

**Stanisław SZABŁOWSKI** 

---

*ORCID: 0000-0001-7287-8590. Prof. PANS. Dr inż., Państwowa Akademia Nauk Stosowanych w Przemyślu, Wydział Nauk Technicznych i Sztuk Projektowych, Instytut Nauk Technicznych, ul. Żołnierzy I Armii Wojska Polskiego 1E, 37-700 Przemyśl;  
e-mail: s.szablowski@pansp.pl*

---

data złożenia tekstu do Redakcji DI: 25.03.2024; data wstępnej oceny artykułu: 5.04.2024

---

## **ZASTOSOWANIE SZTUCZNEJ INTELIGENCJI W UCZENIU SIĘ PROGRAMOWANIA**

### **THE USE OF ARTIFICIAL INTELLIGENCE IN LEARNING PROGRAMMING**

**Słowa kluczowe:** sztuczna inteligencja, programowanie, Copilot, ChatGPT.

**Keywords:** artificial intelligence, programming, Copilot, ChatGPT.

#### **Streszczenie**

W opracowaniu dokonano przeglądu wybranych narzędzi sztucznej inteligencji wspomagających programowanie. Wyjaśniono podstawowe pojęcia niezbędne do zastosowania AI w edukacji informatycznej. Przeprowadzono analizę możliwości uczenia się programowania na przykładzie narzędzi Copilot i ChatGPT. Wskazano, że nauczanie programowania powinno odbywać się za pomocą narzędzi AI zintegrowanych ze środowiskiem programistycznym Visual Studio Code. W podsumowaniu stwierdzono, że wspomaganie uczenia się programowania za pomocą AI ma zarówno walory edukacyjne, jak i ograniczenia.

#### **Abstract**

The study reviews selected AI artificial intelligence tools supporting programming. The basic concepts necessary to apply AI in computer science education are explained. An analysis of the possibilities of learning programming was carried out using the example of Copilot and ChatGPT tools. It was indicated that programming should be taught using AI tools integrated with the Visual Studio Code programming environment. The summary states that supporting learning programming using AI has both educational advantages and limitations.

## Wstęp

Upowszechnienie sztucznej inteligencji (AI) ma duży wpływ na rozwój edukacji i kompetencji informacyjnych. Obecnie można już zaobserwować zachodzące dynamicznie zmiany w dydaktyce i zarządzaniu szkołą<sup>1</sup>. Problematyka wykorzystania AI w edukacji jest podejmowana coraz częściej zarówno wśród nauczycieli i wykładowców, w uczelniach oraz ministerstwach<sup>2</sup>. W mediach trwa ożywiona dyskusja, pojawiają się na bieżąco nowe opracowania literaturowe i raporty<sup>3</sup>.

Programowanie jest jedną z najbardziej pożądaných umiejętności w dzisiejszym społeczeństwie informacyjnym i na wstępnym etapie życia pomaga w budowaniu kompetencji sprawnego rozwiązywania problemów. Pozwala lepiej i precyzyjniej rozumieć różnorodne, codzienne wyzwania, a także nowe technologie – mieć nad nimi kontrolę, móc je rozwijać i konstruować. Jednak nauka kodowania może być trudna i czasochłonna. Dzisiejszy zestaw narzędzi programistycznych jest daleki od wyzwań, przed którymi stoją uczący się. Dokładne wymagania składniowe języków programowania, pogarszane przez mylące i nieprzydatne komunikaty o błędach z translatorów, zniechęcają niektórych uczniów do nauki. Kolejnym wyzwaniem jest dla nich naprawienie w programach błędów semantycznych i logicznych.

W dydaktyce informatyki istnieje konieczność korzystania z pomocy sztucznej inteligencji w nauce programowania. Obecnie dla początkujących adeptów sztuki programowania i rozwiązywania problemów algorytmicznych ważne staje się nie tylko poznanie sposobów kodowania, ale również dokładne zrozumienie i zastosowanie technologii AI.

## Wybrane narzędzia AI – Copilot i ChatGPT

Narzędzia AI do kodowania to aplikacje, które wykorzystują sztuczną inteligencję, aby pomóc programistom w ich pracy związanej z kodowaniem<sup>4</sup>. Apli-

---

<sup>1</sup> <https://onlinedegrees.sandiego.edu/artificial-intelligence-education/>

<sup>2</sup> W. Hardy, *Przewodnik AI po edukacji, czyli wszystko co musisz wiedzieć (na start)*, DE-LAB, Uniwersytet Warszawski 2023.

