

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Sieci neuronowe i al. genetyczne</i>	Liczba godz. wykładu: 15	Liczba godz. ćwic./lab.: 30
Cele przedmiotu: Studenci poznają podstawowe problemy i zasady działania prostych sieci neuronowych i algorytmów genetycznych.		
Zawartość programowa: <ol style="list-style-type: none"> 1. Model neuronu progowego. 2. Model perceptronu jednowarstwowego. 3. Perceptron wielowarstwow. 4. Prosty algorytm genetyczny wykorzystujący krzyżowanie, reprodukcję i mutację. 		
Literatura: J. Hertz, A. Krogh, R. G. Palmer „Wstęp do teorii obliczeń neuronowych”, WNT 1995.		
Forma prowadzonych zajęć: wykład + laboratorium		
Narzędzia programistyczne (zajęcia laboratoryjne): dowolny język programowania i system operacyjny.		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Przetwarzanie obrazów cyfrowych</i>	Liczba godz. wykładu: 30	Liczba godz. ćwicz./lab.: 30
Cele przedmiotu: Zaznajomienie z podstawowymi algorytmami stosowanymi w przetwarzaniu obrazów cyfrowych. Budowanie interfejsu graficznego.		
Zawartość programowa: Na laboratorium implementowane są podstawowe algorytmy przetwarzania obrazów: <ol style="list-style-type: none"> 1. Korekcja jasności, kontrastu i gamma obrazu cyfrowego. 2. Transformacje geometryczne obrazu - translacja, rotacje, skalowanie, pochylenie + interpolacja najbliższym sąsiadem i biliniowa. 3. Modele barwne – RGB, CMYK, HSL, CIE Lab, gray scale. 4. Histogram obrazu – wizualizacja i transformacje histogramu. 5. Maskowanie obrazu - definiowanie maski przez progowanie i rozrost regionu (magic wand), wizualizacja maski, przekształcenia obrazu z uwzględnieniem zdefiniowanej maski. 6. Transformacja obrazu za pomocą krzywych. 7. Filtry splotowe - dolny, górny, krawędziowy, filtr gaussa, filtr unsharp mask. 8. Filtry morfologiczne - mediana, minimum, maximum, despeckle. 9. Transformata fouriera i odwrotna transformata fouriera (DFT lub FFT), wizualizacja transformaty, filtracja w dziedzinie częstotliwości. 		
Literatura: <ol style="list-style-type: none"> 1. Tadeusiewicz R., Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Postępu Telekomunikacji, Kraków, 1997 2. Gonzalez R.C., Woods R.E., Digital Image Processing, Pearson Prentice Hall, 2008 3. Russ J.C., The Image Processing Handbook, CRC Press, 2007 4. Watkins C., Sadun A., Makarenka S., Nowoczesne metody przetwarzania obrazu, WNT, Warszawa, 1995 		
Forma prowadzonych zajęć: wykład + laboratorium		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość języka C++, znajomość podstaw grafiki komputerowej		
Narzędzia programistyczne (zajęcia laboratoryjne): Kompilator języka C++ oraz do wyboru biblioteki i środowisko: biblioteki Qt4, MS Visual C++ + biblioteki .NET, Borland C++ Builder 6, JAVA, MS Visual C#		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Teoretyczne podstawy informatyki</i>	Liczba godz. wykładu: 30	Liczba godz. ćwic./lab.:
Cele przedmiotu: <ol style="list-style-type: none"> 1. wprowadzenie głównych modeli obliczeń komputerowych 2. matematyczna analiza problemu rozstrzygalności 3. związki teorii obliczalności z teorią mnogości 4. wyjaśnienie ograniczeń związanych ze złożonością programów 		
Zawartość programowa: <ol style="list-style-type: none"> 1. Funkcje i relacje obliczalne 2. Rekurencyjna przeliczalność 3. Zagadnienia rozstrzygalności 4. Twierdzenia o punkcie stałym 5. Redukowalność 6. Obliczalność względna 7. Hierarchia arytmetyczna 8. Modele obliczeń 9. Klasy złożoności 10. Modele obliczeń funkcyjnych i logicznych 		
Literatura: <ol style="list-style-type: none"> 1. M. Sipser, "Introduction to Theoretical Computer Science", Spring 2002. 2. A. Grzegorzczak, "Zarys logiki matematycznej", PWN Warszawa 1961. 3. Ch. H. Papadimitriou, "Złożoność obliczeniowa", WNT Warszawa 2002. 4. A. Shen, N. Vereshchagin, "Computable functions", AMS 2003. 5. B. Cooper, "Computability theory", Chapman and Hall/CRC, 2002. 		
Forma prowadzonych zajęć: wykład		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Metody inżynierii oprogramowania</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
Cele przedmiotu: Celem przedmiotu jest prezentacja wybranych metodologii tworzenia oprogramowania, metod analizy i projektowania systemów informatycznych, sposobów przygotowywania dokumentacji technicznej i użytkowej oraz sposobów testowania systemów i szacowania ich niezawodności.		
Zawartość programowa: 1. Zapoznanie z podstawowymi elementami zunifikowanego języka modelowania UML. 2. Znaczenie i praktyczne zastosowanie diagramów UML. 3. Prezentacja wybranych metodologii tworzenia oprogramowania (model kaskadowy, iteracyjny i przyrostowy, SDLC, RAD SDLC, GRAPPLE, RUP, programowanie odkrywczе, prototypowanie, model spiralny). 4. Sposoby testowania systemów. Śledzenie błędów czasu projektowania i wykonywania programu. 5. Systemy archiwizacji błędów czasu wykonywania. 6. Narzędzia CASE.		
