Nazwa Wydziału	Wydział Matematyki i Informatyki
Nazwa jednostki prowadzącej moduł	Instytut Informatyki i Matematyki Komputerowej
Nazwa modułu kształcenia	
Kod modułu	
Język kształcenia	Polski
Efekty kształcenia dla modułu kształcenia	Wiedza • Student zna i rozumie problemy i zagrożenia dotyczące programowania współbieżnego • Student zna różne modele współbieżności • Student zna mechanizmy synchronizacji procesów i zadań • Student zna pojęcia semafora, monitora, zmiennej warunkowej, bariery cyklicznej, zadań asynchronicznych • Student zna pojęcia pamięci współdzielonej, pamięci podręcznej • Student zna algorytmy sprzętowego i algorytmicznego rozwiązywania problemu wzajemnego wykluczania Umiejętności • Student potrafi zaprojektować i napisać system współbieżny w językach C++, JAVA, C#, ADA • Student potrafi zaprojektować i napisać prosty algorytm wykorzystujący moc obliczeniową kart graficznych • Student potrafi poprawnie używać podstawowych mechanizmów synchronizujących w wybranych językach wysokiego poziomu • Student potrafi skorzystać z gotowych API do zrównoleglania prostych zadań (OpenMP, TPL) Kompetencje społeczne • Student rozumie zagrożenia jakie niesie współbieżność dla bezpieczeństwa systemu informatycznego
Typ modułu kształcenia (obowiązkowy/fakultatywny)	Fakultatywny dla następujących specjalności studiów I stopnia: 1. informatyka stosowana 2. inżynieria oprogramowania 3. modelowanie, sztuczna inteligencja i sterowanie 4. matematyka komputerowa

Rok studiów	3 I stopnia, dowolny II stopnia
Semestr	6
Imię i nazwisko osoby/osób prowadzących moduł	Wykład: Daniel Wilczak Laboratorium: P. Kalita, B. Zieliński
Imię i nazwisko osoby/osób egzaminującej/egzaminujących bądź udzielającej zaliczenia, w przypadku gdy nie jest to osoba prowadząca dany moduł	Daniel Wilczak
Sposób realizacji	Wykład, laboratorium
Wymagania wstępne i dodatkowe	Programowanie 1, Programowanie 2, Systemy Operacyjne
Rodzaj i liczba godzin zajęć dydaktycznych wymagających bezpośredniego udziału nauczyciela akademickiego i studentów, gdy w danym module przewidziane są takie zajęcia	Wykład, laboratorium Wykład: 30 Laboratorium: 30 **Łącznie: 60**
Liczba punktów ECTS przypisana modułowi	6
Bilans punktów ECTS	Udział w wykładach - 30 godz. Udział w zajęciach laboratoryjnych – 30 godz. Samodzielna implementacja zadań programistycznych – 90 godz. Łączny nakład pracy studenta: 180 godzin , co odpowiada 6 punktom ECTS
Stosowane metody dydaktyczne	1. Wykład ilustrowany prezentacją komputerową i analizą przykładów. 2. Ćwiczenia w laboratorium komputerowym, połączone z dyskusją przy tablicy. 3. Samodzielne implementacja zadań programistycznych.
Metody sprawdzania i kryteria oceny efektów kształcenia uzyskanych przez studentów	1. Programistyczne zadania domowe (E1, E2 E3,E4, E5) 2. Automatycznie weryfikowane programy zaliczeniowe (E1, E2, E3, E4, E5) 3. Egzamin (E1, E4, E6, E7) 4. Prezentowanie rozwiązań zadań domowych oraz ustna obrona rozwiązań zadań programistycznych wysłanych do serwera automatycznej weryfikacji (E6, E7) 5. Wykorzystanie systemu antyplagiatowego na serwerze weryfikującym zadania