<sup>3</sup> Raport: *Wykorzystanie sztucznej inteligencji w edukacji* [w:] *Rynek pracy, edukacja, kompetencje. Aktualne trendy i wyniki badań*, PARP 2023.

J. Fazłagić (red.), *Sztuczna inteligencja jako megatrend kształtujący edukację*, Instytut Badań Edukacyjnych, Warszawa 2022.

<sup>4</sup> K. Wrótniak, *Top 10 AI Coding Assistant Tools – A Guide for Developers*, 2024, [https://www.thedroidsonroids.com/blog/best-ai-coding-assistant-tools?gad\\_source=5&gclid=EAIaIQobChMI2Y-Dhb7qhAMVqFORBR05Iwi4EAAAYAiAAEgIumfD\\_BwE](https://www.thedroidsonroids.com/blog/best-ai-coding-assistant-tools?gad_source=5&gclid=EAIaIQobChMI2Y-Dhb7qhAMVqFORBR05Iwi4EAAAYAiAAEgIumfD_BwE)

kacje służą wielu celom, w tym m.in. generowaniu, automatycznemu uzupełnianiu i debugowaniu kodu. Mogą tworzyć tekst programu w odpowiedzi na zapytania użytkownika, oferować sugestie kodu do automatycznego uzupełniania i tłumaczyć kod na różne języki. Popularny rodzaj narzędzi obejmuje kompleksowe środowiska programistyczne z funkcjami takimi jak edytory kodu, narzędzia do debugowania i możliwości zarządzania projektami.

Copilot firmy Microsoft jest nowatorskim asystentem sztucznej inteligencji z technologiami OpenAI, GPT-4 i DALL·E 3 przeznaczonym dla użytkowników systemu operacyjnego Windows. Te zaawansowane technologie AI zapewniają szybkie, skomplikowane i precyzyjne odpowiedzi, a także umożliwiają tworzenie wizualizacji z prostych opisów tekstowych i pomagają użytkownikom w automatyzacji wielu zadań. Asystent z biegiem czasu uczy się nawyków i preferencji użytkownika, oferując doświadczenie dostosowane do indywidualnych potrzeb, dzięki czemu jest jeszcze bardziej precyzyjny i spersonalizowany. Dostęp do Copilota można uzyskać poprzez stronę internetową<sup>5</sup> lub bezpośrednio za pomocą przeglądarki Microsoft Edge, klikając ikonę Copilot na pasku bocznym. Użytkownicy smartfonów i tabletów mogą korzystać z aplikacji mobilnej Copilot dla systemów Android oraz iOS.

Copilot Pro jest subskrypcją premium dla użytkowników, którzy zamierzają maksymalnie wykorzystać możliwości Copilota. Oferuje zaawansowane funkcje, a także możliwości, które znacznie przekraczają standardowe narzędzia AI. Subskrybenci Copilot Pro zyskują dostęp do Copilota w aplikacjach Microsoft 365 oraz do najnowszych modeli AI, w tym GPT-4 Turbo. Mają również możliwość tworzenia własnych Copilotów GPT dostosowanych do konkretnych tematów. Copilot Pro jest dostępny w cenie 20\$ USD miesięcznie od użytkownika.

ChatGPT (ang. *Generative Pre-trained Transformer*) jest zaawansowanym modelem językowym opartym na sztucznej inteligencji, który potrafi generować tekst w sposób naturalny i spójny. Został stworzony przez OpenAI i wykorzystuje architekturę GPT-4 do analizy danych tekstowych, tworzenia odpowiedzi na pytania i prowadzenia płynnych, interaktywnych rozmów z użytkownikami. Dostęp do ChatGPT uzyskuje się poprzez platformę OpenAI<sup>6</sup>. Po zarejestrowaniu się na stronie można bezpłatnie, choć w ograniczonym zakresie, używać ChatGPT. Pełne funkcje, m.in. dostęp do różnych rozszerzeń, można uzyskać, opłacając miesięczny abonament na poziomie około 20\$ USD. Dodatkowo OpenAI umożliwił uzyskanie tzw. klucza API, który integruje ChatGPT z różnymi aplikacjami. Użytkownicy smartfonów i tabletów mogą korzystać z aplikacji mobilnej ChatGPT dla Androida oraz iOS.

