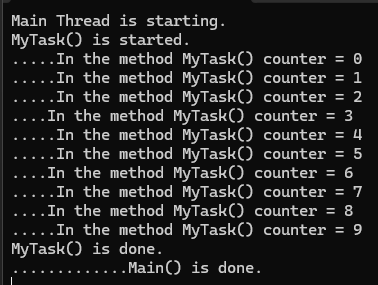
Лабораторна робота №2

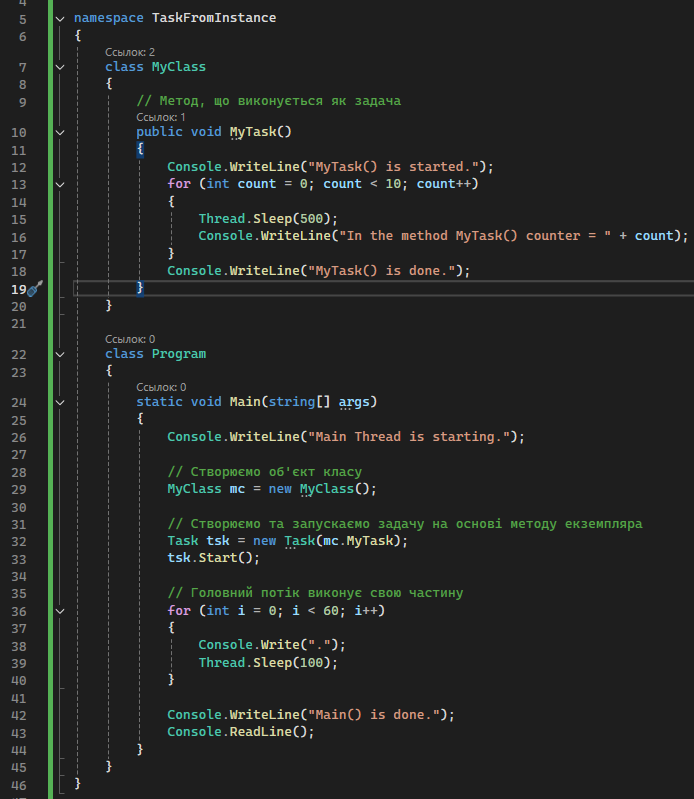
ПАРАЛЕЛІЗМ ДАНИХ ТА ПАРАЛЕЛІЗМ ЗАДАЧ (Ч. 1)

1. Повторити задачі, розв’язані у тексті.
   1. Створення задачі.



Результат виконання:

