

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynierskich
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA ZAAWANSOWANE PROGRAMOWANIE

Testy jednostkowe - MergeSort

Autor:
Jakub Marek Tokarczyk

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2024

Spis treści

| | |
|--|-----------|
| 1. Ogólne określenie wymagań | 3 |
| 1.1. Cel projektu | 3 |
| 2. Analiza problemu | 4 |
| 2.1. Liczba Pi | 4 |
| 2.1.1. Cechy liczby Pi | 4 |
| 2.1.2. Zastosowania liczby Pi | 5 |
| 2.2. ChatGPT | 5 |
| 2.2.1. Funkcjonalności ChatGPT | 5 |
| 2.2.2. Wykorzystanie ChatGPT w różnych dziedzinach | 6 |
| 3. Projektowanie | 8 |
| 3.1. W projekcie będę korzystać z następujących narzędzi | 8 |
| 3.2. Sposób używania git | 9 |
| 4. Implementacja | 10 |
| 4.1. ChatGPT | 12 |
| 5. Wnioski i spostrzeżenia | 14 |
| 5.1. Spostrzeżenia | 14 |
| 5.1.1. Sztuczna inteligencja | 14 |
| 5.1.2. Poprawność kodu podanego przez ChatGPT | 14 |
| 5.2. Wnioski | 15 |
| Literatura | 16 |
| Spis rysunków | 17 |
| Spis tabel | 18 |
| Spis listingów | 19 |

1. Ogólne określenie wymagań

1.1. Cel projektu

Celem projektu jest napisanie programu w C++ wyliczającego przybliżoną wartość liczby PI metodą całkowania numerycznego, w którym umożliwiamy użytkownikowi ustawienie ilości liczb z przedziału całki oraz ilości wątków do zrównoleglenia obliczeń matematycznych. Założenia:

- Wykorzystanie biblioteki "thread" do zrównoleglenia obliczeń w systemach z standardem POSIX.
- Wyświetlanie czasu liczenia całki oraz wyniku na ekranie terminala
- Testy algorytmu dla różnych ilości liczb podziału z przedziału całki oznaczonej (100mln, 1mld, 3mld).
- Uruchamianie programu z różną liczbą wątków (od 1 do 50) i zapisywanie czasów pomiarów.

Program ma zostać napisany z wykorzystaniem modelu sztucznej inteligencji o nazwie ChatGPT oraz pomiary wyników i uwagi dotyczące AI mają zostać uwzględnione w dokumentacji.

2. Analiza problemu

2.1. Liczba Pi

Liczba Pi (π) jest jedną z najbardziej znanych stałych matematycznych i ma szerokie zastosowanie w różnych dziedzinach nauki, od matematyki po fizykę, inżynierię czy informatykę. Jest to liczba, która wyraża stosunek obwodu okręgu do jego średnicy. Wartość ta jest stała, niezależna od rozmiaru okręgu, co czyni ją fundamentem w geometrii i wielu innych dziedzinach.

2.1.1. Cechy liczby Pi

Cechy liczby Pi:

- **Niewymierność i nieskończoność:** liczba Pi jest liczbą niewymierną, co oznacza, że nie da się jej wyrazić jako iloraz dwóch liczb całkowitych (czyli nie jest ułamkiem skończonym ani okresowym). W wyniku tego jej rozwinięcie dziesiętne jest nieskończone i niepowtarzalne. Oznacza to, że w zapisie dziesiętnym liczby Pi występuje nieskończona liczba cyfr, które nie tworzą żadnego powtarzającego się wzorca,
- **Przybliżenia:** choć liczba Pi ma nieskończoną liczbę cyfr po przecinku, najczęściej używamy jej przybliżenia w postaci 3,14159. W praktyce, w zależności od potrzeb, liczba Pi może być przybliżana z różną precyzją. Na przykład, w codziennych obliczeniach wystarcza przybliżenie do dwóch miejsc po przecinku ($\pi \approx 3,14$), ale w bardziej skomplikowanych obliczeniach naukowych używa się wielu miejsc po przecinku, a w przypadku niektórych zaawansowanych obliczeń wykorzystywane są nawet setki tysięcy, a nawet miliony cyfr,
- **Liczba transcendentna:** liczba Pi jest również liczbą transcendentną, co oznacza, że nie jest pierwiastkiem żadnego wielomianu o współczynnikach całkowitych. To oznacza, że nie ma algebraicznego wyrażenia dla tej liczby, co ma istotne znaczenie w kontekście problemu kwadratury koła, który przez wieki stanowił ważne wyzwanie matematyczne. Problem polegał na tym, czy można przy użyciu tylko cyrkla i linijki skonstruować kwadrat o powierzchni równej powierzchni danego koła. Z powodu transcendentalności liczby Pi, okazało się, że jest to niemożliwe.