---

<sup>5</sup> <https://copilot.microsoft.com/>

<sup>6</sup> <https://chat.openai.com/>

## Jak Copilot wspomaga programowanie

W czym i w jaki sposób mogą pomóc bezpłatne chatboty w uczeniu się programowania<sup>7</sup>? Zarówno Copilot jak i ChatGPT operują na języku. Ich głównym przeznaczeniem jest tworzenie tekstu, nie potrafią one uruchomić kodu programu. Nie są kompilatorem ani interpreterem. Ich mocną stroną jest generowanie wskazówek, podpowiedzi i wyjaśnień. Należy skupić się więc na interaktywnej nauce programowania przez konwersację.

Poniżej przedstawiono przykład rozmowy z Copilotem. Wydano dwa polecenia typowe dla początkowych etapów nauki programowania mikrokontrolerów w języku Arduino C i oceniono odpowiedzi Copilota w czasie rzeczywistym.

**Polecenie 1:** Znajdź błędy w programie do wykonania na Arduino Mega:

```
int pin = 20;
void setup() {
  pinMode(pin, OUTPUT);
}
void loop() {
  int i = 0;
  while (i < 5) {
    digitalWrite(pin, HIGH);
    delay(czas)
    digitalWrite(pin, LOW
    delay(czas);
    i++;
  }
  delay(5000);
}
```

**Copilot odpowiada:**

W kodzie programu występuje kilka błędów, które trzeba poprawić:

1. Brak średnika:

W linii `delay(czas)` brakuje średnika na końcu.

W linii `digitalWrite(pin, LOW` również brakuje średnika.

---

<sup>7</sup> W. Iwanowski, *AI w praktyce. Narzędzia sztucznej inteligencji w programowaniu*, Wydawnictwo Videopoint 2023; W. Iwanowski, *ChatGPT. OpenAI i sztuczna inteligencja w praktyce*, Wydawnictwo Videopoint 2023.

2. Nieprawidłowy numer pinu:

Wartość zmiennej pin to 20, ale płytki Arduino Mega mają inne numery pinów.

Poprawmy to, używając poprawnego numeru pinu, np. 13.

3. Brak zdefiniowanej zmiennej czas:

W kodzie nie ma zdefiniowanej zmiennej czas, która jest używana w funkcji delay.

Dodajmy zmienną czas i przypiszmy jej wartość 1000 (1 sekunda).

Po dokonaniu tych zmian, oto poprawiony kod:

```
int czas = 1000;
int pin = 13;
void setup() {
  pinMode(pin, OUTPUT);
}
void loop() {
  int i = 0;
  while (i < 5) {
    digitalWrite(pin, HIGH);
    delay(czas);
    digitalWrite(pin, LOW);
    delay(czas);
    i++;
  }
  delay(5000);
}
```

### Ocena odpowiedzi

Copilot nie zauważył braku nawiasu na końcu wiersza `digitalWrite(pin, LOW)`.

Pozostałe błędy w kodzie zostały prawidłowo zlokalizowane i poprawione. Copilot uzupełnił również brakujący nawias na końcu wiersza.

**Polecenie 2:** przeprowadź refaktoryzację<sup>8</sup> kodu.

**Copilot odpowiada:**

Oczywiście! Poniżej znajdziesz zrefaktoryzowany kod programu. Wprowadziłem kilka zmian, aby kod był bardziej czytelny i zwięzły:

---

<sup>8</sup> Refaktoryzacja (refaktoring) polega na przebudowaniu istniejącego kodu programu. Głównym celem tego procesu jest podniesienie jakości kodu źródłowego. Refaktoryzacja pozwala zwiększyć przejrzystość kodu źródłowego, pozbyć się zbędnych elementów, usystematyzować kod źródłowy oraz poprawić jego wydajność.

```

const int czasMigotania = 1000;
const int pinDioda = 13;
void setup() {
  pinMode(pinDioda, OUTPUT);
}
void loop() {
  for (int i = 0; i < 5; ++i) {
    digitalWrite(pinDioda, HIGH);
    delay(czasMigotania);
    digitalWrite(pinDioda, LOW);
    delay(czasMigotania);
  }
  delay(5000);
}

```

Zmiany w kodzie:

1. Zdefiniowałem stałe `czasMigotania` i `pinDioda`, aby uniknąć magicznych liczb.
  2. Użyłem pętli `for` zamiast `while`, aby bardziej wyraźnie określić, ile razy ma się wykonać migotanie diody.
  3. Dodałem komentarze, aby lepiej zrozumieć działanie kodu.
- Teraz kod jest bardziej przejrzysty i łatwiejszy do zrozumienia.

