

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Narzędzia przetwarzania rozproszonego</b>		Kod <b>1010512311010514017</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 1</b>
Ścieżka obieralności/specjalność <b>Systemy rozproszone</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: - Laboratoria: <b>30</b> Projekty/seminaria: -		Liczba punktów <b>5</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>5 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
<p>dr inż. Dariusz Wawrzyniak email: Dariusz.Wawrzyniak@cs.put.poznan.pl tel. 61 6652963 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_W1-2, K_W4, K_W6-15, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
2	<b>Umiejętności:</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
3	<b>Kompetencje społeczne</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
<p>1. Przekazanie studentom podstawowej wiedzy z zakresu narzędzi przetwarzania rozproszonego oraz współtworzących je usługi i mechanizmów, a także związanych z nimi technik i paradygmatów programowania rozproszonego</p> <p>2. Rozwijanie u studentów umiejętności wyboru właściwego podejścia do rozwiązywania problemów przetwarzania rozproszonego i budowy systemów rozproszonych</p>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, - [K_W4]</p> <p>2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: przetwarzanie współbieżne w tym wielowątkowe, przetwarzanie rozproszone, komunikacja i synchronizacja procesów/wątków, - [K_W5]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, związanych z komunikacją sieciową oraz niezawodnością systemów, - [K_W6]</p> <p>4. ma podstawową wiedzę o cyklu życia systemów informatycznych programowych, - [K_W7]</p> <p>5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z obszaru przetwarzania rozproszonego, - [K_W8]</p>		
<b>Umiejętności:</b>		

<ol style="list-style-type: none"><li>1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, - [K_U1]</li><li>2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, - [K_U5]</li><li>3. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych, - [K_U7]</li><li>4. potrafi planować i przeprowadzać eksperymenty, weryfikujące określone własności systemu, interpretować uzyskane wyniki i wyciągać wnioski, - [K_U8]</li><li>5. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody symulacyjne oraz eksperymentalne, - [K_U9]</li><li>6. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne, - [K_U10]</li><li>7. potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi, - [K_U12]</li><li>8. potrafi ocenić przydatność i możliwość wykorzystania nowych metod i narzędzi oraz nowych produktów informatycznych, - [K_U13]</li><li>9. potrafi ocenić złożoność obliczeniową i komunikacyjną algorytmów i problemów, - [K_U16]</li><li>10. potrafi ocenić architekturę oprogramowania z punktu widzenia wymagań pozafunkcjonalnych, - [K_U18]</li><li>11. potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych, - [K_U21]</li><li>12. potrafi - stosując m.in. koncepcyjnie nowe metody - rozwiązywać złożone zadania informatyczne, w tym zadania nietypowe oraz zadania zawierające komponent badawczy, - [K_U25]</li><li>13. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia, - [K_U24]</li><li>14. potrafi wybrać język programowania odpowiedni do danego zadania programistycznego, - [K_U26]</li><li>15. potrafi - zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne - zaprojektować złożone urządzenie, system informatyczny lub proces oraz zrealizować ten projekt - co najmniej w części. - [K_U27]</li></ol>
<b>Kompetencje społeczne:</b>
<ol style="list-style-type: none"><li>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, - [K_K1]</li><li>2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życie, - [K_K4]</li><li>3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania, - [K_K6]</li></ol>

### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:
  - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów / ćwiczeń:
  - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze problemowym składającym się z 4 pytań otwartych, nie znanych wcześniej, punktowanych w skali od 0 do 25, co daje maksymalnie 100 pkt. za cały egzamin, przy czym próg na ocenę pozytywną wynosi 50 pkt.,
  - na żądanie studenta omówienie wyników egzaminu,
- b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez:
  - ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
  - ocenę postępu i terminowości realizacji zadań projektowych, rejestrowanych w systemie zarządzania projektem,
  - ocenę jakości zrealizowanych projektów, poprzedzoną demonstracją i obroną przez studenta,

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu oraz planowania lub interpretacji wyników eksperymentów,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

### Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Podstawowe pojęcia i koncepcje: system sieciowy, system rozproszony, warstwa pośrednia (middleware). Zagadnienia projektowe: transparentność, otwartość, interoperacyjność, skalowalność itp. Klasyfikacja mechanizmów komunikacji międzyprocesowej: synchroniczne i asynchroniczne, trwałe i przejściowe. Środowiska wymiany komunikatów: porównanie PVM i MPI. Zdalne wywoływanie procedur: zagadnienia projektowe, budowa protokołu, semantyka awarii, idempotentność i bezstanowość, przykład realizacji - Sun RPC. Współbieżność w środowisku Java: podejścia do realizacji przetwarzania współbieżnego, wielowątkowość, mechanizmy synchronizacji oparte na obiektach klasy Object, pakiet java.util.concurrent. Podejście obiektowe do budowy systemów rozproszonych: klasyfikacja obiektów w przetwarzaniu rozproszonym, definicja interfejsu, dziedziczenie, interfejsy jako parametry, tryby przekazywania parametrów. Zdalne wywoływanie metod w środowisku Java - RMI: tworzenie i udostępnianie zdalnych obiektów, przekazywanie obiektów przez parametry, obsługa rejestru zdalnych obiektów, dynamiczna aktywacja obiektów. Systemy klasy MOM (ang. message-oriented middleware): koncepcja MOM, paradygmat kolejowania oraz publikowania-subskrypcji, protokół AMQP, przykład realizacji - RabbitMQ. Przestrzeń krotek: koncepcja przestrzeni krotek oparta na modelu Linda, adresowanie asocjacyjne, specyfikacja JavaSpaces, przykład realizacji - Apache River. Środowiska Ada: podstawy programowania (typy danych, konstrukcje programotwórcze, pakiety, obiektowość), współbieżność (zadania, spotkania, obiekty chronione), przetwarzanie rozproszone oparte na Aneksie E specyfikacji (klasy jednostek kompilacji, język opisu systemu).

Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium wyposażony w stacje robocze pracujące pod kontrolą systemu operacyjnego Linux. Ćwiczenia realizowane są indywidualnie przez studentów, a w uzasadnionych przypadkach przez zespoły 2-osobowe. Program laboratorium obejmuje następujące zagadnienia:

Współbieżność na poziomie systemu operacyjnego: wątki w standardzie POSIX i ich synchronizacja, koncepcja monitora. Realizacja rozproszonego monitora w oparciu o środowisko komunikacyjne PVM/MPI. Sun RPC: testowanie semantyki błędów na przykładzie prostej usługi zdalnego licznika, realizacja złożonej usługi sieciowej wymagającej współbieżnej obsługi zleceń oraz wywołania zwrotnego. Java RMI: porównanie realizacji usługi zdalnego licznika w podejściu obiektowym oraz proceduralnym, realizacja zdalnego bufora. Systemy klasy MOM jako uogólnienie zdalnego bufora: testowanie i porównanie paradygmatu kolejowania oraz publikowania-subskrypcji. Realizacja prostej usługi na bazie zdalnych zwiokrotnionych obiektów udostępnionych przez RMI, z koordynacją replik realizowaną za pomocą systemu klasy MOM. JavaSpaces: przykład wykorzystania adresowania asocjacyjnego do kojarzenia par w grze w szachy. Podstawy programowania w języku Ada na przykładzie problemu sortowania tablicy. Współbieżność w środowisku Ada: tworzenie zadań, synchronizacja zadań poprzez spotkania, obiekty chronione, przykłady realizacji wybranych mechanizmów synchronizacji (np. semaforów) i implementacji rozwiązań klasycznych problemów synchronizacji. Realizacja przetwarzania rozproszonego w modelu zdefiniowanym przez Aneks E: przykłady użycia wybranych klas jednostek kompilacji (RCI, RT, SP).

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, uzupełniona dyskusją nad problemami i ewentualnie ich ilustracją wraz z rozwiązaniami na tablicy;
2. ćwiczenia laboratoryjne: wykonywanie eksperymentów i wyciąganie wniosków odnośnie funkcjonowania analizowanych mechanizmów; dyskusja problemów i wypracowywanie koncepcji rozwiązania zadań, następnie praktyczna implementacja opracowanych koncepcji; w wybranych przypadkach pokaz multimedialny oraz demonstracja zrębów rozwiązań w celu dalszego ich rozwinięcia.

#### Literatura podstawowa:

1. Systemy rozproszone. Podstawy i projektowanie, Coulouris G., Dollimore J., Kindberg T., WNT, W-wa, 1999.
2. Systemy rozproszone. Zasady i paradygmaty, Tanenbaum A., WNT, W-wa, 2006.
3. Concurrent Systems. Operating Systems, Database and Distributed Systems: An Integrated Approach, Bacon J., Addison-Wesley, Harlow (England), 1998.

#### Literatura uzupełniająca:

1. Przetwarzanie rozproszone w systemie UNIX, Gabassi M, Dupouy B, Lupus, W-wa, 1995.
2. ActiveMQ in Action, Snyder B., Bosanac D., Davies R., Maining, 2009.
3. JavaSpaces Principles, Patterns, and Practice, Freeman E., Hupfer S., Arnold K., Addison-Wesley, 1999.
4. Ada 95, Huzar Z., Fryźlewicz Z., Dubielewicz I., Hnatk B, Helion, Gliwice, 2001.
5. Ada 2012, Barnes J., Cambridge University Press, 2014.

#### Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w wykładach	30
2. udział w zajęciach laboratoryjnych	30
3. przygotowanie do ćwiczeń laboratoryjnych	20
4. realizacja zadań laboratoryjnych (napisanie, uruchomienie i weryfikacja programów w czasie poza zajęciami laboratoryjnymi)	20
5. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności zadań laboratoryjnych (częściowo mogą być realizowane drogą elektroniczną)	3
6. przygotowanie do egzaminu i obecność na egzaminie: 18 godz. + 2 godz.	20

#### Obciążenie pracą studenta

<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	123	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	63	2
Zajęcia o charakterze praktycznym	70	3