Literatura: 1. Andrzej Jaskiewicz, Inżynieria oprogramowania, Helion 1997. 2. J.A. Hoffer, J.F. George, J.S. Valacich, Modern Systems Analysis and Design, Addison-Wesley, 1999. 3. J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison-Wesley; 2nd edition, 2004 (lub polskie tłumaczenia tej pozycji).		
Forma prowadzonych zajęć: wykład + laboratorium		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość zasad programowania obiektowego w języku C++		
Narzędzia programistyczne (zajęcia laboratoryjne): Kompilatory C++/BCB/BDS. Narzędzia modelowania wizualnego Visio, Together, Poseidon		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Semantyka i realizacja języków programowania</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Celem przedmiotu jest pokazanie roli, problemów i technik formalizacji opisu języków programowania w formie różnych semantyk. Z drugiej strony przedmiot ma pokazać zastosowanie tych metod w praktyce przez tworzenie programów przetwarzających takie języki (interpretery, kompilatory, programy pokrewne).</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Formalny opis języków programowania 2. Operacyjne i denotacyjne metody definiowania semantyki programów 3. Semantyczne definicje podstawowych konstrukcji programistycznych 4. Analiza leksykalna 5. Analiza składniowa 6. Analiza semantyczna 7. Generacja kodu pośredniego i kodu wynikowego 8. Tablica symboli 9. Użycie flexa (lexa) oraz bisona (yacca) 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. A.V. Aho, M.S. Lam, R. Sethi, J.D. Ullman, <i>Kompilatory. Reguły, metody i narzędzia</i>, Wydawnictwa Naukowo-Techniczne, Warszawa 2002). 2. M. Foryś, W. Foryś, <i>Teoria automatów i języków formalnych</i>, Akademicka Oficyna Wydawnicza Exit, 2005. 3. J.P. Tremblay, P.G. Sorenson, <i>The Theory and Practice of Compiler Writing</i>, McGraw-Hill, 1985. 4. W.M. Waite, G. Goos, <i>Konstrukcja kompilatorów</i>, Wydawnictwa Naukowo-Techniczne, Warszawa 1989. 5. P. Dembiński, J. Manuszyński. <i>Matematyczne metody definiowania języków programowania</i>. WNT, 1981. 6. M. Gordon. <i>Denotacyjny opis języków programowania</i>. WNT, 1983. 		
Forma prowadzonych zajęć: wykład + laboratorium		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) umiejętność programowania w C, znajomość algorytmów i struktur danych		
Narzędzia programistyczne (zajęcia laboratoryjne): gcc, flex, bison		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Pracownia programowania obiektowego</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Zajęcia poświęcone są przedstawieniu zasad programowania obiektowego i zorientowanego obiektowo. Zostają omówione podstawowe typy klas oraz związków, w jakich klasy mogą występować. Rozważania teoretyczne zilustrowane są praktycznymi sposobami wykorzystywania wzorców projektowych.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Definicja klasy, elementy składowe klasy. 2. Klasy polimorficzne, abstrakcyjne i wirtualne. 3. Konstruktory. Cykl życia obiektu stworzonego na bazie klasy. 4. Wzorce klas. 5. Związki klas: dziedziczenie, powiązania (agregacje, kompozycje), zależności. 6. Kontenery biblioteki standardowej. 7. Implementacji związków klas na przykładach wybranych wzorców projektowych. 8. Systemy transakcyjne. 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. Alan Shalloway, James R. Trott, Projektowanie zorientowane obiektowo. Wzorce projektowe, Helion 2005 2. Stephen Prata, Język C++. Szkoła programowania. Wydanie V, Helion 2006 		
<p>Forma prowadzonych zajęć: wykład + laboratorium</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość zasad programowania strukturalnego i proceduralnego w języku C++</p>		
<p>Narzędzia programistyczne: Kompilatory C++/g++/BCB/BDS</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Wykład specjalistyczny I</i>	Liczba godz. wykładu: 30	Liczba godz. ćwicz./lab.:
Cele przedmiotu: Zapoznanie z podstawami projektowania i wykorzystania systemów ekspertowych		
Zawartość programowa: <ol style="list-style-type: none"> 1. Struktura systemów eksperckich. 2. Reprezentacja wiedzy w systemach eksperckich. 3. Pozyskiwanie wiedzy do systemów eksperckich. 4. Przetwarzanie wiedzy w systemach eksperckich. 5. Narzędzia do konstruowania systemów eksperckich. 		