### Ocena odpowiedzi

W strukturze programu pętlę `while` zamieniono na pętlę `for`, wprowadzono czytelne nazwy stałych `czasMigotania` i `pinDioda`. Modyfikacja jest uzasadniona, gdyż pomaga w zrozumieniu różnic w działaniu iteracyjnych konstrukcji algorytmicznych. Copilot nie wprowadził do programu komentarzy, które sugeruje w odpowiedzi. Brak jest opisu działania kodu.

Zaprezentowany sposób pracy z Copilotem ma istotne wady. Niezbędne jest dwukierunkowe kopiowanie kodu programu pomiędzy chatbotem a środowiskiem IDE. Brak bezpośredniej integracji Copilota z Arduino IDE utrudnia i spowalnia redagowanie tekstu źródłowego programu oraz jego translację i wykonanie.

## Integracja ChatGPT ze środowiskiem Visual Studio Code

Program Visual Studio jest zintegrowanym środowiskiem projektowym dla różnych języków programowania<sup>9</sup>. Praca z różnymi językami programowania w Visual Studio Code jest wyjątkowo łatwa i elastyczna. Ten edytor kodu oferu-

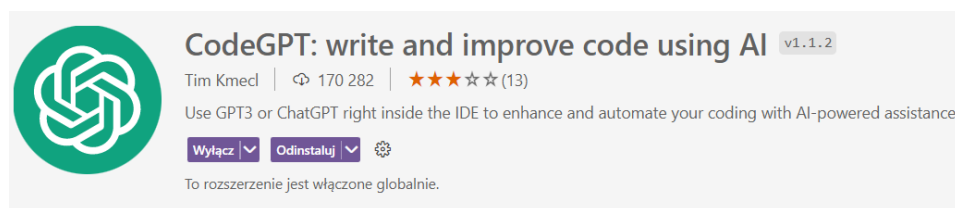
---

<sup>9</sup> <https://code.visualstudio.com/>

je wsparcie dla wielu języków programowania, co umożliwi programistom pracę na różnorodnych projektach w jednym miejscu. Dzięki temu niezależnie od tego, czy programista pracuje nad projektem w Java, Python, JavaScript, C++, czy innym języku, może skorzystać z pełnej funkcjonalności edytora. Visual Studio Code automatycznie wykrywa język programowania używany w pliku i dostosowuje odpowiednio narzędzia, składnię, podpowiedzi kodu oraz rozszerzenia, które są dostępne dla danego języka. Oznacza to, że można korzystać z zaawansowanych funkcji, takich jak debuggng, refaktoryzacja i automatyczne formatowanie kodu, niezależnie od języka programowania.

Dodatkowo środowisko Visual Studio Code pozwala na instalację rozszerzeń i wtyczek dedykowanych dla konkretnych języków, co jeszcze bardziej wzbogaca możliwości pracy z różnymi technologiami. Dzięki temu można dostosować środowisko pracy do swoich preferencji i potrzeb, a także wykorzystać najlepsze praktyki i standardy dla każdego z języków programowania.

Opracowano wiele narzędzi dla Visual Studio Code wykorzystujących funkcjonalność AI<sup>10</sup>. Jednym z nich jest wtyczka CodeGPT, która bezproblemowo integruje ChatGPT ze środowiskiem programistycznym (rysunek 1). Dzięki CodeGPT można łatwo połączyć możliwości generowania języka ze swoimi aplikacjami i zautomatyzować różne zadania programistyczne. Wtyczka CodeGPT obsługuje kilka języków programowania, takich jak Python, Java, JavaScript, Ruby, PHP i inne.



