Instrukcja Użytkowania - Al Code Debugger

Metadata:

Autor: Sebastian Ciborowski

• **Data**: 15.07.2025

• **Wersja**: 1.0

Projekt: Al Code Debugger

• Typ dokumentu: Instrukcja użytkowania

Spis treści

Wprowadzenie do Al Code Debugger	. 1
Wymagania Systemowe	
Instalacja i Pierwsze Uruchomienie	
Interfejs Użytkownika	
Podstawowe Funkcje	
Analiza Kodu	
Praca z Wynikami	. 7
Najczęstsze Problemy	. 8
Wskazówki i Najlepsze Praktyki	9
D. Często Zadawane Pytania1	10

1. Wprowadzenie do Al Code Debugger

1.1 Czym jest Al Code Debugger

Al Code Debugger to zaawansowana aplikacja desktopowa przeznaczona do analizy kodu Python z wykorzystaniem sztucznej inteligencji. Aplikacja wykorzystuje model **GPT-4.1** do generowania inteligentnych sugestii naprawczych dla wykrytych błędów i problemów w kodzie.

1.2 Główne Funkcje

- \bigcirc Analiza kodu Python wykrywanie błędów składniowych i logicznych
- Sugestie AI inteligentne rekomendacje naprawcze od GPT-4.1
- Podświetlanie składni kolorowanie kodu dla lepszej czytelności
- Numeracja linii łatwe odnajdywanie błędów
- Historia analiz przechowywanie poprzednich rezultatów

1.3 Korzyści z Użytkowania

Korzyść	Opis
Szybsze debugowanie	Automatyczne wykrywanie problemów
✓ Nauka programowania	Wyjaśnienia i sugestie poprawek
✓ Oszczędność czasu	Redukcja czasu spędzonego na szukaniu błędów
Poprawa jakości kodu	Rekomendacje najlepszych praktyk

2. Wymagania Systemowe

2.1 Wymagania Minimalne

Komponent	Specyfikacja
System operacyjny	Windows 10 (wersja 1903 lub nowsza)
Procesor	Intel Core i3 lub AMD Ryzen 3
Pamięć RAM	4 GB
Miejsce na dysku	200 MB wolnej przestrzeni
Połączenie internetowe	Wymagane do komunikacji z API OpenAI

2.2 Wymagania Zalecane

Komponent	Specyfikacja
System operacyjny	Windows 11
Procesor	Intel Core i5 lub AMD Ryzen 5
Pamięć RAM	8 GB lub więcej
Miejsce na dysku	500 MB wolnej przestrzeni
Połączenie internetowe	Szerokopasmowe połączenie

2.3 Dodatkowe Informacje

Ważne informacje:

- Aplikacja nie wymaga uprawnień administratora
- Działa w trybie offline dla podstawowych funkcji (bez sugestii Al)
- Obsługuje pliki Python (.py) do 10 MB

3. Instalacja i Pierwsze Uruchomienie

3.1 Pobieranie Aplikacji

- 1. Pobierz plik AlCodeDebugger.exe z repozytorium GitHub
- 2. Zapisz plik w wybranej lokalizacji na dysku
- 3. Nie jest wymagana dodatkowa instalacja

3.2 Pierwsze Uruchomienie

3.2.1 Uruchomienie Aplikacji

- 1. Kliknij dwukrotnie na plik AlCodeDebugger.exe
- 2. Windows może wyświetlić ostrzeżenie kliknij "Więcej informacji" → "Uruchom mimo to"
- 3. Aplikacja uruchomi się i wyświetli główne okno

3.2.2 Pierwsze Wyświetlenie

Po uruchomieniu zobaczysz:

- 🖳 Główne okno aplikacji z podziałem na dwie części
- Panel edytora kodu po lewej stronie
- Panel wyników po prawej stronie
- Pasek narzędzi u góry z przyciskami funkcji

4. Interfejs Użytkownika

4.1 Główne Okno Aplikacji

Aplikacja składa się z następujących elementów:

4.2 Pasek Narzędzi

Dostępne przyciski:

Przycisk	Funkcja
🗁 Otwórz plik	Wczytuje plik Python do analizy
🔾 Analizuj	Rozpoczyna analizę aktualnego kodu
Wyczyść Wyczyść	Czyści edytor i wyniki

4.3 Panel Edytora Kodu

Lokalizacja: Lewa strona aplikacji

Funkcje:

- Edycja kodu Python
- Numeracja linii
- Podświetlanie składni

4.4 Panel Wyników

Lokalizacja: Prawa strona aplikacji

Zakładki dostępne:

- **Błędy** lista wykrytych problemów w kodzie
- Sugestie AI rekomendacje naprawcze od GPT-4.1
- Metryki podstawowe statystyki kodu
- 🗏 **Historia** poprzednie analizy

5. Podstawowe Funkcje

5.1 Otwieranie Plików

Jak otworzyć plik:

- 1. Kliknij przycisk " Otwórz plik"
- 2. Wybierz plik Python (.py) z dysku
- 3. Kod zostanie załadowany do edytora

Obsługiwane formaty:

- .py pliki Python
- Maksymalny rozmiar: 10 MB

5.2 Edycja Kodu

Podstawowe funkcje edytora:

- Ø Wpisywanie i edycja kodu Python
- Numeracja linii po lewej stronie

5.3 Czyszczenie Edytora

Jak wyczyścić edytor:

- 1. Kliknij przycisk " Wyczyść"
- 2. Edytor i panel wyników zostaną wyczyszczone
- 3. Możesz rozpocząć pracę z nowym kodem

6. Analiza Kodu

6.1 Rozpoczynanie Analizy

Kroki analizy:

