

Politechnika Białostocka										
Kierunek studiów	Informatyka							Poziom i forma studiów	drugiego stopnia stacjonarne	
Specjalność / Ścieżka dyplomowania	Inteligentne Technologie Internetowe							Profil kształcenia	ogólnoakademicki	
Nazwa przedmiotu	Tworzenie aplikacji rozproszonych							Kod przedmiotu	INF2TAR	
								Rodzaj przedmiotu	obieralny	
Forma zajęć i liczba godzin	W	Ć	L	P	Ps	T	S	Semestr	2/3	
	30				15			Punkty ECTS	2	
Przedmioty wprowadzające										
Cele przedmiotu	Zapoznanie studentów ze specyfiką tworzenia aplikacji rozproszonych, wymagających współdzielenia zasobów, ze szczególnym uwzględnieniem technologii gridowych i chmurowych. Nauczenie studentów praktycznego stosowania wybranych modeli i technologii do projektowania, budowania i uruchamiania aplikacji rozproszonych i współdzielących zasoby.									
Treści programowe	Wykład: Problemy w tworzeniu systemów rozproszonych i we współdzieleniu zasobów. Dobre praktyki (wzorce) w tworzeniu systemów rozproszonych i współdzieleniu zasobów. Wybrane modele i technologie w tworzeniu systemów rozproszonych i współdzieleniu zasobów: bezpieczeństwo (PKI, certyfikaty X.509, CA), Condor, serwisy webowe, serwisy gridowe (WSRF), Globus Toolkit, przetwarzanie w chmurach (cloud computing): IaaS (Amazon Web Services, Eucalyptus), PaaS (Google App Engine), model obliczeniowy map-reduce (Apache Hadoop). Pracownia specjalistyczna: Tworzenie aplikacji rozproszonych z wykorzystaniem technologii: Condor, Google App Engine, Apache Hadoop, Apache Spark, Apache ZooKeeper.									
Metody dydaktyczne	programowanie z użyciem komputera, wykład informacyjny, wykład problemowy,									
Forma zaliczenia	Wykład: ocena z PS + sprawdzian z wiedzy teoretycznej. Pracownia specjalistyczna: ocena zadań praktycznych wykorzystujących wybrane modele technologie.									
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się		
EU1	wymienia i opisuje (w tym na przykładzie własnych aplikacji) modele programistyczne i obliczeniowe stosowane w tworzeniu aplikacji rozproszonych.							K_W06 K_W07		
EU2	wymienia i opisuje (w tym na przykładzie własnych aplikacji) wybrane technologie stosowane w tworzeniu aplikacji rozproszonych.							K_W06 K_W07		
EU3	tworzy aplikacje rozproszone korzystające z modelu SOA							K_U11 K_U14		
EU4	tworzy aplikacje rozproszone korzystające z zaawansowanych modeli obliczeniowych							K_U11 K_U14		
Symbol efektu uczenia się	Sposób weryfikacji efektu uczenia się							Forma zajęć na której zachodzi weryfikacja		
EU1	Wykład: Sprawdzian z wiedzy teoretycznej Ps: Pytania teoretyczne podczas oddawania programów							W, Ps		
EU2	Wykład: Sprawdzian z wiedzy teoretycznej Ps: Pytania teoretyczne podczas oddawania programów							W, Ps		
EU3	Ocena zadań programistycznych realizowanych w semestrze							Ps		
EU4	Ocena zadań programistycznych realizowanych w semestrze							Ps		
Bilans nakładu pracy studenta (w godzinach)								Liczba godz.		
Wyliczenie										
	1 - Udział w wykładach -							30		
	2 - Udział w pracowni specjalistycznej -							15		
	4 - Realizacja zadań programistycznych poza zajęciami -							10		
	5 - Udział w konsultacjach -							2		
	6 - Przygotowanie do zaliczenia -							3		
RAZEM:								60		
Wskaźniki ilościowe								GODZINY	ECTS	
Nakład pracy studenta związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela								47 (5)+(1)+(2)	1,6	
Nakład pracy studenta związany z zajęciami o charakterze praktycznym								25 (4)+(2)	0,8	
Literatura podstawowa	1. A.S. Tanenbaum, M. Van Steen: Systemy rozproszone : zasady i paradymaty, PWN, 2006. 2. Dokumentacja systemu Condor (www.cs.wisc.edu/condor/) 3. Dokumentacja systemu Globus (www.globus.org). 4. Dokumentacja systemu GAE (code.google.com/appengine). 5. Dokumentacja systemu AWS (aws.amazon.com). 6. Dokumentacja systemu Eucalyptus (www.eucalyptus.com). 7. Dokumentacja systemu Apache Hadoop (hadoop.apache.org). 8. Dokumentacja systemu Apache Spark (spark.apache.org/documentation.html).									
Literatura uzupełniająca	1. I. Foster, C. Kesselman, The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, Elsevier, 2004. 2. F. Buschmann, K. Henney, D.C. Schmidt: Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing. Wiley & Sons 2007. 3. D.C. Schmidt, M. Stal, H. Rohnert, F. Buschmann: Pattern-Oriented Software Architecture, Volume 3: Patterns for Concurrent and Networked Objects. Wiley & Sons, 2000. 4. M. Kircher, P. Jain: Pattern-Oriented Software Architecture, Volume 3: Patterns for Resource Management, Wiley & Sons 2004.									
Jednostka realizująca	Katedra Oprogramowania							Data opracowania programu		
Program opracował(a)	dr inż. Cezary Bóldak							5 kwietnia 2019		