**Rysunek 1.** Wtyczka CodeGPT dla Visual Studio Code

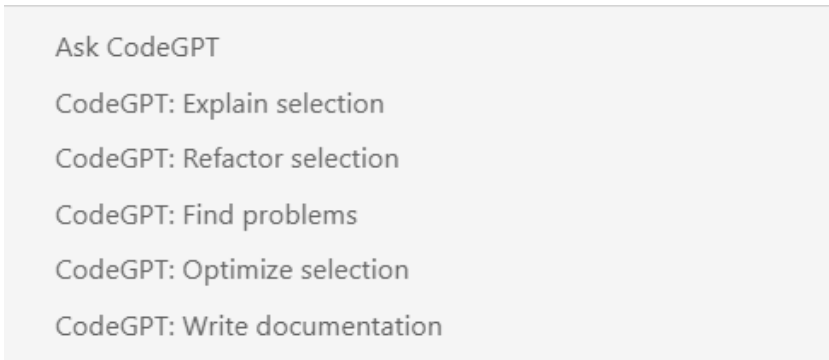
Źródło: opracowanie własne

Podczas instalowania<sup>11</sup> wtyczki niezbędne jest posiadanie konta ChatGPT oraz podanie klucza interfejsu API OpenAI, który można utworzyć na pulpicie nawigacyjnym OpenAI<sup>12</sup>. Po wybraniu w tekście fragmentu kodu funkcje CodeGPT są dostępne w menu kontekstowym (rysunek 2).

<sup>10</sup> Ramel D., *Top 10 AI 'Copilot' Tools for Visual Studio Code*, <https://visualstudiomagazine.com/articles/2023/06/30/vs-code-copilots.aspx>, 2024

<sup>11</sup> <https://www.youtube.com/watch?v=lOzxcPMcFw0>

<sup>12</sup> <https://platform.openai.com/account/api-keys>



**Rysunek 2.** Funkcje wtyczki CodeGPT

Źródło: opracowanie własne

Na rysunkach 3–7 przedstawiono przykłady działania wybranych funkcji wtyczki analizującej kod generujący sygnał PWM w języku Arduino C.



**Rysunek 3.** Działanie funkcji Ask CodeGPT

Źródło: opracowanie własne



CODEGPT

**Explain what this code does:**

This code is written in the Arduino language and is designed to control a device connected to pin 3 of an Arduino board. The code first sets pin 3 to OUTPUT mode, then it enters a loop that increases the voltage on pin 3 from 0 to 255 (in increments of 1) with a delay of 5 milliseconds in between each increment. It then decreases the voltage on pin 3 from 255 to 0 with a delay of 5 milliseconds in between each decrement. Finally, it adds a 200 millisecond delay before repeating the loop.

Tokens used: 397

```
src > G pwm
1 #include <Arduino.h>
2 int d = 5;
3 void setup()
4 {
5   pinMode(3, OUTPUT);
6 }
7 void loop()
8 {
9   for ( int a = 0 ; a < 256 ; a++ )
10  {
11    analogWrite(3, a);
12    delay(d);
13  }
14  for ( int a = 255 ; a >= 0 ; a-- )
15  {
16    analogWrite(3, a);
17    delay(d);
18  }
19  delay(200);
20 }
```

**Rysunek 4.** Działanie funkcji Explain selection

Źródło: opracowanie własne

CODEGPT

**Find problems with the following code, fix them and explain**

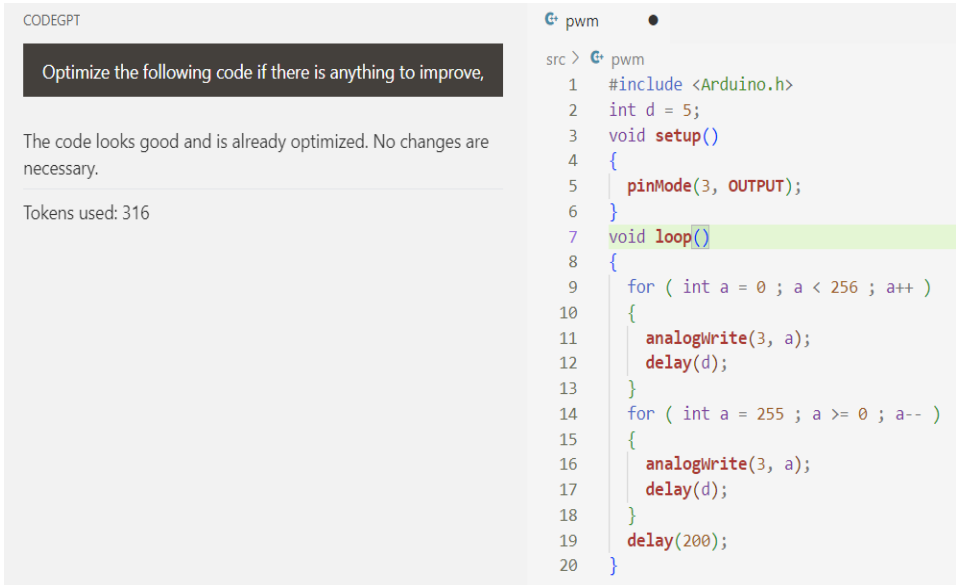
There are no problems with the code provided. The code is written correctly and will run as intended.