Literatura: <ul style="list-style-type: none"> • Bubnicki Z., <i>Wstęp do systemów ekspertowych</i>, PWN, 1990 • Mulawka J., <i>Systemy ekspertowe</i>, WNT, 1996 • Świątnicki Z., <i>Expert Systems. Introduction</i>, Bellona, 1998 		
Forma prowadzonych zajęć: wykład		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Informatyka medyczna</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Poznanie zakresu i możliwości zastosowania informatyki w medycynie, oraz nabycie umiejętności tworzenia oprogramowania do analizy sygnałów biomedycznych.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Analiza sygnałów biomedycznych <ul style="list-style-type: none"> - Źródła sygnałów - Akwizycja i przetwarzanie sygnałów - Sygnały dźwiękowe - struktura plików dźwiękowych 2. Analiza sygnałów w dziedzinie czasu <ul style="list-style-type: none"> - wartość średnia i energia sygnału - częstotliwość przejść przez zero - obwiednia amplitudowa sygnału - analiza korelacyjna 3. Częstotliwościowa analiza sygnałów <ul style="list-style-type: none"> - Algorytm szybkiej transformacji Fouriera (FFT) - Widmowa gęstość mocy - Rozkład czasowo-spektralny - Spektrogramy 4. Metoda liniowej predykcji 5. Analiza falkowa w zastosowaniach medycznych 6. Filtry cyfrowe w zastosowaniach medycznych 7. Systemy obrazowania medycznego <ul style="list-style-type: none"> - techniki akwizycji obrazu - przetwarzanie obrazów biomedycznych 8. Zastosowanie zbiorów rozmytych w medycynie 9. Miary rozmyte i ich zastosowanie w diagnostyce 10. Zastosowanie zbiorów przybliżonych w medycynie 11. Sieci neuronowe w diagnostyce medycznej 12. Systemy komputerowe dla różnych szczebli opieki zdrowotnej 13. Sieci komputerowe w służbie zdrowia 14. Standardy zapisu i przesyłania danych medycznych 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. Zajdel R., Kęcki E., Szczepaniak P., Kurzyński M., Kompendium informatyki medycznej, Alfa Medica Press, Bielsko-Biała 2003. 2. Kelemen A., Abraham A., Liang Y., computational Intelligence in Medical Informatics, Springer 2007. 3. Tadeusiewicz R., Ogiela M. R., Medical Image Understanding Technology, Springer 2004. 4. Informatyka medyczna, red. R. Rudowski, PWN, Warszawa 2003. 5. Haas O. C. L., Burnham K. J., Intelligent and Adaptive Systems in Medicine, Taylor&Francis 2008. 6. Annales UMCS, sec. Informatica, Lublin, 2005-8. 7. Ed.: J. Tan, E-Health Care Information Systems, Jossey-Bass 2005. 		
<p>Forma prowadzonych zajęć: wykład + laboratorium</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość podstaw teorii sygnałów oraz cyfrowych metod ich przetwarzania, grafiki komputerowej oraz umiejętności programowania komputerów</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 1
Nazwa przedmiotu: <i>Seminarium</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Głównym celem zajęć jest przygotowanie i ustalenie zakresu tematyki związanej z pracą dyplomową. Referaty głoszone przez studentów mają również za zadanie przygotowanie studenta do ewentualnej przyszłej pracy dydaktycznej i umiejętności wyboru i przekazywania informacji w wystąpieniach na forum publicznym .</p>		
<p>Zawartość programowa: Zakres materiału prezentowanego podczas referatów i dyskusji związana jest ściśle z tematyką prac magisterskich poszczególnych studentów.</p>		
<p>Literatura: Dobierana do indywidualnej tematyki prac magisterskich.</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Ogólna wiedza zdobyta przez studenta podczas zajęć dydaktycznych prowadzonych na studiach.</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Algorytmy logiki rozmytej</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Poznanie podstaw teoretycznych i algorytmów logiki rozmytej oraz nabycie umiejętności tworzenia programów komputerowych przy wykorzystaniu tej techniki.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Podstawowe pojęcia, definicje i przykłady zbiorów rozmytych 2. Funkcje przynależności 3. Własności zbiorów rozmytych 4. Operacje na zbiorach rozmytych <ul style="list-style-type: none"> - mnogościowe, algebraiczne, logiczne i drastyczne sumy i iloczyny - dopełnienie - normy trójkątne - modyfikatory - różnice zbiorów rozmytych - kompensatory - iloczyn kartezjański 5. Zasada rozszerzania 6. Miary <ul style="list-style-type: none"> - miary rozmyte - całka rozmyta - miary odległości i rozmytości - inkluzja rozmyta i równość rozmyta 7. Liczby rozmyte <ul style="list-style-type: none"> - podział liczb rozmytych - działania na liczbach rozmytych 8. Liczby LR i ich arytmetyka 9. Relacje rozmyte i ich właściwości 10. Logika rozmyta <ul style="list-style-type: none"> - zmienna lingwistyczna - podstawowe reguły wnioskowania w logice rozmytej - reguły rozmytej implikacji 11. Sterowniki rozmyte <ul style="list-style-type: none"> - klasyczny sterownik rozmyty - sterownik rozmyty Takagi-Sugeno - przykłady praktycznych zastosowań sterowników rozmytych - Zbiory rozmyte typu 2. 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. Łęski J., Systemy neuronowo-rozmyte, WNT Warszawa 2008. 2. Rutkowski L., Metody i techniki sztucznej inteligencji, PWN Warszawa 2005. 3. Cader A., Rutkowski L., Tadeusiewicz R., Zurada J., Artificial Intelligence and Soft Computing, EXIT Warszawa 2006. 4. Łachwa A., Rozmyty świat zbiorów, liczb, relacji, faktów i decyzji, Akademicka Oficyna wydawnicza EXIT, Warszawa 2001. 5. Sanchez E., Fuzzy logic and the Semantic Web, Elsevier 2006. 6. Klir G. J., Yuan B., Fuzzy Sets and Fuzzy Logic, Prentice Hall PTR 2008. 		