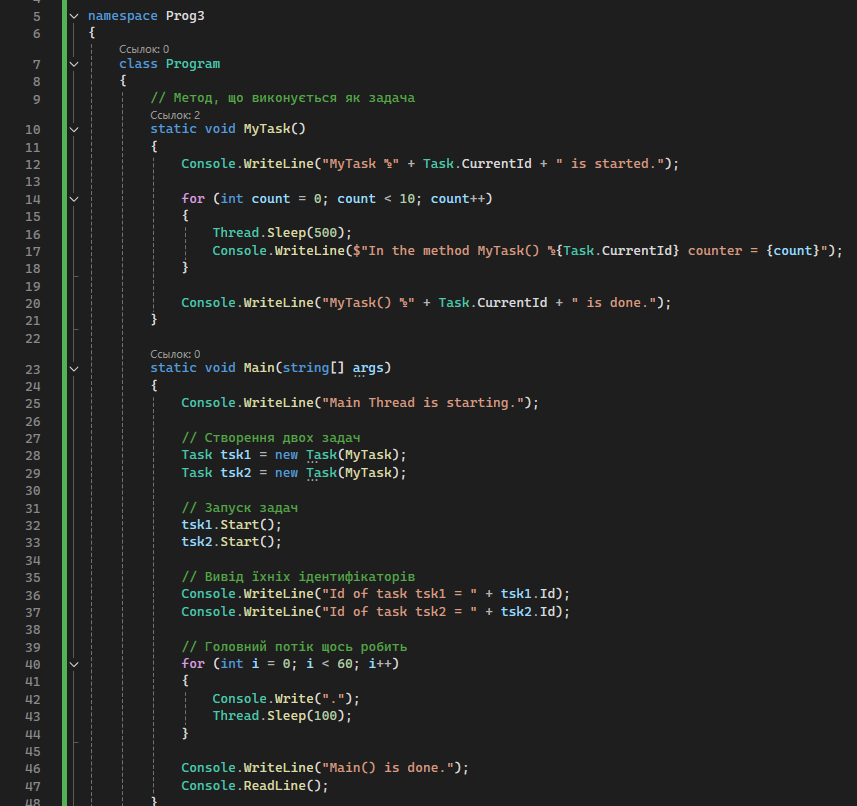




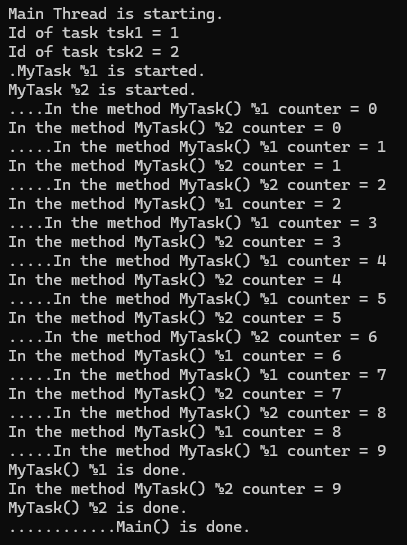
Результат виконання аналогічний до попереднього.

Цей приклад показує, що метод задачі не обов’язково має бути статичним. Ми створюємо екземпляр класу, передаємо його метод у задачу й запускаємо її.

* 1. Застосування ідентифікатора задачі.

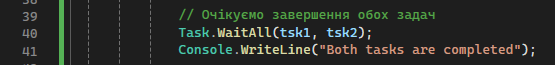


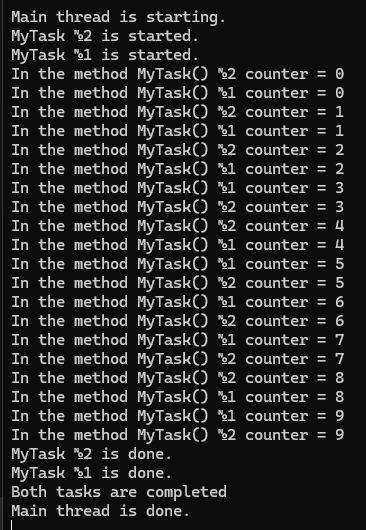
Результат виконання:



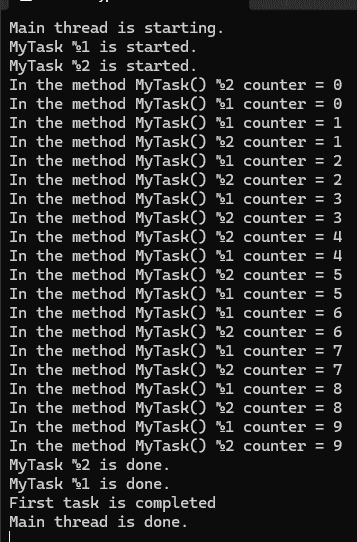
tsk1.Id та tsk2.Id — унікальні цілі числа, які присвоюються при створенні задач. Всередині MyTask() – Task.CurrentId дозволяє вивести, яка саме задача зараз виконується. Вивід покаже, як обидві задачі паралельно змінюють свої лічильники

Спробуємо використати Очікування завершення задач: Wait(), WaitAll(), WaitAny()