2.1.2. Zastosowania liczby Pi

Liczba Pi występuje w szerokim zakresie zastosowań matematycznych i praktycznych. Oto niektóre z nich:

- **Geometria:** obwód okręgu, powierzchnia koła, objętość kuli,
- **Fizyka:** liczba Pi pojawia się w wielu równaniach fizycznych, zwłaszcza w teorii fal, optyce, teorii względności, a także w obliczeniach związanych z ruchem harmonicznym, w tym w analizie drgań i oscylacji,
- **Statystyka i rachunek prawdopodobieństwa:** w statystyce liczba Pi występuje w rozkładzie normalnym, który jest powszechnie stosowany w analizach statystycznych. Wzór na funkcję gęstości rozkładu normalnego zawiera π , ponieważ funkcja ta jest oparta na krzywej Gaussa, której kształt jest związany z okręgiem.

Liczba Pi jest fundamentalnym elementem matematyki, mającym szerokie zastosowanie w naukach ścisłych. Jest liczbą niewymierną, transcendentną, a jej rozwinięcie dziesiętne jest nieskończone i niepowtarzalne. Choć jej pełna wartość jest niemożliwa do zapisania w postaci skończonej liczby cyfr, jej przybliżenie do kilku miejsc po przecinku jest wystarczające w większości zastosowań praktycznych. Zrozumienie liczby Pi jest kluczowe dla wielu dziedzin matematyki, fizyki, inżynierii i statystyki, a jej obliczenia stanowią ważny temat w historii matematyki i współczesnych badań komputerowych.

2.2. ChatGPT

ChatGPT to zaawansowany chatbot, stworzony przez OpenAI, który opiera się na modelu GPT (Generative Pretrained Transformer). Jego główną funkcją jest generowanie odpowiedzi na pytania użytkowników, ale jego możliwości wykraczają poza proste odpowiadanie na pytania – potrafi on także angażować się w bardziej złożone rozmowy na różnorodne tematy. ChatGPT wykorzystuje sztuczną inteligencję, która jest trenowana na ogromnych zbiorach danych tekstowych, aby umożliwić naturalną i płynną interakcję między człowiekiem a maszyną.

2.2.1. Funkcjonalności ChatGPT

Funkcjonalności ChatGPT:

- **Generowanie odpowiedzi:** ChatGPT potrafi odpowiedzieć na szeroki wachlarz pytań – od prostych, takich jak definicje, przez bardziej złożone pytania, które wymagają analizy kontekstu, aż po zadania wymagające logicznego rozumowania. Model jest w stanie wykorzystywać informacje zawarte w swojej bazie danych do generowania odpowiedzi, które są najbardziej trafne w kontekście zapytania użytkownika,
- **Angażowanie się w rozmowy:** ChatGPT może utrzymywać płynne rozmowy z użytkownikiem, przechodzić z tematu na temat i dostosowywać swoje odpowiedzi do przebiegu konwersacji. Dzięki temu przypomina bardziej naturalne rozmowy prowadzone z drugim człowiekiem, gdzie model rozumie kontekst rozmowy i stara się odpowiedzieć w sposób, który będzie pasował do sytuacji,
- **Wyspecjalizowane dziedziny wiedzy:** chociaż ChatGPT ma dostęp do ogólnych informacji na szeroki zakres tematów, to jego zakres wiedzy obejmuje również bardziej zaawansowane dziedziny. Model jest w stanie odpowiadać na pytania dotyczące matematyki, nauk ścisłych, programowania, historii, sztuki czy filozofii, choć należy pamiętać, że jego odpowiedzi bazują na danych, na których został wytrenowany, i mogą nie zawsze być w pełni aktualne.