<p>Forma prowadzonych zajęć: wykład + laboratorium</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Ogólna znajomość podstaw teorii zbiorów ostrych i logiki klasycznej, umiejętność programowania komputerów</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Podstawy telekomunikacji</i>	Liczba godz. wykładu: 30	Liczba godz. ćwic./lab.:
Cele przedmiotu: Zapoznanie słuchaczy z podstawowymi technologiami i technikami stosowanymi współcześnie w telekomunikacji.		
Zawartość programowa: <ol style="list-style-type: none"> 1. Sieci teleinformatyczne i standardy komunikacji międzysieciowej. Typy sieci telekomunikacyjnych i informatycznych. 2. Media transmisyjne: podstawowe charakterystyki okablowania miedzianego i światłowodowego, właściwości i podział fal radiowych, zakres widzialny fal elektromagnetycznych. 3. Współczesne usługi oferowane w sieciach teleinformatycznych: sieci inteligentne, sieci wirtualne, sieci domowe, przewodowe i bezprzewodowe usługi szerokopasmowe. 4. Telefonía mobilna i sieci bezprzewodowe, standardy 802.11, bluetooth, telekomunikacja satelitarna; zarządzanie i monitorowanie sieci. 5. Technologie głosowe VoDATA, przekazy multimedialnych oraz komunikacja internetowa. 6. Tworzenie optycznych sieci szkieletowych DWDM. 7. Sieci następnej generacji NGN. 8. Protokoły mediacji i trasowania w sieciach. 9. Bezpieczeństwo przekazów, tworzenie nowoczesnych systemów archiwizacji, zagadnienia bezpieczeństwa i obsługi centrum danych. 10. Metody testowania sieci teleinformatycznych i urządzeń teleinformatyki 		
Literatura: <ol style="list-style-type: none"> 1. Praca zbiorowa: <i>Vademecum teleinformatyka</i>, cz. I, II, III (IDG 2004) 2. Mark Noris: <i>Teleinformatyka</i> (WKŁ 2002) 3. Roger J. Sutton: <i>Bezpieczeństwo telekomunikacji</i> (WKŁ 2004) 4. Andrzej Jajszczyk: <i>Wstęp do telekomunikacji</i> (WNT 2004) 5. Krzysztof Wesołowski: <i>Systemy radiokomunikacji ruchomej</i> (WKŁ 2004) 		
Forma prowadzonych zajęć: wykład		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Wymagana wiedza i umiejętności dotyczące sieci komputerowych, fal elektromagnetycznych, światłowodów, kompresji danych.		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Metody sztucznej inteligencji</i>	Liczba godz. wykładu: 30	Liczba godz. ćwic./lab.:
Cele przedmiotu: Opanowanie wiedzy z zakresu różnych metod i narzędzi sztucznej inteligencji.		
Zawartość programowa: <ol style="list-style-type: none"> 1. Systemy ekspertowe i reprezentacja wiedzy 2. Reprezentacja wiedzy niepewnej 3. Programowanie w logice 4. Programowanie w logice z ograniczeniami 5. Silniki reguł 6. Ontologie 7. Case based reasoning 8. Systemy uczące się 9. Data Mining 10. Rozpoznawanie wzorców 11. Teoria gier 		
Literatura: <p>A. Niederliński „Regułowe Systemy Ekspertowe”, Pracownia Komputerowa Jacka Skalmierskiego 2000.</p> <p>P. Cichosz „Systemy uczące się”, WNT 2001.</p> <p>Russel, Norvig “Artificial Intelligence: A modern Aproach”, Prentice Hall 2003.</p> <p>T. Żurek „Komputerowe wspomaganie podejmowania decyzji kredytowych”, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2005.</p> <p>W. F. Clocksin, C. S. Mellish „Prolog Programowanie”, Helion 2003.</p>		
Forma prowadzonych zajęć: wykład		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami): programowanie w dowolnym języku		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Programowanie aplikacji bazodanowych</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 45
Cele przedmiotu: Projektowanie i implementacja aplikacji internetowych. Narzędzia i środowiska wytwarzania oprogramowania. Technologie wspierające programowanie aplikacji (frameworki sieciowe, aplikacje itp.). Korzystanie z Application Programming Interface .		
Zawartość programowa: <ol style="list-style-type: none"> 1. Aplikacje desktopowe w javie z wykorzystaniem języka sql 2. Aplikacje desktopowe w javie z wykorzystaniem języka QL (TopLink oraz Hibernate jako narzędzia odwzorowań obiektowo relacyjnych) 3. Framework Java Server Faces w połączeniu z POJO (Plain Object Java ...) 4. Komponenty zarządzające, nawigacja. 5. Dane tabelaryczne i pojedyncze. CRUD (Create, Update, Delete, ...) 6. Konwersja i walidacja danych 7. Komponent użytkownika (konwerter lub walidator, tworzenie bibliotek tld) 8. Architektura trój- i więcej warstwowa (EJB3.0). 9. Komponenty encyjne i sesyjne. 10. Komponenty zarządzane kontenerem oraz komponenty zarządzane aplikacją. 11. Java Persistence Application (relacje, typy relacji). 12. Alternatywne technologie (Spring, SEAM). Pliki konfiguracyjne. 		