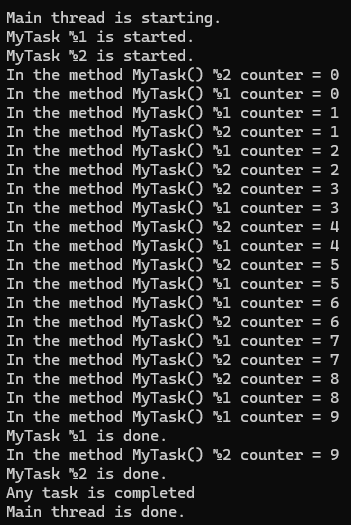




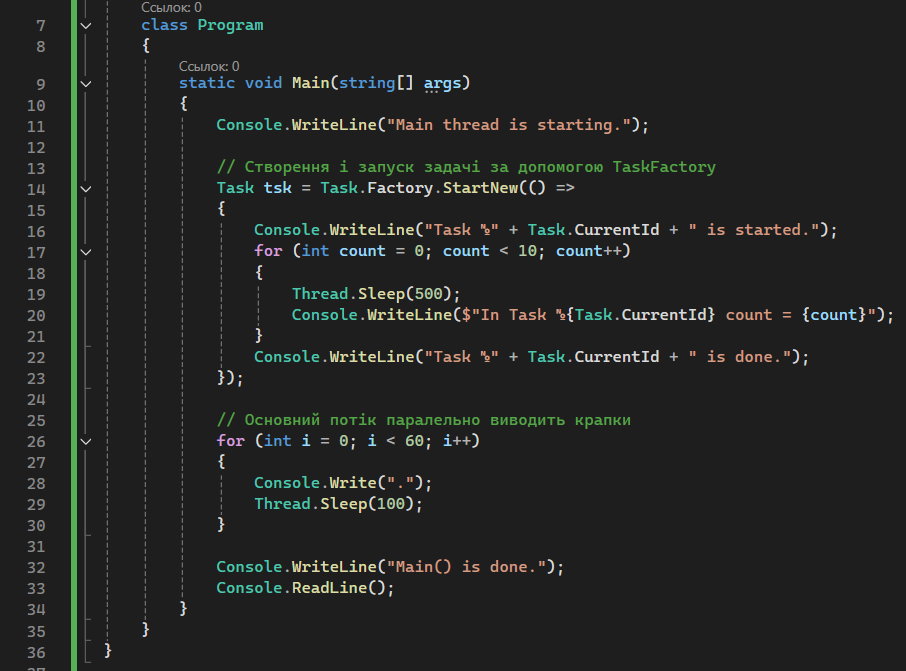


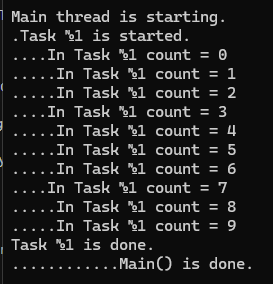




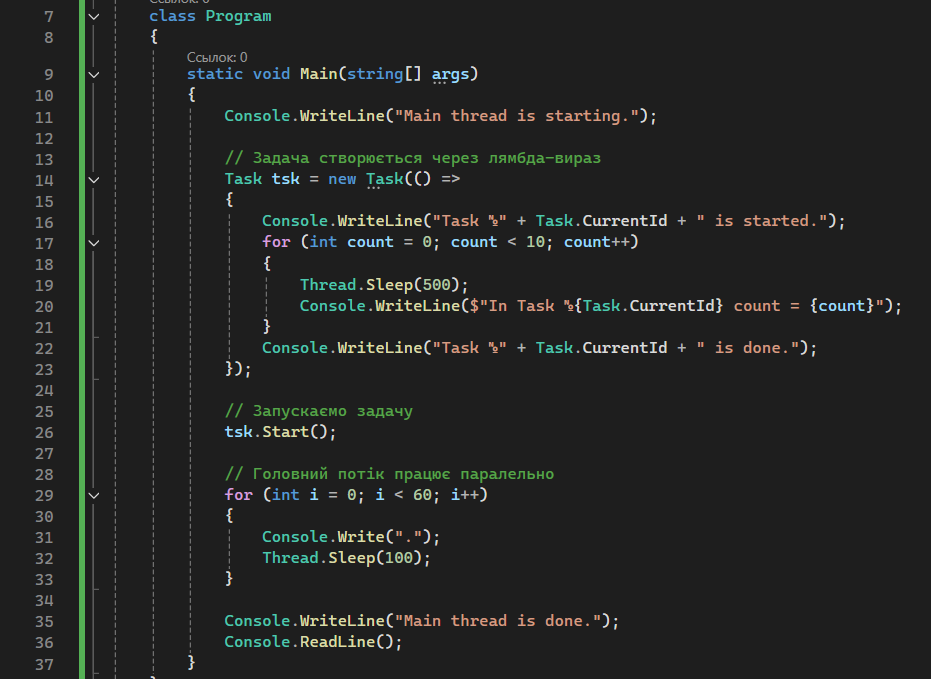