- 1. 🗁 Wczytaj kod do edytora (plik lub wpisz ręcznie)
- 2. 🔍 Kliknij przycisk "Analizuj"
- 3. 🛮 Poczekaj na zakończenie procesu
- 4. 📋 Wyniki pojawią się w panelu po prawej stronie

6.2 Proces Analizy

Co dzieje się podczas analizy:

graph LR

A[Analiza składni] --> B[Wykrywanie błędów]

B --> C[W Komunikacja z Al]

C --> D[♀ Generowanie sugestii]

- 1. Analiza składni sprawdzenie poprawności kodu
- 2. Wykrywanie błędów identyfikacja problemów
- 3. Komunikacja z AI przesłanie zapytania do GPT-4.1
- 4. Generowanie sugestii otrzymanie rekomendacji naprawczych

7. Praca z Wynikami

7.1 Zakładka " Błędy"

Zawartość:

- Spiration in the property of the
- Numer linii gdzie występuje błąd
- Typ błędu (składniowy, logiczny)
- 📝 Opis problemu

Format wyświetlania:

ho Linia 15: SyntaxError - Missing closing parenthesis

☐ Linia 23: NameError - Variable 'x' is not defined

7.2 Zakładka " Sugestie AI"

Zawartość:

- Sugestie naprawcze wygenerowane przez GPT-4.1
- Wyjaśnienia problemów
- Przykłady poprawnego kodu

7.3 Pozostałe Zakładki

Metryki:

Podstawowe statystyki analizowanego kodu

Historia:

• Przechowuje poprzednie analizy kodu

8. Najczęstsze Problemy

8.1 Problemy z Uruchomieniem

Problem	Rozwiązanie
	- Sprawdź czy masz Windows 10/11 - Uruchom jako administrator (jeśli potrzeba) - Sprawdź czy plik nie jest zablokowany przez antywirus
⚠ Ostrzeżenie bezpieczeństwa Windows	- Kliknij "Więcej informacji" → "Uruchom mimo to" - Przyczyna: Aplikacja nie ma certyfikatu cyfrowego

8.2 Problemy z Analizą

Problem	Rozwiązanie
Ⅲ Analiza się nie rozpoczyna	- Sprawdź połączenie internetowe - Upewnij się że kod nie jest pusty - Sprawdź czy plik nie przekracza 10 MB
₩ Brak sugestii Al	- Przyczyna: Brak połączenia z internetem - Sprawdź połączenie i spróbuj ponownie

8.3 Problemy z Plikami

Problem	Rozwiązanie
	- Sprawdź czy plik ma rozszerzenie .py - Sprawdź uprawnienia do pliku - Sprawdź czy plik nie jest uszkodzony

9. Wskazówki i Najlepsze Praktyki

9.1 Optymalne Używanie Aplikacji

Przygotowanie kodu:

- Upewnij się że kod jest zapisany w kodowaniu UTF-8
- Usuń zbędne komentarze przed analizą
- Podziel bardzo długie pliki na mniejsze części

Analiza kodu:

- Analizuj kod częściami dla lepszych wyników
- Sprawdź sugestie Al zanim zastosujesz zmiany
- Zapisuj wyniki dla późniejszego porównania

9.2 Poprawa Wydajności

← Szybsza analiza:

- Úżywaj plików o rozmiarze do 1 MB dla najszybszych rezultatów
- 🕎 Zamknij inne aplikacje podczas analizy

9.3 Interpretacja Wyników

Jak czytać sugestie Al:

- Zawsze sprawdź logikę sugerowanych zmian
- Nie stosuj automatycznie wszystkich sugestii
- 🗏 Ucz się na otrzymanych wyjaśnieniach

10. Często Zadawane Pytania

10.1 Pytania Ogólne

- ? Czy aplikacja jest bezpłatna?
- ✓ Tak, aplikacja jest dostępna za darmo
- ? Czy potrzebuję klucza API OpenAI?
- ✓ Nie, klucze są wbudowane w aplikację
- ? Czy aplikacja działa offline?
- 🛕 Częściowo podstawowa analiza tak, sugestie Al wymagają internetu

10.2 Pytania Techniczne

- ? Jak dokładne są sugestie Al?
- GPT-4.1 ma wysoką dokładność (~85-95%) dla typowych problemów Python

- ? Czy mogę analizować większe projekty?
- ✓ Tak, ale analizuj pliki pojedynczo (maksymalnie 10 MB na plik)

10.3 Pytania o Bezpieczeństwo

- ? Czy aplikacja jest bezpieczna?
- ✓ Tak, aplikacja nie zbiera danych osobowych ani nie instaluje dodatkowego oprogramowania
- ? Gdzie są przechowywane wyniki analiz?
- Lokalnie na Twoim komputerze w bazie danych aplikacji

10.4 Wsparcie i Pomoc

- Gdzie mogę zgłosić błąd?
- ✓ Na stronie GitHub projektu w sekcji Issues
- ¶ Jak mogę zaproponować nową funkcję?
- ✓ Przez sekcję Issues na GitHub lub bezpośredni kontakt z autorem
- ? Czy planowane są aktualizacje?
- ✓ Tak, regularnie dodawane są nowe funkcje i poprawki

& Informacje Kontaktowe

Autor: Sebastian Ciborowski

Projekt: Al Code Debugger

Wersja dokumentacji: 1.0

Data aktualizacji: 15.07.2025

GitHub: github.com/sebastian-c87

Podziękowania

Dziękujemy za korzystanie z Al Code Debugger! Mamy nadzieję, że aplikacja pomoże Ci w codziennej pracy z kodem Python.

Wskazówka: Regularne korzystanie z aplikacji pomoże Ci nie tylko znajdować błędy, ale także uczyć się lepszych praktyk programowania w Python!