2.2.2. Wykorzystanie ChatGPT w różnych dziedzinach

Wykorzystanie ChatGPT w różnych dziedzinach:

- **Edukacja:** ChatGPT znajduje zastosowanie w edukacji, pomagając uczniom i studentom w nauce, wyjaśniając trudne pojęcia, rozwiązywaniu zadań matematycznych czy udzielaniu informacji na temat różnych dziedzin nauki. Może również służyć jako narzędzie do nauki języków obcych, umożliwiając prowadzenie rozmów na różne tematy w wybranym języku,
- **Obsługa klienta:** ChatGPT jest wykorzystywany w obsłudze klienta do automatyzacji odpowiedzi na pytania dotyczące produktów, usług czy procesów w firmach. Dzięki zdolności do rozumienia i generowania języka naturalnego, model może skutecznie odpowiadać na zapytania, zmniejszając potrzebę angażowania człowieka do każdej interakcji,
- **Tworzenie treści:** dzięki swojej zdolności do generowania tekstów na różne tematy, ChatGPT może być używany do wspomagania twórców treści, dziennikarzy czy pisarzy. Może dostarczyć inspiracji, pomóc w pisaniu artykułów,

generować teksty oparte na określonych słowach kluczowych czy nawet pisać kody komputerowe w różnych językach programowania,

- **Zastosowania w biznesie i marketingu:** ChatGPT jest także wykorzystywany w marketingu do tworzenia treści reklamowych, analizowania opinii klientów czy wspierania działań związanych z komunikacją wewnętrzną i zewnętrzną. Może także pomóc w analizie danych i generowaniu raportów.

Pomimo ogromnych możliwości, ChatGPT napotyka pewne ograniczenia:

- **Brak rzeczywistej wiedzy:** model, choć jest w stanie generować odpowiedzi, nie posiada prawdziwej "wiedzy" o świecie. Odpowiedzi opierają się na danych, na których model został wytrenowany, ale nie są one weryfikowane w czasie rzeczywistym, co może prowadzić do błędnych lub nieaktualnych informacji,
- **Ograniczenia etyczne i bezpieczeństwo:** istnieją również obawy dotyczące etycznego wykorzystania technologii. ChatGPT może być używany do generowania dezinformacji, nieodpowiednich treści lub nawet do manipulacji opinią publiczną. Dodatkowo, choć model stara się ograniczać nieodpowiednie lub szkodliwe odpowiedzi, nie zawsze udaje się uniknąć generowania kontrowersyjnych treści,
- **Zrozumienie kontekstu:** chociaż ChatGPT jest bardzo zaawansowany, to wciąż może mieć trudności ze zrozumieniem kontekstu w bardziej skomplikowanych sytuacjach, szczególnie gdy chodzi o subtelne niuanse emocjonalne czy kulturowe. Może również mieć trudności w generowaniu odpowiedzi w sytuacjach wymagających głębokiego rozumienia i kreatywności.

ChatGPT jest jednym z najnowocześniejszych osiągnięć w dziedzinie sztucznej inteligencji, który zrewolucjonizował sposób interakcji człowieka z maszyną. Dzięki zaawansowanemu modelowi GPT, ChatGPT może angażować się w naturalne, płynne rozmowy na różnorodne tematy, dostarczając cennych informacji i wsparcia w wielu dziedzinach. Mimo swoich licznych zalet, narzędzie to nadal boryka się z wyzwaniami związanymi z dokładnością, etyką i kontekstem odpowiedzi, które wymagają dalszego rozwoju technologii i ścisłej kontroli nad jej wykorzystaniem.