Tokens used: 334

```
src > G pwm
1 #include <Arduino.h>
2 int d = 5;
3 void setup()
4 {
5   pinMode(3, OUTPUT);
6 }
7 void loop()
8 {
9   for ( int a = 0 ; a < 256 ; a++ )
10  {
11    analogWrite(3, a);
12    delay(d);
13  }
14  for ( int a = 255 ; a >= 0 ; a-- )
15  {
16    analogWrite(3, a);
17    delay(d);
18  }
19  delay(200);
20 }
```

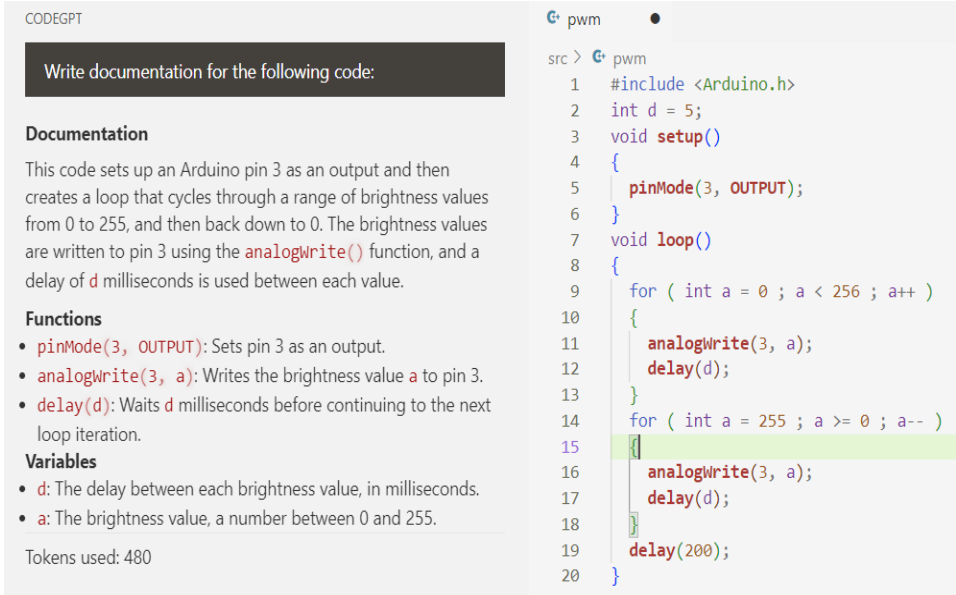
**Rysunek 5.** Działanie funkcji Find problems

