

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Programowanie rozproszone</b>		Kod <b>1010332511010335196</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>1 / 1</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: <b>-</b> Laboratoria: <b>15</b> Projekty/seminaria: <b>15</b>		Liczba punktów <b>6</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>(brak)</b>		(ogólnouczelniany, z innego kierunku) <b>(brak)</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b> <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>6 100%</b> <b>6 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Adam Meissner email: Adam.Meissner@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		dr inż. Krzysztof Zwierzyński email: Krzysztof.Zwierzynski@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych algorytmów i ich analizy, technik projektowania algorytmów, abstrakcyjnych struktur danych i ich implementacji, problemów obliczeniowo trudnych; ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie architektury systemów komputerowych, zasad działania systemów operacyjnych i ich rodzajów; zna typowe informatyczne technologie inżynierskie.
2	<b>Umiejętności:</b>	Student posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się, a także czytania ze wskazanej literatury przedmiotowej; potrafi konstruować algorytmy z wykorzystaniem podstawowych technik algorytmicznych i dokonać analizy ich złożoności; potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego.
3	<b>Kompetencje społeczne</b>	Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.
<b>Cel przedmiotu:</b> przedstawienie podstawowych modeli systemów rozproszonych oraz najważniejszych metod synchronizacji i komunikacji w takich systemach; prezentacja wybranych problemów budowy systemów rozproszonych.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie technologii sieciowych - [K_W07] 2. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie technologii internetowych - [K_W11]		
<b>Umiejętności:</b>		
1. Student potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów - [K_U02] 2. Student potrafi planować i przeprowadzać eksperymenty, wykorzystywać poznane metody i modele matematyczne, a także symulacje komputerowe do testowania, analizy i oceny działania systemów informatycznych i ich składowych - [K_U07] 3. Student potrafi dokonać krytycznej analizy sposobu funkcjonowania sprzętu komputerowego, systemu operacyjnego (lub ich fragmentów) i sieci komputerowych - [K_U11]		
<b>Kompetencje społeczne:</b>		

1. Student ma świadomość roli społecznej absolwenta uczelni techn., rozumie potrzebę przekazywania społeczeństwu - m.in. przez środki masowego przekazu - informacji dotyczących informatyki; podejmuje starania, by przekazać informacje w sposób zrozumiały - [K\_K06]
2. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K\_K07]

### Sposoby sprawdzenia efektów kształcenia

Wykład. Egzamin pisemny obejmujący pytania teoretyczne oraz proste zadania z zakresu programowania rozproszonego.  
 Laboratorium. Ustne lub pisemne sprawdzenie wiadomości niezbędnych do wykonania bieżących ćwiczeń, przygotowanie i terminowe składanie sprawozdań, ocena aktywności uczestników zajęć.  
 Projekt. Terminowa realizacja kolejnych etapów zadania projektowego. Przygotowanie dokumentacji.  
 Warunki zaliczenia wykładu, laboratorium i projektu: należy uzyskać co najmniej 50,1% całkowitej liczby punktów.

### Treści programowe

Wykład. Programowanie rozproszone a programowanie współbieżne, rozproszony model programu współbieżnego, przezroczystość sieciowa, model klient-serwer, standard MPI, środowisko Open CL, metody synchronizacji wątków i procesów, miary wydajności systemów rozproszonych, projektowanie algorytmów rozproszonych, elementy programowania w modelu klient-serwer, problemy zawodności i bezpieczeństwa w systemach rozproszonych, programowanie rozproszone w metodyce programowania wieloparadygmatowego.

Laboratorium. Programowanie rozproszone przy użyciu biblioteki MPI. Rozpraszenie obliczeń z wykorzystaniem technologii GPGPU. Programowanie rozproszone jako wariant programowania wieloparadygmatowego w środowisku Mozart/Oz. Kolejowanie zadań w systemach superkomputerowych (zadanie opcjonalne).

Projekt. Realizacja zadania projektowego ilustrującego możliwości rozpraszania obliczeń za pomocą wybranej platformy programistyczno-sprzętowej.

#### Literatura podstawowa:

1. Programowanie współbieżne i rozproszone, Weiss Z., Gruźlewski T., Wyd. Naukowo-Techniczne, Warszawa, 1993
2. Programowanie. Konceptcje, techniki i modele, Roy P. van, Haridi S., Wyd. Helion, Gliwice, 2005
3. Systemy rozproszone. Zasady i paradygmaty, Tanenbaum A.S., Steen M. van, Wyd. Naukowo-Techniczne, Warszawa, 2006

#### Literatura uzupełniająca:

1. Sztuka programowania wieloprocesorowego, Herlihy M., Shavit N., PWN, Warszawa, 2008
2. Introduction to Parallel Computing, Barney B., [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)
3. A User's Guide to MPI, Pacheco P.S., [http://www.wellesley.edu/CS/courses/CS331/notes/mpl\\_guide.pdf](http://www.wellesley.edu/CS/courses/CS331/notes/mpl_guide.pdf)
4. Ericsson AB, Erlang/OTP System Documentation, <http://erlang.org/doc/pdf/otp-systemdocumentation.pdf>

### Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. Wykłady	30
2. Laboratoria	15
3. Projekt	15
4. Konsultacje i egzamin	15
5. Przygotowanie do ćwiczeń lab., wykonanie sprawozdań	15
6. Wykonanie projektu	30
7. Przygotowanie do egzaminu	30

### Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	150	6
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	75	3
Zajęcia o charakterze praktycznym	75	3