3. Projektowanie

3.1. W projekcie będę korzystać z następujących narzędzi

- **Visual Studio 2022¹**: środowisko programistyczne, które będzie wykorzystywane do pisania, kompilowania i debugowania kodu. Visual Studio 2022 zapewnia pełne wsparcie dla języka C++ oraz zaawansowane funkcje debugowania i zarządzania projektem,
- **Język programowania C++²**: programowanie w języku C++ pozwoli na efektywną implementację algorytmów oraz struktur danych, takich jak lista dwukierunkowa. Język ten zapewnia dużą kontrolę nad pamięcią i wydajnością, co jest istotne w wielu projektach algorytmicznych i systemowych,
- **Git³**: system kontroli wersji, który będzie wykorzystywany do zarządzania wersjami kodu, śledzenia zmian. Git umożliwia łatwe zarządzanie historią projektu i integrację z platformami takimi jak GitHub,
- **GitHub⁴**: platforma oparta na systemie Git, która umożliwia hostowanie repozytoriów kodu w chmurze. GitHub nie tylko zapewnia zdalne przechowywanie projektów, ale także oferuje zaawansowane narzędzia do współpracy, takie jak pull requesty, które umożliwiają przeglądanie i zatwierdzanie zmian przed ich włączeniem do głównej wersji kodu. Platforma oferuje również funkcje zarządzania projektami, takie jak issues do śledzenia błędów i zadań, a także wikis i documentation, które pomagają w tworzeniu pełnej dokumentacji projektu. GitHub wspiera również integrację z narzędziami CI/CD, co pozwala na automatyzację procesów budowania i testowania aplikacji.
- **ChatGPT⁵**: to zaawansowany chatbot, stworzony przez OpenAI, który opiera się na modelu GPT. Jego główną funkcją jest generowanie odpowiedzi na pytania użytkowników, ale jego możliwości wykraczają poza proste odpowiadanie na pytania – potrafi on także angażować się w bardziej złożone rozmowy na różnorodne tematy. ChatGPT wykorzystuje sztuczną inteligencję, która jest trenowana na ogromnych zbiorach danych tekstowych, aby umożliwić naturalną i płynną interakcję między człowiekiem a maszyną.

¹Visual Studio 2022[1]

²C++[2]

³Strona główna systemu kontroli wersji Git[3]

⁴GitHub[4]

⁵ChatGPT[5]

3.2. Sposób używania git

Git to system kontroli wersji, który pozwala na śledzenie i zarządzanie historią zmian w plikach projektu. Aby rozpocząć korzystanie z Gita, należy wykonać kilka podstawowych kroków, które obejmują konfigurację, tworzenie repozytorium oraz zarządzanie wersjami plików. Na początku trzeba zainstalować Git na swoim komputerze. Git jest dostępny dla systemów operacyjnych Windows, macOS i Linux, a instalację można pobrać ze strony oficjalnej. Po instalacji warto skonfigurować swoje dane użytkownika, aby Git wiedział, kto dokonał zmian w repozytorium. Po skonfigurowaniu Gita, można rozpocząć pracę nad repozytorium. Pierwszym krokiem jest utworzenie nowego repozytorium. Kiedy repozytorium jest gotowe, możemy zacząć pisanie kodu. Git umożliwia synchronizowanie lokalnych zmian z zewnętrznym repozytorium, co oznacza że możemy wysłać i pobrać zmiany kodu. Aby pobrać najnowsze zmiany z repozytorium, używamy komendy `git pull` a aby wysłać nasze zmiany na serwer, używamy `git push`. Dzięki temu systemowi każdy programista może swobodnie pracować nad projektem, a Git zapewnia kontrolę nad historią zmian, umożliwiając powrót do poprzednich wersji plików.

4. Implementacja

```
1 #include <iostream>
2 #include <thread>
3 #include <vector>
4 #include <chrono>
5 #include <cmath>
6
7 // Funkcja licząca wartość części całki na danym przedziale
8 void calculate_partial_integral(double start, double end, double
    step, double& result) {
9     double partial_sum = 0.0;
10    for (double x = start; x < end; x += step) {
11        partial_sum += 4.0 / (1.0 + x * x) * step;
12    }
13    result = partial_sum;
14 }
15
16 int main() {
17     // Ustawienia początkowe
18     size_t num_intervals; // Liczba przedziałów (np. 100 mln, 1
    mld, 3 mld)
19     int num_threads;      // Liczba wątków (1-50)
20
21     std::cout << "Podaj liczbę przedziałów: ";
22     std::cin >> num_intervals;
23     std::cout << "Podaj liczbę wątków: ";
24     std::cin >> num_threads;
25
26     // Obliczanie kroku całkowania
27     double step = 1.0 / num_intervals;
28
29     // Wektory przechowujące wątki i wyniki
30     std::vector<std::thread> threads;
31     std::vector<double> results(num_threads, 0.0);
32
33     // Mierzenie czasu
34     auto start_time = std::chrono::high_resolution_clock::now();
35
36     // Przydzielenie przedziałów dla każdego wątku
37     for (int i = 0; i < num_threads; ++i) {
38         double thread_start = i * (1.0 / num_threads);
39         double thread_end = (i + 1) * (1.0 / num_threads);
40         threads.emplace_back(calculate_partial_integral,
            thread_start, thread_end, step, std::ref(results[i]));
    }
```

```

41     }
42
43     // Dolaczenie watkow
44     for (auto& thread : threads) {
45         thread.join();
46     }
47
48     // Sumowanie wynikow
49     double pi = 0.0;
50     for (const auto& result : results) {
51         pi += result;
52     }
53
54     // Mierzenie czasu
55     auto end_time = std::chrono::high_resolution_clock::now();
56     std::chrono::duration<double> elapsed = end_time - start_time;
57
58     // Wyswietlanie wynikow
59     std::cout << "Przyblizona wartosc liczby PI: " << pi << std::
endl;
60     std::cout << "Czas obliczen: " << elapsed.count() << " sekund"
<< std::endl;
61
62     return 0;
63 }

