

## 1. Monitor czasu przy komputerze

### Cel programu:

Napisz program, który pozwala użytkownikowi mierzyć czas spędzony przy komputerze.

Każda sesja (od momentu wpisania `start` do `stop`) ma zostać zapisana w pliku tekstowym z datą, godziną i długością trwania.

---

### Wymagania:

Menu programu:

Po uruchomieniu programu wyświetli menu:

1. Rozpocznij nową sesję
2. Zakończ sesję
3. Pokaż historię sesji
4. Wyjdź

1. Pobierz wybór użytkownika przy użyciu `Scanner`.

2. Rozpoczęcie sesji:

- Zapisz czas rozpoczęcia w zmiennej typu `LocalDateTime`.
- Jeśli użytkownik spróbuje rozpoczęć sesję, gdy już jedna trwa — rzuć wyjątek (`IllegalStateException`).

3. Zakończenie sesji:

- Oblicz długość trwania sesji w minutach za pomocą `Duration.between(start, end).toMinutes()`.

Zapisz dane do pliku `sesje.txt` w formacie:

[2025-11-13 18:45] Sesja trwała: 95 minut

- 
- Jeśli nie rozpoczęto wcześniej sesji → rzuć wyjątek.

4. Wyświetlenie historii sesji:

- Odczytaj zawartość pliku `sesje.txt`.

Jeśli plik nie istnieje — utwórz go i wyświetl informację:

`Brak zapisanych sesji.`

○

## 5. Obsługa wyjątków:

- Użyj `try/catch/finally` dla operacji na plikach i błędnych wyborów w menu.

W bloku `finally` wyświetl komunikat:

`Operacja zakończona.`

○

---

## Wymagane biblioteki i funkcje:

Biblioteki:

```
import java.io.*;                                // FileReader, FileWriter,  
BufferedReader, IOException  
import java.time.*;                             // LocalDateTime, Duration  
import java.time.format.DateTimeFormatter; // formatowanie daty  
import java.util.Scanner;                        // pobieranie danych od  
użytkownika
```

Funkcje/metody:

- `Scanner.nextLine()` – wczytywanie tekstu
- `LocalDateTime.now()` – aktualna data i godzina
- `Duration.between(start, end).toMinutes()` – obliczanie różnicy czasu
- `FileWriter("sesje.txt", true)` – zapis do pliku
- `BufferedReader.readLine()` – odczyt linii z pliku
- `DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm")` – formatowanie daty