Literatura: <ol style="list-style-type: none"> 1. Cay Horstmann, Gary Cornell Core Java 2, Techniki zaawansowane, Helion, 2005 2. Cay Horstmann Java Server Faces, wydanie I 3. JEE5 Sun Java System Application Server 9.1., http://java.sun.com/javaee/5/docs/tutorial/doc/ 4. Enterprise Java Bean 3.0 Burk, Helion 2007 (wydanie polskojęzyczne) 5. Shang Shin, Java Passion http://www.javapassion.com/j2ee/ 		
Forma prowadzonych zajęć: wykład + laboratorium		
Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) <ul style="list-style-type: none"> - podstawy języka programowania Java - podstawowe wiadomości z inżynierii oprogramowania 		
Narzędzia programistyczne (zajęcia laboratoryjne): <ul style="list-style-type: none"> - Środowisko programistyczne NetBeans lub Eclipse wraz z pluginem do modelowania diagramów - DBDesigner lub zbliżony program - Baza danych Oracle XE lub MySQL 		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Budowa sieci komputerowych</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 75
Cele przedmiotu: Poznanie zasad budowy i konfiguracji sieci komputerowych. Zapoznanie z koncepcjami zaawansowanego routingu i protokołów pomocniczych		
Zawartość programowa: <ol style="list-style-type: none"> 1. Wstęp do routingu i przesyłania pakietów 2. Routing statyczny 3. Wstęp do protokołów routingu dynamicznego 4. Protokoły wektora odległości 5. RIP v. 1 6. VLSM i CIDR 7. RIP v. 2 8. Bliższe zapoznanie z tablicą routingu 9. EIGRP 10. Protokoły stanu łącza 11. OSPF 		
Literatura: <ol style="list-style-type: none"> 1. Andrew S. Tanenbaum, Sieci Komputerowego, Helion 2004. 2. Akademia Sieci Cisco, Pierwszy Rok Nauki, MIKON 2002. 3. Kevin Dooley, Ian J. Brown, Cisco. Receptury, Helion 2004. 4. Innokenty Rudenko, Tsunami Computing, Routery Cisco. Czarna księga, Helion 2001. 		
Forma prowadzonych zajęć: laboratorium		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Programowanie równoległe</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.:
<p>Cele przedmiotu: Wykład ma na celu zaznajomienie studentów ze współczesnymi technologiami dotyczącymi programowania obliczeń równoległych. Celem laboratorium jest nabycie przez studentów umiejętności tworzenia oprogramowania dla środowisk równoległych oraz zapoznanie się z różnymi standardami wspierającymi równoległość i zwiększającymi wydajność aplikacji. Studenci mają wykorzystać nabytą na wykładzie wiedzę do tworzenia własnych programów wykorzystujących algorytmy równoległe. Poprzez rozwiązywanie specjalnie dobranych zadań, studenci mają również nauczyć się wykrywać i unikać typowych błędów związanych z programowaniem dla architektur równoległych z pamięcią wspólną oraz z pamięcią rozproszoną.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Potrzeba obliczeń wysokowydajnych, dziedziny zastosowań. Idea obliczeń równoległych. 2. Klasyfikacja komputerów wg Flynna, podział komputerów wieloprocesorowych ze względu na pamięć. Wady i zalety programowania komputerów z pamięcią wspólną oraz komputerów z pamięcią rozproszoną. 3. Modele i metody analizy programów równoległych i ograniczenia równoległości. Prawo Amdahla. Model Gustafsona. Zastosowanie modeli do predykcji czasu wykonania algorytmów oraz poszukiwania algorytmów optymalnych. 4. Programowanie komputerów z pamięcią rozproszoną przy użyciu MPI. Instalacja i konfiguracja środowiska MPI w systemie Linux. Tworzenie programów równoległych w środowisku MPI dla komputerów z pamięcią rozproszoną. Procedury służące do komunikacji w środowisku MPI. Komunikacja punkt-punkt (komunikacja z blokowaniem, komunikacja bez blokowania, tryby komunikacji), komunikacja grupowa (broadcast, rozsyłanie i zbieranie danych z wielu procesorów, obliczenia globalne). Tworzenie grup i komunikatorów, mechanizm topologii. Analiza przyspieszenia i efektywności programów równoległych w środowisku MPI. Metody dystrybucji danych dla obliczeń na komputerach z pamięcią rozproszoną. 5. Programowanie komputerów z pamięcią wspólną z wykorzystaniem standardu OpenMP. Dyrektywy, tworzenie wątków, konstrukcje specyfikujące równoległe wykonanie kodu. Kompilacja warunkowa. Konstrukcje określające zakres widoczności zmiennych. Podział pracy pomiędzy wątki, zrównoleglenie pętli oraz nieregularnych fragmentów kodu, lokalna jednowątkowość, konstrukcje synchronizacji, sekcje krytyczne. Zmienne środowiskowe i biblioteka funkcji OpenMP. Analiza przyspieszenia i efektywności programów równoległych w standardzie OpenMP. 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. Wilkinson B., Allen M.; Parallel programming, Techniques and Applications Using Networked Workstations and Parallel Computers; Prentice Hall; 1999. 2. Michael J. Quinn; Parallel Programming in C with MPI and OpenMP; OREGON STATE UNIVERSITY; 2004. 3. Erricos John Kontoghiorghes; Handbook of Parallel Computing and Statistics; University of Cyprus and University of London, UK; 2005; 4. George Em Karniadakis, Robert M. Kirby II, Parallel Scientific Computing in C++ and MPI, Cambridge University Press, 2007. 5. Ian Foster; Designing and Building Parallel Programs; Addison-Wesley; 1996. 6. http://www-unix.mcs.anl.gov/mpi/ 7. http://openmp.org/ 		