```

Listing 1. Klasa MergeSort

Powyżej został ukazany kod programu. Poniżej znajdują się wyjaśnienia kilku kluczowych linii:

Linijka 8: funkcja `calculatePartialIntegral` oblicza wartość częściową całki w zakresie od `start` do `end` z krokiem `step`. Używa wzoru na całkę funkcji, co odpowiada równaniu dla wyznaczenia liczby PI metodą całkowania numerycznego. Wynik zapisywany jest do zmiennej `result`, przekazanej przez referencję.

Linijki 30-31: to dynamiczna lista wątków. Każdy wątek będzie wykonywał obliczenia dla innego fragmentu całki, `std::vector<double>` przechowuje wyniki cząstkowe obliczone przez wątki.

Linijka 34 rejestracja aktualnego czasu na początku obliczeń, funkcja `high_resolution_clock::now()` daje precyzyjny znacznik czasu.

Linijki 40-41: tworzenie wątków:

- funkcja `emplace_back` dodaje nowy wątek do wektora `threads`,
- wątek wykonuje funkcję `calculatePartialIntegral`, a jego wyniki są zapisywane

w `results[i]`,

- `std::ref(results[i])`: przekazuje referencję do odpowiedniego elementu wektora wyników, co umożliwia zapisanie wyniku przez wątek.

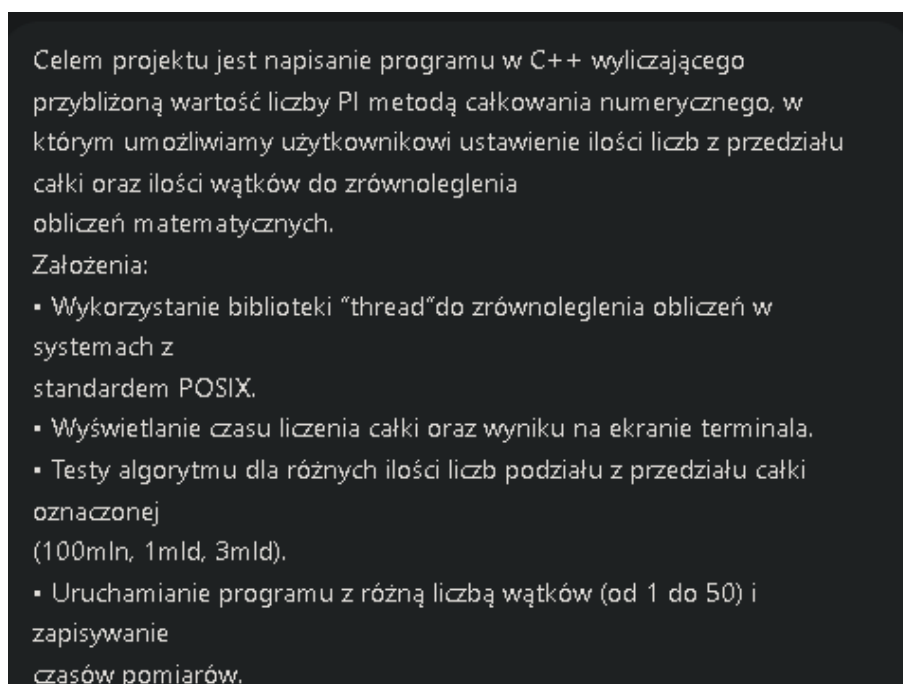
Linijki 44-46: funkcja `join()` zatrzymuje główny wątek programu, dopóki wszystkie wątki w wektorze `threads` nie zakończą swoich obliczeń, bez tej liniiki główny program mógłby zakończyć działanie przed ukończeniem obliczeń w wątkach.