* 1. Застосування класу TaskFactory для запуску задачі.

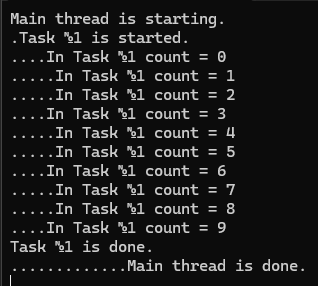




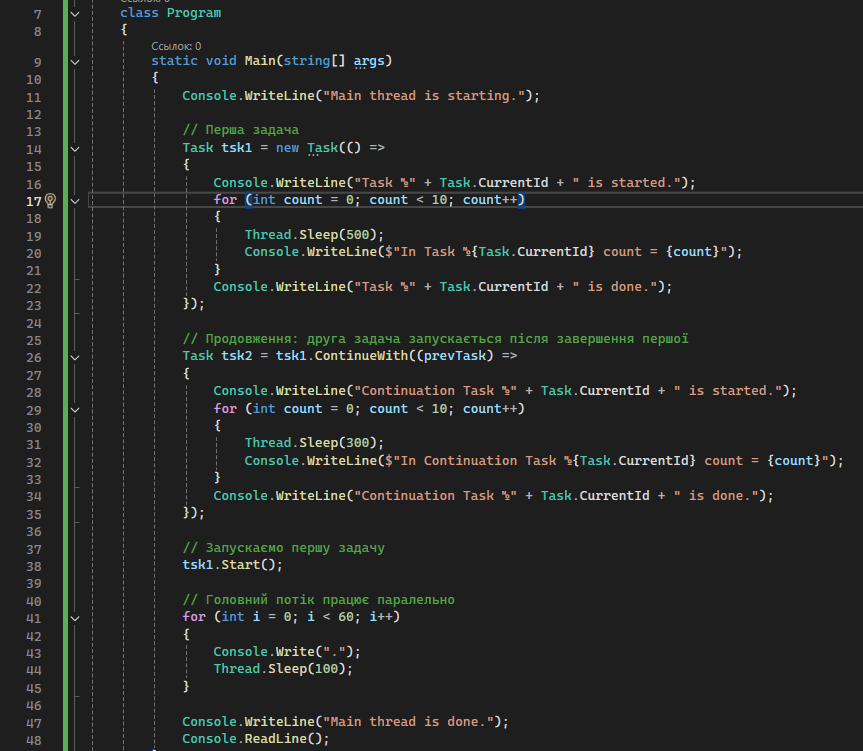
* 1. Застосування лямбда-виразу в якості задачі.