Forma prowadzonych zajęć: wykład		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość problemów omawianych na przedmiocie Programowanie współbieżne i rozproszone. Zaliczony przedmiot Programowanie w języku C++.</p>		
<p>Narzędzia programistyczne (zajęcia laboratoryjne): kompilator GNU C/C++, środowisko Open MPI</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Programowanie równoległe</i>	Liczba godz. wykładu:	Liczba godz. ćwic./lab.: 30
<p>Cele przedmiotu: Celem laboratorium jest nabycie przez studentów umiejętności programowania w środowiskach równoległych oraz zapoznanie się z różnymi standardami wspierającymi równoległość i zwiększającymi wydajność aplikacji. Studenci mają wykorzystać nabytą na wykładzie wiedzę do tworzenia własnych programów wykorzystujących algorytmy równoległe. Poprzez rozwiązywanie specjalnie dobranych zadań, studenci mają również nauczyć się wykrywać i unikać typowych błędów związanych z programowaniem architekturach równoległych z pamięcią wspólną oraz z pamięcią rozproszoną.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Hierarchia pamięci komputera. Biblioteka BLAS. Funkcje BLAS poziomu 1 (operacje wektor-wektor), funkcje BLAS poziomu 2 (operacje macierz-wektor). 2. Funkcje BLAS poziomu 3 (operacje macierz-macierz). Porównanie wyników rozwiązania zadań z wykorzystaniem funkcji BLAS wszystkich poziomów. 3. Instalacja i konfiguracja środowiska MPI w systemie Linux. 4. Tworzenie programów równoległych w środowisku MPI dla komputerów z pamięcią rozproszoną. 5. Procedury służące do komunikacji w środowisku MPI. Komunikacja punkt-punkt (komunikacja z blokowaniem, komunikacja bez blokowania, tryby komunikacji), komunikacja grupowa (broadcast, rozsyłanie i zbieranie danych z wielu procesorów, obliczenia globalne). 6. Tworzenie grup i komunikatorów, mechanizm topologii. 7. Analiza przyspieszenia i efektywności programów równoległych w środowisku MPI. 8. Instalacja kompilatora Omni OpenMP w środowisku Linux. 9. Tworzenie programów wielowątkowych dla komputerów z pamięcią wspólną z wykorzystaniem środowiska OpenMP. Dyrektywy, tworzenie wątków, konstrukcje specyfikujące równoległe wykonanie kodu. Kompilacja warunkowa. Konstrukcje określające zakres widoczności zmiennych. 10. Podział pracy pomiędzy wątki, zrównoleglenie pętli oraz nieregularnych fragmentów kodu, lokalna jednowątkowość, konstrukcje synchronizacji, sekcje krytyczne. 11. Zmienne środowiskowe i biblioteka funkcji OpenMP. 12. Tworzenie równoległych wersji funkcji z biblioteki BLAS z wykorzystaniem standardu OpenMP. 13. Analiza przyspieszenia i efektywności programów równoległych w standardzie OpenMP. 		
<p>Literatura:</p> <ul style="list-style-type: none"> • Wilkinson B., Allen M.; Parallel programming, Techniques and Applications Using Networked Workstations and Parallel Computers; Prentice Hall; 1999. • http://www-unix.mcs.anl.gov/mpi/ • http://netlib.org/blas/index.html • http://netlib.org/atlas/index.html • Michael J. Quinn; Parallel Programming in C with MPI and OpenMP; OREGON STATE UNIVERSITY; 2004. • Erricos John Kontoghiorghes; Handbook of Parallel Computing and Statistics; University of Cyprus and University of London, UK; 2005; • Ian Foster; Designing and Building Parallel Programs; Addison-Wesley; 1996. 		
Forma prowadzonych zajęć: laboratorium		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Znajomość problemów omawianych na przedmiocie Programowanie współbieżne i rozproszone</p>		
<p>Narzędzia programistyczne (zajęcia laboratoryjne): kompilator GNU C/C++, kompilator Ompi OpenMP, środowisko mpich lub lam mpi, kompilator Berkeley UPC</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: I / 2
Nazwa przedmiotu: <i>Seminarium</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Głównym celem zajęć jest przygotowanie i ustalenie zakresu tematyki związanej z pracą dyplomową. Referaty głoszone przez studentów mają również za zadanie przygotowanie studenta do ewentualnej przyszłej pracy dydaktycznej i umiejętności wyboru i przekazywania informacji w wystąpieniach na forum publicznym .</p>		
<p>Zawartość programowa: Zakres materiału prezentowanego podczas referatów i dyskusji związana jest ściśle z tematyką prac magisterskich poszczególnych studentów.</p>		
<p>Literatura: Dobierana do indywidualnej tematyki prac magisterskich.</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Ogólna wiedza zdobyta przez studenta podczas zajęć dydaktycznych prowadzonych na studiach.</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Podstawy i zastosowania teorii grafów</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
Cele przedmiotu: Zapoznanie studentów z podstawowymi pojęciami i algorytmami teorii grafów.		