Linijki 50-52: iteracja przez wektor `results` i dodawanie wyników obliczonych przez poszczególne wątki, suma cząstkowych wartości całki daje przybliżoną wartość liczby π .

Linijki 55-56: rejestracja końcowego znacznika czasu i obliczenie czasu trwania programu za pomocą różnicy między `end time` a `start time`.

4.1. ChatGPT

Zgodnie z założeniami projektu program został napisany po przez tworzenie zapytań używając ChatGPT. Uzyskane odpowiedzi były wykorzystywane podczas pisania programu i ewentualne błędy były dalej konsultowane z "chatem".



Celem projektu jest napisanie programu w C++ wyliczającego przybliżoną wartość liczby π metodą całkowania numerycznego, w którym umożliwiamy użytkownikowi ustawienie ilości liczb z przedziału całki oraz ilości wątków do zrównoleglenia obliczeń matematycznych.

Założenia:

- Wykorzystanie biblioteki "thread" do zrównoleglenia obliczeń w systemach z standardem POSIX.
- Wyświetlanie czasu liczenia całki oraz wyniku na ekranie terminala.
- Testy algorytmu dla różnych ilości liczb podziału z przedziału całki oznaczonej (100mln, 1mld, 3mld).
- Uruchamianie programu z różną liczbą wątków (od 1 do 50) i zapisywanie czasów pomiarów.

Rys. 4.1. Zapytanie

```
1  #include <iostream>
2  #include <thread>
3  #include <vector>
4  #include <chrono>
5  #include <cmath>
6
7  // Funkcja licząca wartość części całki na zadanym przedziale
8  void calculate_partial_integral(double start, double end, double step, double& result) {
9      double partial_sum = 0.0;
10     for (double x = start; x < end; x += step) {
11         partial_sum += 4.0 / (1.0 + x * x) * step;
12     }
13     result = partial_sum;
14 }
15
16 int main() {
17     // Ustawienia początkowe
18     size_t num_intervals; // Liczba przedziałów (np. 100 mln, 1 mld, 3 mld)
19     int num_threads;      // Liczba wątków (1-50)
20
21     std::cout << "Podaj liczbę przedziałów: ";
22     std::cin >> num_intervals;
23     std::cout << "Podaj liczbę wątków: ";
24     std::cin >> num_threads;
25
26     // Obliczanie kroku całkowania
27     double step = 1.0 / num_intervals;
28
29     // Wektory przechowujące wątki i wyniki
30     std::vector<std::thread> threads;
31     std::vector<double> results(num_threads, 0.0);
32
33     // Mierzenie czasu
34     auto start_time = std::chrono::high_resolution_clock::now();
35
36     // Przydzielenie przedziałów dla każdego wątku
37     for (int i = 0; i < num_threads; ++i) {
38         double thread_start = i * (1.0 / num_threads);
39         double thread_end = (i + 1) * (1.0 / num_threads);
40         threads.emplace_back(calculate_partial_integral, thread_start, thread_end, step, std::ref(results[i]));
41     }
42
43     // Dołączenie wątków
44     for (auto& thread : threads) {
45         thread.join();
46     }
47
48     // Sumowanie wyników
49     double pi = 0.0;
50     for (const auto& result : results) {
51         pi += result;
52     }
53
54     // Mierzenie czasu
55     auto end_time = std::chrono::high_resolution_clock::now();
56     std::chrono::duration<double> elapsed = end_time - start_time;
57
58     // Wyświetlanie wyników
59     std::cout << "Przybliżona wartość liczby PI: " << pi << std::endl;
60     std::cout << "Czas obliczeń: " << elapsed.count() << " sekund" << std::endl;
61
62     return 0;
63 }
64
```