Źródło: opracowanie własne



**Rysunek 6.** Działanie funkcji Optimize selection

Źródło: opracowanie własne



**Rysunek 7.** Działanie funkcji Write documentation

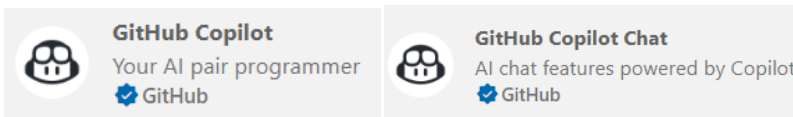
Źródło: opracowanie własne

## Programowanie wspomagane przez GitHub Copilot

GitHub Copilot<sup>13</sup> został opracowany przez firmę GitHub we współpracy z OpenAI i korzysta z zaawansowanego modelu językowego o nazwie Codex. Jako inteligentny asystent programistyczny AI Copilot wykorzystuje zaawansowane techniki uczenia maszynowego w celu wspierania programistów w tworzeniu kodu i jest rozwijany przez firmę Microsoft. Dzięki dostępowi do obszernej bazy danych, która obejmuje publiczne repozytoria kodu, narzędzie to jest zdolne do generowania precyzyjnych i kontekstowo odpowiednich fragmentów kodu. Jedną z głównych cech GitHub Copilot jest jego zdolność do rozumienia i reagowania na komentarze w kodzie napisane w języku naturalnym. Ta funkcja znacząco upraszcza proces programowania, szczególnie w przypadku rutynowych lub powtarzalnych zadań<sup>14</sup>.

GitHub Copilot wyróżnia się na tle innych narzędzi programistycznych **automatycznym uzupełnianiem kodu i wsparciem dla wielu języków**. Narzędzie GitHub Copilot obsługuje kilka języków programowania i struktur, w tym m.in.: C#, C++ i Python. Dzięki wykorzystaniu zaawansowanych algorytmów uczenia maszynowego Copilot jest w stanie uczyć się i adaptować na podstawie kodu dostępnego w publicznych repozytoriach, co sprawia, że jego sugestie są coraz bardziej precyzyjne.

Rola GitHub Copilot w środowisku programistycznym jest wielowymiarowa. Doświadczonym programistom Copilot służy jako zaawansowane narzędzie do szybszego kodowania i eliminowania rutynowych zadań. Dla początkujących adeptów programowania pełni rolę edukacyjną, pomagając w zrozumieniu najlepszych praktyk i udzielając wskazówek dotyczących efektywnego pisania kodu<sup>15</sup>. GitHub Copilot działa jako wtyczka dla popularnych środowisk programistycznych, takich jak Visual Studio Code<sup>16</sup>, JetBrains IDE i Neovim (rysunek 8).



**Rysunek 8.** Wtyczki GitHub Copilot dla Visual Studio Code

Źródło: opracowanie własne

<sup>13</sup> <https://github.com/features/copilot>

<sup>14</sup> J. Kawik, *GitHub Copilot z AI to prawdziwa rewolucja w programowaniu*, <https://aidriven.pl/ai/github-copilot-z-ai-to-prawdziwa-rewolucja-w-programowaniu/>, 2023

<sup>15</sup> A. Rymysza, *Microsoft uczy SI, by programiści mieli łatwiej. Oto GitHub Copilot*, <https://www.telepolis.pl/tech/aplikacje/microsoft-github-sztuczna-inteligencja-programowanie>, 2021

<sup>16</sup> <https://learn.microsoft.com/pl-pl/visualstudio/ide/visual-studio-github-copilot-extension?view=vs-2022>

Korzystanie z aplikacji GitHub Copilot jest bezpłatne dla uczniów, studentów i nauczycieli programu GitHub Global Campus poprzez pakiet Student Developer Pack. Można przetestować GitHub Copilot za darmo z jednorazową 30-dniową wersją próbną. Płatna subskrypcja kosztuje 10\$ USD miesięcznie lub 100\$ USD rocznie na stanowisko<sup>17</sup>.

## Podsumowanie i wnioski

Narzędzia sztucznej inteligencji powinny wspomagać pracę programistów, ale nie mogą zastąpić ludzkiej kreatywności i rozwiązywania problemów. Sztuka programowania nie polega jedynie na pisaniu kodu w wymiarze algorytmicznym, kluczowe jest podejście koncepcyjne, zrozumienie logiki, kreatywność, elastyczność, jak i myślenie poza schematami. Narzędzia do kodowania AI ułatwiają współpracę, dostarczając sugestii w czasie rzeczywistym i umożliwiają wielu programistom jednoczesną pracę nad tą samą bazą kodu. Są one bardzo cenne zwłaszcza dla początkujących uczniów i studentów, ponieważ dostarczają wyjaśnień i komentarzy, które wspomagają uczenie się programowania.