Zawartość programowa: <ol style="list-style-type: none"> 1. Przeszukiwanie grafów wszerz i w głąb 2. Kolorowanie grafów (algorytm First Fit) 3. Sortowanie topologiczne wierzchołków acyklicznych grafów skierowanych. 4. Metoda ścieżki krytycznej 5. Cykl Eulera 6. Problem komiwojażera i cykl Hamiltona 7. Najkrótsza ścieżka w grafie – algorytm Dijkstry 8. Minimalne drzewo rozpinające – algorytm Prima 9. Najkrótsza ścieżka w grafie -algorytmy Forda-Belmana i Floyda-Warshola 10. Przepływy w grafie. 11. Drzewa poszukiwań binarnych 12. Drzewa mini-max i drzewa gry 13. Kopce (kolejka priorytetowa, sortowanie przez kopcowanie) 		
Literatura: Donald Knuth, <i>Sztuka programowania</i> , WNT Warszawa 2000. Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., Stein Clifford, <i>Wprowadzenie do algorytmów</i> , WNT Warszawa 2004 Lech Banachowski, Krzysztof Diks, Wojciech Rytter, <i>Algorytmy i struktury danych</i> , WNT Warszawa 1989. http://pl.wikipedia.org/wiki/Teoria_grafów		
Forma prowadzonych zajęć: wykład + laboratorium		
Narzędzia programistyczne (zajęcia laboratoryjne): Edytor tekstu w środowisku Unix, kompilator do języka C (gcc/g++)		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Wykład monograficzny</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.:
Cele przedmiotu: Zapoznanie z podstawowymi zagadnieniami cybernetyki jako nauki o sterowaniu w systemach technicznych, biologicznych i społecznych		
Zawartość programowa: <ol style="list-style-type: none"> 1. Cybernetyka. Systemy sterowania. 2. Sterowanie przy zadanym stanie (wyjściu). 3. Sterowanie optymalne. 4. Optymalizacja parametryczna. 5. Sterowanie w warunkach niepewności. 6. Sterowanie w systemie zamkniętym. Stabilność. 7. Adaptacyjne i uczące się systemy sterowania 		
Literatura: <ol style="list-style-type: none"> 1. Ashby W.R.: <i>Wstęp do cybernetyki</i>, PWN, 1961. 2. Bubnicki Z.: <i>Teoria i algorytmy sterowania</i>, PWN, 2002. 3. Kaczorek T.: <i>Teoria układów regulacji automatycznej</i>, WNT, 1971. 4. Kwiatkowski W.: <i>Wprowadzenie do automatyki</i>, BEL Studio, 2005 5. Lerner A.J.: <i>Zarys cybernetyki</i>, WNT, 1971. 		
Forma prowadzonych zajęć: wykład		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Metody optymalizacji</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Celem przedmiotu jest przedstawienie teoretycznych zagadnień oraz numerycznych algorytmów służących do rozwiązywania zadań optymalizacji statycznej. Omówione będą niezbędne podstawy matematyczne, a następnie wybrane, najczęściej używane algorytmy służące do rozwiązywania takich zadań.</p>		
<p>Zawartość programowa:</p> <ol style="list-style-type: none"> 1. Przykłady zadań optymalizacji, klasyfikacje zadań optymalizacji. 2. Wprowadzenie do metod rozwiązywania zadań optymalizacji statycznej. 3. Podstawowe własności zadania programowania liniowego; metoda simplex. 4. Podstawy matematycznej analizy nieliniowych zadań optymalizacji statycznej. 5. Podstawy metod optymalizacji bez ograniczeń. 6. Metody rozwiązywania zadania poprawy. 7. Gradientowe algorytmy rozwiązywania zadań optymalizacji bez ograniczeń. 8. Wpływ ograniczeń na rozwiązywanie zadań optymalizacji. 9. Analiza matematyczna zadań optymalizacji z ograniczeniami. 10. Metody i algorytmy rozwiązywania zadań optymalizacji z ograniczeniami. 11. Elementy programowania wielokryterialnego. 		
<p>Literatura:</p> <ol style="list-style-type: none"> 1. A. Stachurski, A. Wierzbicki „Podstawy optymalizacji” Oficyna Wydawnicza PW, Warszawa 1999. 2. W. Findeisen, J. Szymanowski, A. Wierzbicki „Teoria i metody obliczeniowe optymalizacji” Państwowe Wydawnictwo Naukowe, Warszawa 1977 (część I). 3. M. Oswald „Podstawy optymalizacji konstrukcji” Wydawnictwo Politechniki Poznańskiej, Poznań 2003. 4. M. Brdyś, A. Ruszczyński „Metody optymalizacji w zadaniach” Wydawnictwa Naukowo-Techniczne, Warszawa 1985. 		
Forma prowadzonych zajęć: wykład + laboratorium		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Symulacja i modelowanie</i>	Liczba godz. wykładu: 15	Liczba godz. ćwicz./lab.: 30
Cele przedmiotu: Zapoznanie z podstawami modelowania zjawisk i procesów. Analiza wydajności zbudowanego modelu.		
Zawartość programowa:		
<ol style="list-style-type: none"> 1. Generatory liczb pseudolosowych, <ul style="list-style-type: none"> - Generowanie liczb pseudolosowych o rozkładzie jednostajnym - Generatory liniowe - Uogólnione generatory Fibonacciego - Generatory nieliniowe - Inne rozkłady liczb pseudolosowych – rozkład wykładniczy i normalny 2. Metody matematycznego modelowania złożonych procesów. Metoda Monte Carlo w zastosowaniach <ul style="list-style-type: none"> - Całkowanie numeryczne, - Symulacja rozkładu temperatury na płaszczyźnie, - Symulacja przepływu cząstek w środowisku homogenicznym 3. Automaty komórkowe <ul style="list-style-type: none"> - Gra w życie Conwaya, - Mrówka Langtona , - Wire World, - Generowanie liczb pierwszych 4. Modele kolejkowe probabilistyczne i modele z naturalnym systemem obsługi <ul style="list-style-type: none"> - Elementy składowe i analiza modelu M/M/1 - Elementy składowe i analiza modelu M/M/c/K/H 5. Problemy i metody modelowania systemów sieci komputerowych za pomocą modeli kolejkowych. 		