Rys. 4.2. Odpowiedź

5. Wnioski i spostrzeżenia

5.1. Spostrzeżenia

5.1.1. Sztuczna inteligencja

- Sztuczna inteligencja staje się coraz bardziej powszechnym narzędziem w pracy programistów, wspomagając proces tworzenia oprogramowania. Jednak na początku korzystania z takich narzędzi warto samodzielnie zweryfikować wygenerowany kod. Ręczna weryfikacja jest kluczowa, aby upewnić się, że kod jest czytelny, zgodny z wymaganiami projektu oraz spełnia oczekiwania jakościowe. Choć SI może znacznie usprawnić pracę, to nadal główną rolę w tworzeniu oprogramowania odgrywa doświadczenie i wiedza programisty. To on nie tylko zna aspekty techniczne, ale także rozumie kontekst projektu, co jest niezbędne do dostosowania rozwiązania do jego specyficznych potrzeb.

5.1.2. Poprawność kodu podanego przez ChatGPT

- Podczas korzystania z ChatGPT, może się zdarzyć, że proponowane sugestie nie zawsze będą idealne lub w pełni zrozumiałe. Często generowany kod może nie odpowiadać zamierzonym celom lub zawierać błędy składniowe. Jednak w miarę kolejnych interakcji, gdy użytkownik dostarcza dodatkowe wskazówki i korekty, ChatGPT staje się coraz lepszy w dostarczaniu trafniejszych odpowiedzi. Wynika to z faktu, że model uczy się na podstawie tych interakcji, co pozwala mu lepiej zrozumieć intencje programisty. Choć ChatGPT może stanowić wartościowe wsparcie w procesie pisania kodu, należy pamiętać, że generowane odpowiedzi nie zawsze będą idealne. Istnieje również ryzyko, że sztuczna inteligencja nie zawsze w pełni uchwyci kontekst projektu lub specyfikę rozwiązywanego problemu.

5.2. Wnioski

Projekt dotyczył narzędzia sztucznej inteligencji o nazwie ChatGPT, stworzonego przez firmę OpenAI. Choć model ten jest bardzo przydatny w kontekście pisania kodu, warto podkreślić zarówno jego mocne strony, jak i ograniczenia. Oto kluczowe aspekty dotyczące ChatGPT w kontekście tworzenia oprogramowania:

- **Przydatność w pisaniu kodu:** ChatGPT jest narzędziem, które znacząco wspomaga programistów w codziennej pracy, szczególnie w zakresie generowania kodu, rozwiązywania problemów programistycznych oraz oferowania sugestii,
- **Ograniczenia i niedoskonałości:** mimo zaawansowanej technologii, ChatGPT nie zawsze potrafi dokładnie zrozumieć intencje programisty, co może prowadzić do błędów w generowanym kodzie. Czasami kod może być nieoptymalny lub nie spełniać założeń projektowych. W przypadku bardziej skomplikowanych projektów, model może nie w pełni uchwycić specyfikę problemu, co skutkuje generowaniem nieadekwatnych rozwiązań,
- **Automatyzacja procesu tworzenia kodu:** ChatGPT stanowi krok naprzód w automatyzacji tworzenia kodu. Dzięki temu programiści mogą zaoszczędzić czas, unikając konieczności pisania prostych fragmentów kodu od podstaw. Narzędzie może zautomatyzować część rutynowych zadań, umożliwiając programistom skoncentrowanie się na bardziej skomplikowanych częściach projektu,
- **Potencjał na przyszłość:** mimo swoich obecnych ograniczeń, ChatGPT rozwija się w szybkim tempie. Z biegiem czasu i dzięki dalszym ulepszeniom, model może stać się jeszcze bardziej precyzyjny i użyteczny.

Bibliografia

- [1] *Visual Studio 2022*. URL: <https://visualstudio.microsoft.com/pl/vs/getting-started/>.
- [2] *Nauka c++*. URL: <https://www.w3schools.com/cpp/>.
- [3] *Git*. URL: <https://git-scm.com/>.
- [4] *Github*. URL: <https://github.com/>.
- [5] *ChatGPT*. URL: <https://openai.com/index/chatgpt/>.

Spis rysunków

| | |
|--------------------------|----|
| 4.1. Zapytanie | 12 |
| 4.2. Odpowiedź | 13 |

Spis tabel

Spis listingów

| | | |
|----|---------------------------|----|
| 1. | Klasa MergeSort | 10 |
|----|---------------------------|----|