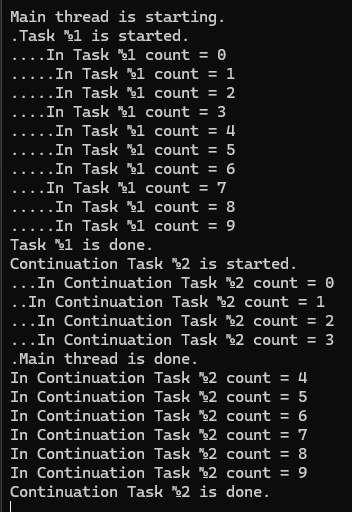
Результат виконання:



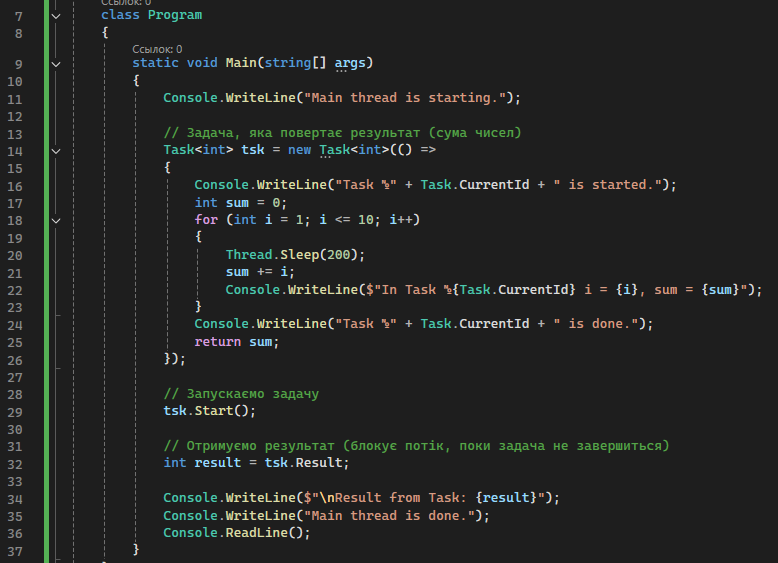
* 1. Створення продовження задачі.



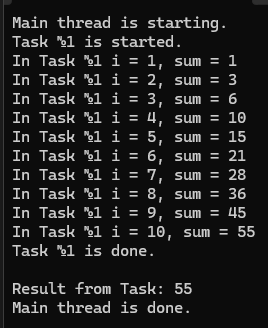
Результат виконання:



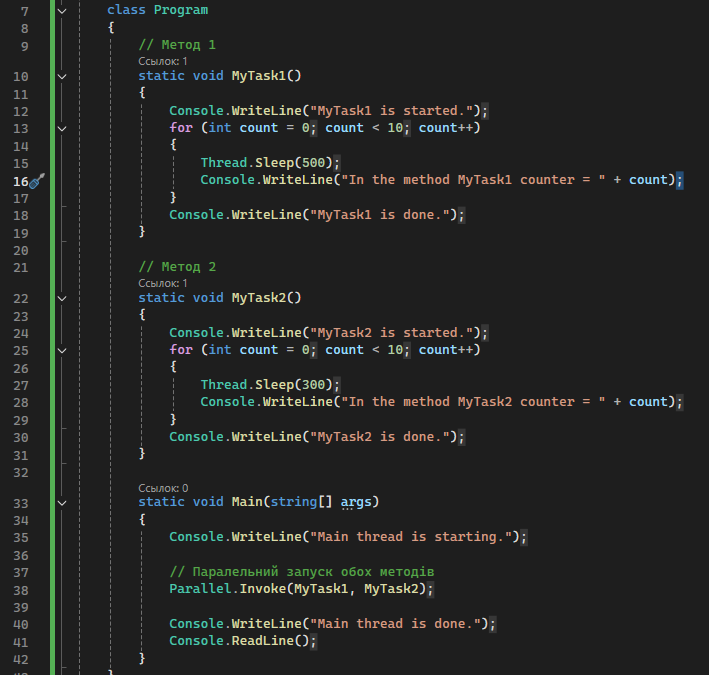
* 1. Повернення значення із задачі.



