

ІНТЕЛЕКТУАЛЬНІ ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО КОДУ

Шпінталь М.Я.¹⁾, Паславський А.Ю.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Постановка проблеми

Написання програмного коду або скриптів є клопітким та рутинним процесом. Тому з метою реалізації деякого алгоритму певною мовою програмування, насамперед потрібен зручний і практичний редактор коду. Багато редакторів коду надають функції, які є особливо корисні для програмістів. Підсвічування тексту, автоматичні відступи та підкреслення помилок – це є найважливіші з інструментів такого роду. Вони дозволяють з одного погляду зрозуміти, що введений код в загальних рисах коректний, що всі блоки правильно вкладені один в один і не містять очевидних помилок. Кодування є важливою частиною програмування, поряд з такими етапами, як аналіз, проектування, компіляція, тестування, налагодження і супровід [1].

ІІ. Мета роботи

Метою роботи є вивчення алгоритмів та методів сучасних популярних редакторів коду та створення, зручного користувацького редактора коду, який буде надавати користувачу зручні інструменти для роботи.

Завдання:

- Виявити, та дослідити можливості популярних редакторів коду та алгоритми їх реалізації.
- Проаналізувати алгоритмічні засоби, редакторів коду, для виявлення найбільш ефективних за функціональним вимогам.
- Розробити зручний та багатфункціональний редактор коду.

ІІІ. Структура текстового редактора

Для реалізації поставленої задачі запропоновано розподілити програмний засіб на наступні модулі: AdvTextBox, AssemblyInfo, Block, BlockList, Document, Group, GroupList, Log, Parser.

У відповідних до модулів файлах реалізовано наступні функції: у файлі AdvTextBox.cs описаний клас AdvTextBox, об'єктом якого є поле для редагування тексту програм. У цьому класі переважана процедура обробки повідомлень.

Основна інформація про програму зберігається у файлі AssemblyInfo.cs. У файлі Block.cs описаний клас Block, який використовується для зберігання блоків тексту. Файл BlockList.cs містить опис масиву об'єктів класу Block. У файлі Document.cs реалізований пошук блоків тексту, в яких були зроблені зміни. Потім ці блоки передаються автоматом Parser для отримання остаточної розмальовки. У файлі Group.cs описаний клас Group, який служить для зберігання ключових слів. Клас Log, який описаний у файлі Log.cs, використовується для протоколювання дій автомата. Автомат Parser описаний у файлі Parser.cs. У ньому реалізована обробка рядків, в результаті якої текст змінює свій колір.

Висновок

Запропонований, алгоритм роботи, та сам текстовий редактор дозволяє ефективніше створювати програмні файли та швидше локалізувати помилки за рахунок структуризації та підсвічування окремих лексем програми.

Список використаних джерел

1. Компаниец Р. И. Системное программирование. Основы построения трансляторов: уч. пособие для высших учебных заведений / Р.И. Компаниец, Е.В. Маньков, Н.Е. Филатов. – СПб.: КОРОНА-Принт, 2000. – 256 с.