# OOR Ćwiczenie 1 – Wielowątkowość

Autor: Kornel Ostrowski

Grupa: 125NCI\_B

Link do projektu: <https://github.com/ostrowskik/Wielowatkowosc_oor_cw1>

Język programowania: C#

**Zadania do zrealizowania:**

1. Za pomocą wybranego przez siebie języka programowania zademonstruj tworzenie i synchronizowanie wątków.
2. Zaprezentuj użycie metod run() i join() związanych z wątkami.
3. Zaprezentuj użycie metod acquire() i release() związanych z blokadami (Lock).

**Przykładowe pytania:**

* Co to jest program współbieżny? Omów na przykładzie.

Jest to program wykonywany w ten sposób, że jego wątki są przełączane i wykonywane sekwencyjnie, na zmianę, w małych odstępach czasu

(np. sprawdzanie poprawnej pisowni w edytorach tekstowych)

* Co to jest program równoległy? Omów na przykładzie.

Jest to program, którego wątki są wykonywane w tym samym czasie przez inne rdzenie procesora (fizyczne lub logiczne). Przykładem jest gra komputerowa, w której jeden rdzeń może być zajęty fizyką, a drugi sztuczną inteligencją.

* Co to jest program rozproszony? Omów na przykładzie.

Jest to program, którego wykonywanie podzielone jest na osobne zadania, a każde zadanie (lub jego część) może być wykonywane przez inny komputer w sieci. Często jeden z komputerów realizuje podział zadań między podpiętymi pozostałymi komputerami. Przykładem może być poszukiwanie liczb pierwszych w chmurze, gdzie każdy z komputerów dostaje inne liczby do sprawdzenia.

* Co to jest przeplot w programowaniu współbieżnym?

Jest to przełączanie między jednocześnie wykonywanymi wątkami.

* Co to jest proces?

Egzemplarz wykonywanego programu, posiadający przydzieloną pamięć i zasoby. Pojedyńcza aplikacja może mieć wiele procesów.

* Co to jest wątek?

Część programu wykonywana współbieżnie w obrębie jednego procesu.

* Wymień podobieństwa i różnice między procesem a wątkiem.

Podobieństwa: przypisywany czas procesora. Różnice: każdy proces ma oddzielne zasoby i pamięć adresową.

* Podaj zalety i wady komunikacji międzywątkowej oraz międzyprocesorowej.

Międzywątkowa: pozwala na używanie tych samych zasobów, ale może wywoływać błędy, jeśli zasoby są użytkowane jednocześnie przez dwa różne wątki. Międzyprocesowa: separuje używane zasoby, ale zajmuje więcej pamięci, a synchronizacja jest bardziej skomplikowana.

* Scharakteryzuj metody run(), join(), acquire(), release().

Run() – służy do uruchamiania wątka.

Join() – czeka na zakończenie innych wątków, „zawiesza” wątek w celu synchronizacji.

Acquire() – zajmuje pewien zasób tylko dla wybranego wątku.

Release() – zwalnia zasób pozwalając wykorzystać go przez inne programy.

# Realizacja zadań

Tworzymy wątki w oparciu o klasę obiektu Thread, aby rozpocząć wykonywanie wątku używamy do tego metody Start(). Każdemu z wątków przydzieliliśmy nazwę Thread(nr wątku). Dla synchronizacji ich pracy stosujemy metodę Join(), która powoduje oczekiwanie na wykonanie wątku wywołanego tą metodą.

Aby zaprezentować stosowanie funkcji lock() umieściłem osobne wywołanie 10 wątków z pętli for(), które są kolejno wywoływane do odpowiedzi, blokując wywołanie kolejnych wątków w tak krótkim czasie. Wątki te zwracają ilość milisekund od momentu włączenia systemu operacyjnego.

Przykładowe wywołanie aplikacji:

