА ПОДСВЕТКИ СИНТАКСИСА ДЛЯ C++, РЕАЛИЗОВАННАЯ НА САМОМ C++:

- В этой программе определена функция `highlightCppCode`, которая принимает строку кода C++ в качестве входных данных и возвращает выделенную версию кода. При этом используется `std::map`, называемый `keywordColors`, для хранения ключевых слов C++ в качестве ключей и соответствующих им цветовых кодов ANSI в качестве значений.
- Функция `highlightCppCode` выполняет итерацию по каждому символу входного кода. Проверяется, является ли символ буквенно-цифровым или символом подчеркивания, и если да, то создается текущее слово. Как только встречается не буквенно-цифровой символ, проверяется, является ли "текущее слово" ключевым словом, просматривая его на карте "Цвета ключевых слов". Если это ключевое слово, добавляется соответствующий цветовой экранирующий код перед словом и сбрасывается значение `currentWord`. В противном случае добавляется слово как оно есть.
- В функции `main` задается пример фрагмента кода на C++ и передается функции `highlightCppCode`. Выделенный код сохраняется в переменной `highlightedCode` и затем выводится на консоль.

ПРОГРАММА НА PYTHON, КОТОРАЯ ПОКАЗЫВАЕТ ПОДСВЕТКУ СИНТАКСИСА НА PYTHON

```
main.py | syntaxpython.py ⋮
9      print(highlighted_code)
10
11     # Example code to be highlighted
12     code = '''
13     # This is a comment
14     def greet(name):
15         print("Hello, " + name + "!") # This is a string
16
17     greet("Alice") # This is a function call
18     '''
19
20     # Highlight the code and print the output
21     highlight_code(code)
22
```

input

```
# This is a comment
def greet(name):
    print("Hello, " + name + "!") # This is a string

greet("Alice") # This is a function call
```

Keywords:

- False
- None
- True
- and
- as
- assert
- async
- await

ПРОГРАММА НА PYTHON, КОТОРАЯ ПОКАЗЫВАЕТ ПОДСВЕТКУ СИНТАКСИСА НА PYTHON

- В приведенном выше коде определена функция `highlight_code()`, которая принимает строку кода Python в качестве входных данных и использует пигменты для выделения кода с помощью лексера `PythonLexer` и форматировщика `TerminalFormatter`. В конце выводится выделенный код.
- При желании можно загрузить код из файла. Выделенный код будет отображен в выходных данных терминала/консоли.
- Приведенный код использует `TerminalFormatter` для простоты. Однако можно выбрать другие форматировщики, предоставляемые библиотекой `Pgments` для генерации выделенного кода в различных выходных форматах, таких как HTML, LaTeX или файлы изображений.

ЗАКЛЮЧЕНИЕ

1. Был проведен обзор аналогов разрабатываемого приложения.
2. Был проведен обзор средств реализации и отобраны наиболее подходящие варианты.
3. Было разработано программное обеспечение для выделения синтаксиса текста программ на алгоритмических языках Python и C++.
4. Программное обеспечение было протестировано.

СПАСИБО ЗА ВНИМАНИЕ
АКИНБУЛИ О .Э