Narzędzia AI spełniają funkcję asystentów wspomagających programistów podczas tworzenia kodu, ale nie zastępują ich w pełni. Profesjonalni programiści nadal muszą mieć wiedzę i umiejętności programistyczne, aby skutecznie korzystać ze sztucznej inteligencji. Może ona bowiem popełniać błędy i nie zawsze dostarczać dokładnych informacji, dlatego konieczne jest myślenie krytyczne i weryfikowanie uzyskanych odpowiedzi. Ograniczenia AI obejmują niezawodność informacji przez tworzenie nieprawdziwych i nieaktualnych odpowiedzi na zapytania, brak pełnego zrozumienia kontekstu oraz zdolności do adaptacji do dynamicznych sytuacji.

Popularne i bezpłatne chatboty AI, takie jak Copilot i ChatGP, mogą z powodzeniem skutecznie wspomagać edukację informatyczną w zakresie podstaw programowania i przekazywać uczniom spersonalizowane interaktywne informacje zwrotne oraz wskazówki podobnie jak nauczyciel. Rozszerzone podejście do nauczania programowania powinno uwzględniać integrację narzędzi AI ze środowiskiem programistycznym Visual Studio Code.

Podsumowując, warto podkreślić, że wykorzystanie AI w nauczaniu programowania niesie ze sobą duże możliwości, jak i ograniczenia. AI może sobie poradzić z konkretnymi zadaniami, nie zastąpi jednak kreatywności i umiejętności rozwiązywania problemów przez programistów. Sztuczna inteligencja nie

---

<sup>17</sup> <https://github.com/github-copilot/signup>

powinna zniechęcać uczących się do studiowania metodologii programowania. AI należy postrzegać jako narzędzie, które odpowiednio wykorzystane uzupełnia i wzmacnia umiejętności zarówno uczących się, jak i profesjonalistów.

## Bibliografia

- Fazlagić J. (red.), *Sztuczna inteligencja jako megatrend kształtujący edukację*, Instytut Badań Edukacyjnych, Warszawa 2022.
- Hardy W., *Przewodnik AI po edukacji, czyli wszystko co musisz wiedzieć (na start)*, DELAB, Uniwersytet Warszawski 2023.
- Iwanowski W., *AI w praktyce. Narzędzia sztucznej inteligencji w programowaniu*, Wydawnictwo Videopoint 2023.
- Iwanowski W., *ChatGPT. OpenAI i sztuczna inteligencja w praktyce*, Wydawnictwo Videopoint 2023.
- Raport: *Wykorzystanie sztucznej inteligencji w edukacji [w:] Rynek pracy, edukacja, kompetencje. Aktualne trendy i wyniki badań*, PARP 2023.

## Netografia

- <https://chat.openai.com/>
- <https://code.visualstudio.com/>
- <https://copilot.microsoft.com/>
- <https://github.com/features/copilot>
- <https://github.com/github-copilot/signup>
- <https://intellipaat.com/blog/ai-tools-for-coding/>
- <https://learn.microsoft.com/pl-pl/visualstudio/ide/visual-studio-github-copilot-extension?view=vs-2022>
- <https://onlinedegrees.sandiego.edu/artificial-intelligence-education/>
- <https://platform.openai.com/account/api-keys>
- <https://www.youtube.com/watch?v=lOzxkPMcFw0>
- Kawik J., *GitHub Copilot z AI to prawdziwa rewolucja w programowaniu*, <https://aidriven.pl/ai/github-copilot-z-ai-to-prawdziwa-rewolucja-w-programowaniu/>, 2023
- Ramel D., *Top 10 AI 'Copilot' Tools for Visual Studio Code*, <https://visualstudiomagazine.com/articles/2023/06/30/vs-code-copilots.aspx> 2023
- Rymsza A., *Microsoft uczy SI, by programiści mieli łatwiej. Oto GitHub Copilot*, <https://www.telepolis.pl/tech/aplikacje/microsoft-github-sztuczna-inteligencja-programowanie>, 2021.
- Wrótniak K., *Top 10 AI Coding Assistant Tools – A Guide for Developers*, [https://www.thedroidsonroids.com/blog/best-ai-coding-assistant-tools?gad\\_source=5&gclid=EAI%20aIQobChMI2YDhb7qhAMVqFORBR05Iwi4EAAYAiAAEgIumfD\\_BwE](https://www.thedroidsonroids.com/blog/best-ai-coding-assistant-tools?gad_source=5&gclid=EAI%20aIQobChMI2YDhb7qhAMVqFORBR05Iwi4EAAYAiAAEgIumfD_BwE), 2024.