	programistyczne (E7)
Forma i warunki zaliczenia modułu, w tym zasady dopuszczenia do egzaminu, zaliczenia, a także forma i warunki zaliczenia poszczególnych zajęć wchodzących w zakres danego modułu	Student otrzymuje ocenę końcowa z ćwiczeń na podstawie punktów przyznawanych za systematycznie oddawane zadania domowe. Warunkiem otrzymania zaliczenia ćwiczeń jest uzyskanie łącznie co najmniej 50% możliwych do zdobycia punktów. Do egzaminu końcowego są dopuszczani jedynie studenci, którzy uzyskali zaliczenie ćwiczeń na ocenę pozytywną. Ocena końcowa z kursu jest wyznaczana na podstawie wyniku egzaminu według kryteriów przyjętych i ogłoszonych studentom przez wykładowcę.
Treści modułu kształcenia	1. Podstawowe idee programowania współbieżnego: współbieżność, równoległość, rozproszoność. 2. Pojęcia pierwotne programowania współbieżnego: procesy i wątki, ich implementacja w systemach Windows i Unix/Linux. 3. Przykłady API korzystającego z mechanizmów procesów i wątków: procesy w C, wątki w C (na przykładzie API POSIX), wątki Java, wątki C#. 4. Komunikacja i synchronizacja w programowaniu współbieżnym: pamięć współdzielona, race condition i problem wzajemnego wykluczania, sekcje krytyczne. 5. Rozwiązania problemu wzajemnego wykluczania: podejście algorytmiczne, podejście sprzętowe. 6. Podstawowe struktury danych służące do synchronizacji i ich wzajemne zależności: mutex, 7. semafor, zmienna warunkowa (condition), monitor. 8. Przykładowe API służące do synchronizacji: nieobiektowe API dla C (np POSIX), przykład obiektowego API dla C++, JAVA, C#. 9. Mechanizm rendezvous, przykład dla tasków w języku Ada. 10. Model PRAM i jego podstawowe techniki algorytmiczne. 11. Obliczenia współbieżne z wykorzystaniem procesorów graficznych – standard OpenCL orazjęzyk CUDA. 12. Klasyczne problemy synchronizacji, ich

	warianty i rozwiązania z wykorzystaniem różnych mechanizmów: problem producenta i konsumenta, problem 5 filozofów, problem czytelników i pisarzy. 13. Przykłady mniej klasycznych problemów synchronizacji. 14. Problem szeregowania w przydziale zasobów: klasyczne algorytmy szeregowania, przykładowa implementacja rozwiązania problemu szeregowania, formalna definicja problemu szeregowania jako problemu optymalizacji dyskretnej, elementy teorii szeregowania. 15. Metody formalne analizy i dowodzenia poprawności programów współbieżnych. 16. Zakleszczenia i metody radzenia sobie z nimi: podejście zapobiegania zakleszczeniom, podejście unikania zakleszczeń. 17. Współbieżność a bazy danych: tranzakcje i blokady. 18. Przykłady algorytmów współbieżnych (współbieżne sortowanie mergesort i quicksort).
Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu	1. A. Silberschatz, P.B. Galvin, G. Gagne, Podstawy systemów operacyjnych (wyd. 7), WNT, Warszawa 2006. 2. W. R. Stevens, UNIX programowanie usług sieciowych (tom 1,2), WNT, Warszawa (tom 1) 2001, (tom 2) 2002. 3. M. Ben-Ari, Principles of Concurrent and Distributed Programming (2nd Edition), Prentice-Hall, 2006 (wyd. polskie: Podstawy programowania współbieżnego i rozproszonego, WNT, Warszawa 1996). 4. Z. Weiss, T. Gruźlewski, Programowanie współbieżne i rozproszone w przykładach i zadaniach, WNT, Warszawa 1993. 5. Z. Huzar, Z. Fryźlewicz, I. Dubielewicz, B. Hnatk, Ada 95, Helion 1998. 6. B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea, Java. Współbieżność dla praktyków, Helion 2007.
Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program	Nie dotyczy

kształcenia przewiduje praktyki	