

Laboratorium 2 - semafor

- Semafor to mechanizm synchronizacji procesów zaproponowany przez Dijkstrę.
- Semafor jest zmienną całkowitą, która przyjmuje wartości nieujemne (≥ 0) lub — w przypadku semaforów binarnych — logiczne. Zmienna semaforowa musi mieć nadaną początkową wartość (nieujemną).
- Po nadaniu początkowej wartości zmiennej semaforowej można na niej wykonywać tylko dwa rodzaje operacji:
 1. **P** — **opuszczanie semafora (hol. proberen)**, powoduje zmniejszenie wartości zmiennej semaforowej,
 2. **V** — **podnoszenie semafora (hol. verhogen)**, powoduje zwiększanie wartości zmiennej semaforowej.
- Wykonując operację semaforową, proces może zastać zablokowany (przejsć w stan oczekiwania). Typowym przypadkiem jest blokowanie w operacji opuszczania semafora. Operacja opuszczania nie zakończy się do czasu, aż wartość zmiennej semaforowej będzie na tyle duża, że zmniejszenie jej wartości w wyniku tej operacji nie spowoduje przyjęcia wartości ujemnej.
- Rodzaje semaforów
 1. **semafor binarny** ma dwa stany: true (podniesiony otwarty) i false (opuszczony zamknięty). Wielokrotne podnoszenie takiego semafora nie zmieni jego stanu — skutkiem będzie stan otwarcia. W niektórych rozwiązaniach przyjmuje się, że próba podniesienia otwartego semafora sygnalizowana jest błędem.
 2. **semafor ogólny** „pamięta” liczbę operacji podniesienia. Zwykle inicjalizowany jest ilością dostępnego zasobu. Można bez blokowania procesu wykonać tyle operacji opuszczenia semafora, aby jego wartość była nieujemna.
- Zadanie
 - zaimplementować [semafor binarny](#) za pomocą metod wait i notify/notifyall, użyć go do synchronizacji wycigu z poprzedniego laboratorium.
 - zaimplementować [semafor licznikowy \(ogólny\)](#) za pomocą metod wait i notify/notifyall. Przetestować semafor na prostej symulacji sklepu samoobsługowego z ograniczoną ilością koszyków.

Bibliografia

1. Z Weiss, T Gruzlewski "Programowanie współbieżne i rozproszone w przykładach i zadaniach"
2. Jacek Rumiki, Język Java. [Rozdział o watach](#)
3. Bill Venners, [Inside the Java Virtual Machine](#) (rozdz. 5, *The Java Virtual Machine*), McGraw-Hill Companies; 2nd Bk&Cdr edition, 2000.

Katarzyna Rycerz, kzajac@agh.edu.pl

(także na podstawie opracowań dr B. Balisia)