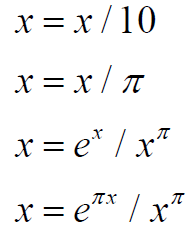
**Лабораторна робота №3**

Тема 1: ПАРАЛЕЛІЗМ ДАНИХ ТА ПАРАЛЕЛІЗМ ЗАДАЧ (Ч. 2)

Завдання:

1. Створити програму, що використовує для розпаралелювання метод Parallel.For(). Провести ряд обчислювальних експериментів із різним типом елементів масиву (int, double); різною кількістю елементів масиву; різною складністю обчислень:

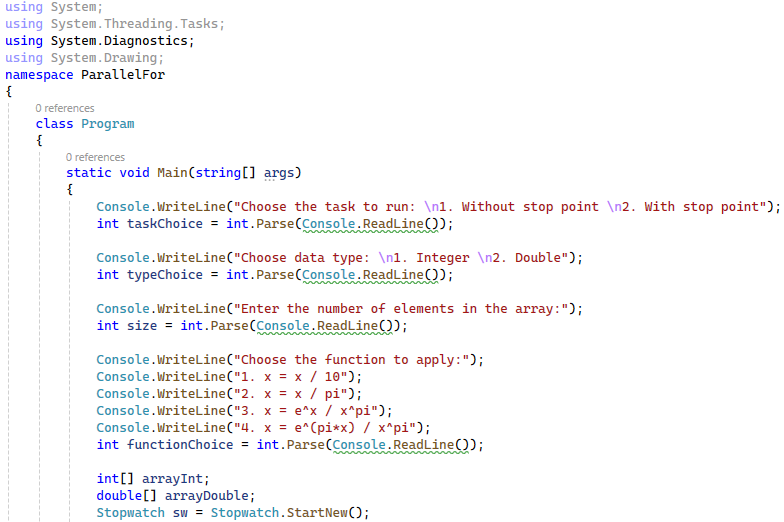


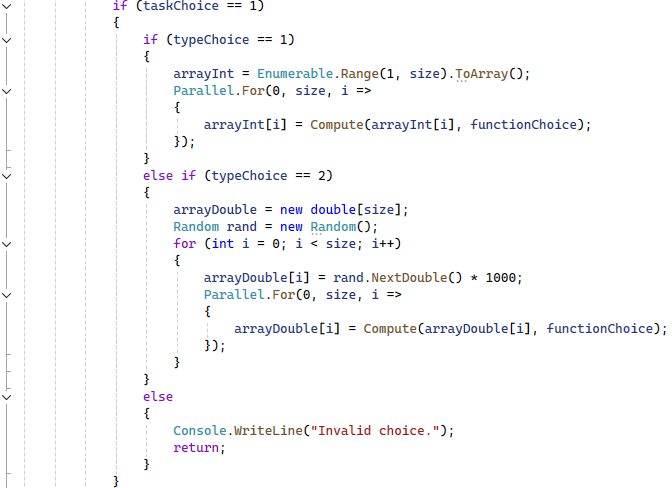
Виміряти час, що витрачається паралельним способом обробки та послідовним. Результати експериментів оформити у вигляді таблиці. Зробити висновки.

1. Модифікувати попередню програму (створивши новий проект у рішенні) таким чином, щоб відбувався вихід з паралельного циклу за умови входження значення елемента у деякий окіл деякого числа (число та відхилення задаються константами).
2. Повторити приклади розпаралелювання за допомогою ForEach().
3. Модифікувати попередню програму (створивши новий проект у рішенні) таким чином, щоб тіло паралельного циклу задавалося лямбда-виразом.

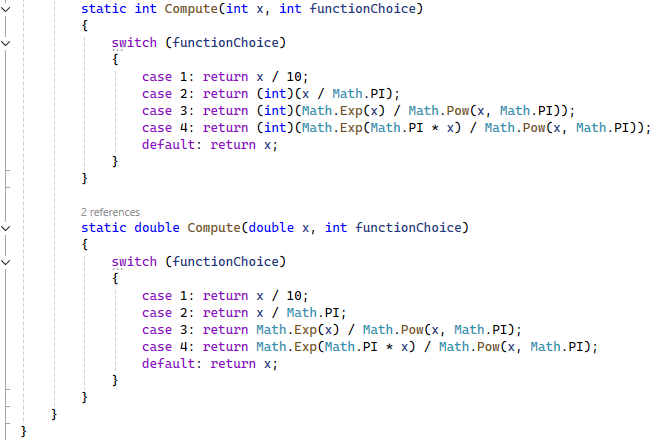
Хід роботи:

1. Для створення програми завдань 1-2 пропишемо такий код:









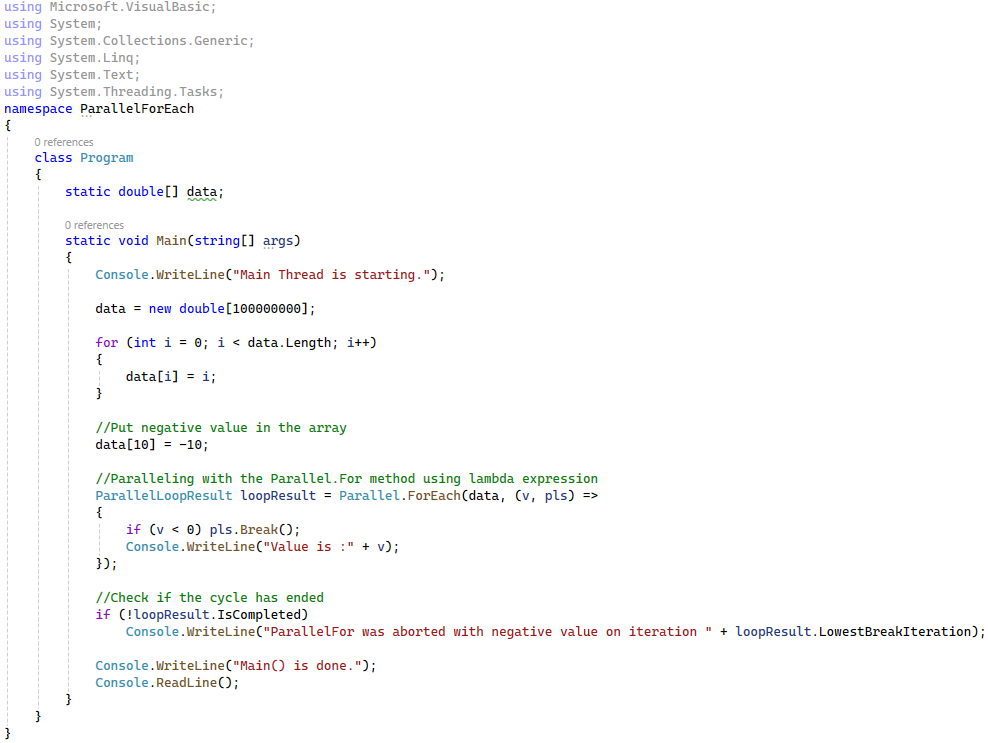
Тут користувач може обрати, який саме варіант завдання виконати (1 чи 2), задаючи необхідні параметри.

Результати експериментів витраченого часу паралельним способом обробки та послідовним:

| Спосіб обробки  Складність обчислень | Паралельний | | Послідовний | |
| --- | --- | --- | --- | --- |
| Час обробки (секунди) | | | |
| Integer | Double | Integer | Double |
|  | 0.109124 | 0.0492139 | 0.0519989 | 0.002187 |
|  | 0.0318339 | 0.0214097 | 0.0004277 | 0.0007788 |
|  | 0.0125981 | 0.0293806 | 0.002932 | 0.0040015 |
|  | 0.0423109 | 0.014487 | 0.0017966 | 0.0026564 |

Паралельна обробка в даному експерименті не забезпечує приросту продуктивності, а навпаки, поступається послідовному виконанню. У всіх тестованих випадках послідовний алгоритм працює швидше, ніж паралельний. Це можна пояснити тим, що створення та управління потоками вимагає додаткових ресурсів, що призводить до значних накладних витрат. Особливо це помітно для простих операцій (, ), де час обробки у паралельному режимі значно більший порівняно з послідовним. Навіть для складніших обчислень (, ) перевага послідовного підходу зберігається, що свідчить про те, що використання багатопотоковості неефективне для такого типу задач. Крім того, обробка чисел типу Double виявляється швидшою, ніж обробка Integer, незалежно від обраного методу.

1. Для завдання 3-4 перепишемо код і модифікуємо тіло паралельного циклу у лямбда вираз:



**Висновки**: у ході лабораторної роботи було детальніше розглянуто термін паралелізм. Було досліджено створення паралельних задач, використовуючи методи Parallel.For() та ForEach(). Також було проведено експеримент на дослідження витрати часу послідовно оброблюваних даних та паралельно оброблюваних даних. У результаті дійдено висновків, що не завжди паралельний підхід обробки даних є ефективнішим за послідовний.