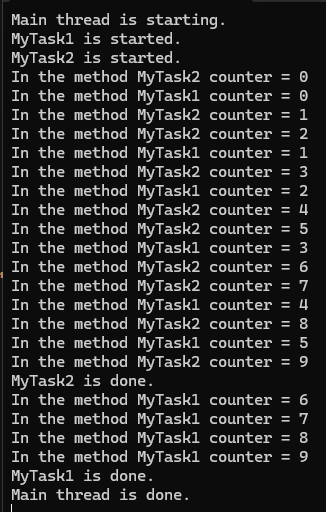
Результат виконання:



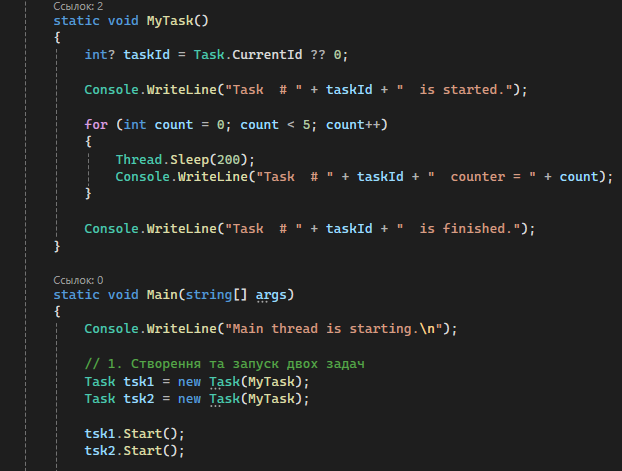
* 1. Розпаралелювання задач методом Invoke().



Результат виконання:



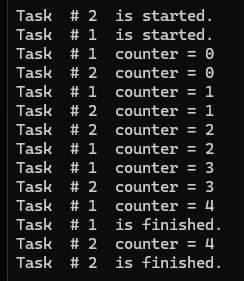
1. Створити програму, що створює дві задачі, які виконуються паралельно. Затримку методом Sleep() організувати на величину 200мс та пропорційно ідентифікатору задачі.



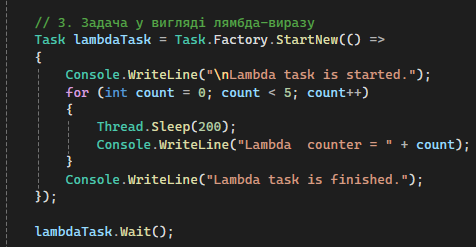
1. Організувати очікування виконання задач методом WaitAll().



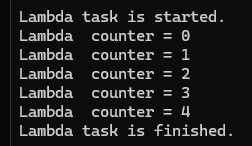
Результат виконання:



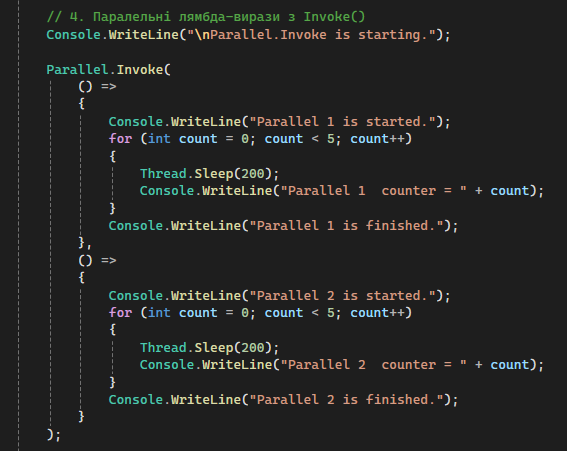
1. Визначити задачу для виконання у вигляді лямбда-виразу.



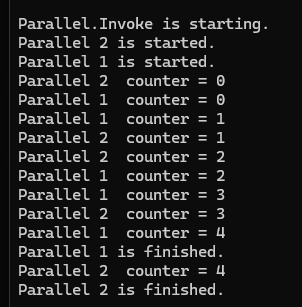
Результат виконання:



1. Створити програму паралельних обчислень за допомогою виклику методу Invoke(), де в якості аргументів застосовуються лямбда-вирази



Результат виконання:



Висновок: у ході виконання цієї лабораторної роботи я працював з паралельним виконанням задач використовуючи клас Task.