

## Lab 02

### Komunikacja i synchronizacja między wątkami.

W tym labolatorium utworzyłem klasę **MarketClass**, której działanie przypomina wykorzystywanie karty bankowej. Posiada dwie metody – **deposit** oraz **withdraw**.

Na początku programu tworzę dwa wątki – **wife** i **husband**. Generują one ruch w klasie, wpłacając i wypłacając różne wartości z konta.

Metody klasy ustawiłem jako synchroniczne, aby dostęp do globalnej ilości środków konta był pojedynczy.

```
Current savings: 10000
wife withdrew 235, updated balance: 9765
husband withdrew 11, updated balance: 9754
husband deposited 617, updated balance: 10371
wife deposited 791, updated balance: 11162
husband withdrew 707, updated balance: 10455
husband deposited 184, updated balance: 10639
wife withdrew 277, updated balance: 10362
wife deposited 135, updated balance: 10497
```

Następnie dokonałem usunięcia klauzuli **synchronized** z metod, co spowodowało dowolny dostęp wątków do kwoty na koncie.

```
Current savings: 10000
husband withdrew 1011, updated balance: 8279
wife withdrew 710, updated balance: 8279
husband deposited 541, updated balance: 8820
wife deposited 603, updated balance: 8820
husband withdrew 749, updated balance: 8663
wife withdrew 157, updated balance: 8663
wife deposited 893, updated balance: 10021
husband deposited 465, updated balance: 10021
```

Już od samego początku widać, że kolejność dostępu do konta została zakłócona, przez co niektóre operacje nie były respektowane przy zmianie ilości środków na koncie. Tłumaczy to idealnie istotę synchronizowania wątków, umożliwiając ograniczenie dostępu do zasobu tylko jednej instancji naraz.

Przy dłuższym działaniu programu widać rozbieżność w otrzymywanych wynikach – nawet po wprowadzeniu wyższych możliwych kwot do wypłacenia bardzo często program bez synchronizacji wykazuje wzrost środków na koncie, nawet przy bardzo częstym pomijaniu wpłat. Obrazuje to jak ważnym elementem jest zrozumienie funkcjonowania pisanego przez nas programu oraz możliwości wybrania rozwiązania zachowującego najbardziej realistyczne, a przez to pożądane zachowanie systemu.