

Zaawansowanie programowanie obiektowe

Lista 8: Programowanie współbieżne

Karol Kulinowski, Agnieszka Kazimierska

9 stycznia 2024

Termin składania rozwiązań: 30.01

Programy przeznaczone do oceny należy umieszczać w specjalnie utworzonym repozytorium na portalu [GitHub](#). Programy należy oddawać w formie plików źródłowych z rozszerzeniem .java, bez pakowania. Podczas rozwiązywania zadań:

- Programy powinny zawierać komentarze dokumentacyjne i kontrolę wyjątków (jeśli taka jest potrzebna).
- Klasy powinny przestrzegać zasady hermetyzacji (enkapsulacji).
- Pamiętaj o **krótkich** komentarzach do commitów.

Zadanie 1: Implementacja interfejsu Runnable

Stwórz program, który używa interfejsu Runnable do zdefiniowania zadania, a następnie utwórz wiele wątków do jednoczesnego jego wykonania. Zadanie może polegać na czymś takim jak drukowanie liczb od 1 do 10. Każdy wątek powinien drukować unikalny zestaw liczb.

Zadanie 2: Synchronizacja wątków

Rozszerz program z poprzedniego zadania, dodając wspólny zasób (np. licznik), który będzie inkrementowany przez wiele wątków. Zaimplementuj synchronizację, aby zapewnić poprawne aktualizowanie licznika bez rywalizacji.

Zadanie 3: Bankowość

Zaprojektuj klasę reprezentującą bank (np. `SharedBank`) posiadającą pole `balance` oraz metodę `withdraw`, która będzie pobierać środki z konta (zmieniać wartość pola `balance`).

```
class SharedBank {
    ...
    public void withdraw(int amount) {
        int currentBalance = ...
        if (amount <= currentBalance) {
            // Simulate some time-consuming
operation
            try {
                Thread.sleep(10); // ms
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            ...
        } else {
            ...
        }
    }
}

class WithdrawThread implements Runnable { ... }

/**
 * SharedBank class test
 */
public class SharedBankDemo {
    public static void main(String[] args) { ... }
}
```

Niech klasa demo tworzy dwa (lub więcej) wątki zmieniające wartość zmiennej `balance` poprzez metodę `withdraw(int)`. Napisz go w dwóch wersjach:

- niezawierającym synchronizacji,
- zawierającym synchronizację.