**Dokumentacja projektu Lab3**

**Dane autora:** Dawid Różański 263524

**Link do repozytorium:** <https://github.com/RozDaw/.NET-i-Java>

**Drzewo projektu:**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, multimedia

Opis wygenerowany automatycznie**

**Opis projektu:**

Wykonałem jedynie dwa pierwsze zadania, na przetwarzanie obrazów nie starczyło mi już czasu. Projekt został wykonany przy użyciu windows forms.

Wszystkie modyfikacje problemu, sterownie ilością wątków itd, modyfikowane są z poziomu aplikacji okienkowej.

**Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie**

Użytkownik ma możliwość:

* Zmiany ziarna losowania macierzy wyjściowych
* Zaznaczania checkBoxa z wyborem czy macierze mają się wyświetlać na ekranie
* Poruszanie suwakiem zmieniającym wielkość wyświetlania macierzy
* Zmiana ilości tworzonych wątków
* Zmiana rozmiaru macierzy (tylko kwadratowa)
* Wybór czy do rozwiązania problemu użyć klasy Thread czy Parallel

Ponadto z prawej strony ekranu wyświetla się lista stworzonych wątków, w raz z zadaniami które dostały (dotyczy jedynie trybu Thread)

W projekcie wykorzystałem trzy klasy:

* Klasa matrix, przechowująca macierze wejściowe, obsługę losowania ich zawartości, oraz przeciążoną metodę toString().
* Klasa result, przechowująca rozwiązanie problemu i obsługę wielowątkowości.
* Klasa form1, obsługująca GUI.

Po kliknięciu przycisku Start, program generuje instancję problemu, rozpoczyna mierzenie czasu i wywołuje metodę result.start, rozpoczynającą mnożenie macierzy. Na samym końcu drukuje podsumowanie.

Fragment kodu odpowiedzialny za wielowątkowe mnożenie z wykorzystaniem Thread:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Fragment kodu, odpowiedzialny za wielowątkowe mnożenie z wykorzystaniem Parallel:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

Na pierwszy rzut oka rzuca się dużo mniejsza objętość kodu, przy wykorzystaniu Parallel.

W tabeli opracowane zostały wyniki pomiaru czasu (w sekundach), dla obu rozwiązań, przy różnych rozmiarach macierzy:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Można zauważyć że wykorzystywanie wielu wątków, istotnie skraca czas rozwiązania problemu, w przybliżeniu tyle samo razy, ile razy więcej wątków zostało wykorzystane.

Jednak czas ten przestaje spadać, gdy ilość wątków przekracza ilość rdzeni logicznych procesora. Wątki zaczynają się wówczas prześcigać i działać quasi równolegle.

Można również zauważyć, że przy małych rozmiarach macierzy, a co za tym idzie, krótkim czasie rozwiązania problemu, i przy stosunkowo dużej ilości wątków; czas rozwiązanie problemu, również przestaje spadać, gdyż czas zużywany na tworzenie i obsługę wątków zaczyna być relatywnie duży.

Ostatnim wnioskiem jest porównanie efektywności klas Thread i Parallel. Moje pomiary wskazują na niewielką różnicę w działaniu obu rozwiązań, z lekką przewagą dla klasy Thread.