

Zaawansowane środowiska programistyczne

Ćwiczenie 4

Bazując na programie z ćwiczenia 3 napisać program wielowątkowy zgodnie z następującymi założeniami:

1. W programie występuje tylko jeden obiekt klasy *Professor* i kilka obiektów klasy *Student*
2. Studenci przychodzą na konsultacje do profesora. Jeśli profesor nie udziela konsultacji żadnemu studentowi może od razu przyjąć studenta, w przeciwnym razie student zapisuje się do kolejki FIFO i jest obsługiwany zgodnie z kolejnością zgłoszenia, gdy profesor zakończy konsultacje z poprzednim studentem.
3. Dany student może być w danej chwili w jednym miejscu w kolejce.
4. Zadania profesora i studenta podczas konsultacji są realizowane w oddzielnych wątkach.
5. W klasach *Professor* i *Student* znajdują się odpowiednie metody *receiveStudents()* i *consultProfessor()* wywołujące odpowiednie wątki.
6. Czas trwania konsultacji jest odwrotnie proporcjonalny do wiedzy studenta i jest symulowany przez uśpienie wątku.
7. Po zakończeniu konsultacji wiedza studenta zwiększa się o 1.
8. Wątek studenta (*StudentConsultationThread*) kolejno: rezerwuje miejsce w kolejce, czeka na potwierdzenie od profesora o możliwości wejścia do gabinetu (usunięciu z kolejki), odbiera uzyskaną w czasie konsultacji wiedzę i kończy swe działanie.
9. Wątek profesora (*ProfessorConsultationThread*) kolejno: czeka na pojawienie się studenta w kolejce i usuwa go z kolejki, przekazuje studentowi wiedzę w czasie zależnym od poziomu wcześniejszej wiedzy, czeka aż student odbierze wiedzę i przyjmuje kolejnego studenta z kolejki lub czeka na jego pojawienie się.

Szczególną uwagę podczas pisania programu należy zwrócić na problem synchronizacji wątków np. zabezpieczenie kolejki FIFO przed jednoczesną modyfikacją przez kilka wątków naraz

Przetestować na dowolnym przykładzie wykorzystując obiekty poszczególnych klas i zaimplementowane metody.