Literatura:		
<ul style="list-style-type: none"> - Wykłady z podstaw informatyki prof. Stefana Węgrzyna. Gliwice 2003 - T. Czachórski. Modele kolejkowe systemów komputerowych. Wydawnictwo Politechniki Śląskiej 1999 - Wieczorkowski R., Zieliński R. Komputerowe generatory liczb losowych. WNT Warszawa 2005. 		
Forma prowadzonych zajęć: wykład + laboratorium		
Wymagania wstępne: znajomość problemów omawianych na przedmiotach: programowanie obiektowe w języku C++/Java, analiza matematyczna, algorytmy i struktury danych		
Narzędzia programistyczne (zajęcia laboratoryjne); Do wyboru: Delphi. Builder c++, Visual Studio, QT, Java		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Informatyczna pracownia magisterska</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 30
Cele przedmiotu: Celem przedmiotu jest udostępnienie specjalistycznego sprzętu i oprogramowania niezbędnego podczas pisania pracy magisterskiej. Pomoc nauczyciela akademickiego w rozwiązywaniu problemów napotkanych w trakcie realizacji tematu.		
Zawartość programowa: Praktyczna realizacja zagadnień związanych z pisaniem pracy dyplomowej.		
Literatura: Określana indywidualnie do każdego tematu.		
Wymagania wstępne: znajomość przedmiotów niezbędnych do realizacji pracy dyplomowej.		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Złożoność obliczeniowa</i>	Liczba godz. wykładu: 30	Liczba godz. ćwic./lab.:
Cele przedmiotu: Zaznajomienie studentów z podstawowymi zagadnieniami złożoności obliczeniowej oraz wskazanie praktycznej aplikacji analizy złożoności.		
Zawartość programowa: <ol style="list-style-type: none"> 1. Różne modele obliczeniowe i ich równoważność 2. Twierdzenia o przyspieszeniu i kompresji 3. Twierdzenia o separacji 4. Główne klasy złożoności czasowej i pamięciowej 5. Redukowalność i problemy zupełne 6. Charakterystyka klas P i NP. 7. Klasy L, NL, PSPACE, EXPTIME 8. Zagadnienia złożoności obliczeń równoległych i współbieżnych 9. Problemy aproksymacji 10. Abstrakcyjna teoria złożoności 		
Literatura: <ol style="list-style-type: none"> 1. C.H. Papadimitriou, Złożoność obliczeniowa, WNT 2007 2. J. L. Balcazar, J. Diaz, J. Gabarro, Structural Complexity, Springer-Verlag I (1988), II (1990) 3. D. Kozen, Theory of Computation, Springer-Verlag, 2006 		
Forma prowadzonych zajęć: wykład		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Seminarium</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 30
<p>Cele przedmiotu: Głównym celem zajęć jest przygotowanie i ustalenie zakresu tematyki związanej z pracą dyplomową. Referaty głoszone przez studentów mają również za zadanie przygotowanie studenta do ewentualnej przyszłej pracy dydaktycznej i umiejętności wyboru i przekazywania informacji w wystąpieniach na forum publicznym .</p>		
<p>Zawartość programowa: Zakres materiału prezentowanego podczas referatów i dyskusji związana jest ściśle z tematyką prac magisterskich poszczególnych studentów.</p>		
<p>Literatura: Dobierana do indywidualnej tematyki prac magisterskich.</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Ogólna wiedza zdobyta przez studenta podczas zajęć dydaktycznych prowadzonych na studiach.</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Oprogramowanie sieci komputerowych</i>	Liczba godz. wykładu:	Liczba godz. ćwicz./lab.: 75
Cele przedmiotu: Zapoznanie z zagadnieniami przełączania w sieciach lokalnych i podstawami sieci bezprzewodowej, opanowanie praktycznej umiejętności konfigurowania przełączników		
Zawartość programowa: <ol style="list-style-type: none"> 1. Projektowanie sieci lokalnych 2. Podstawy działania i konfiguracji przełączników 3. Wirtualne sieci LAN 4. Protokół VTP 5. Protokół drzewa opinającego (STP) 6. Routing między wirtualnymi sieciami LAN 7. Podstawy działania i konfiguracji lokalnych sieci bezprzewodowych 		
Literatura: <ol style="list-style-type: none"> 1. W. Lewis, Akademia sieci Cisco CCNA Exploration semestr 3 z płytą CD, PWN 2009 2. W.A. Kaczmarek, Certyfikacja CCNA Podstawy działania sieci komputerowych zasady przełączania i routingu, PWN 2007 3. L. Jonathan, P. Roshan, Bezprzewodowe sieci LAN 802.11 podstawy, PWN 2007 		
Forma prowadzonych zajęć: laboratorium		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 3
Nazwa przedmiotu: <i>Specjalistyczna pracownia magisterska</i>	Liczba godz. wykładu:	Liczba godz. ćwic./lab.: 15
Cele przedmiotu: Celem przedmiotu jest udostępnienie specjalistycznego sprzętu i oprogramowania niezbędnego podczas pisania pracy magisterskiej. Pomoc nauczyciela akademickiego w rozwiązywaniu problemów napotkanych w trakcie realizacji tematu.		
Zawartość programowa: Praktyczna realizacja zagadnień związanych z pisaniem pracy dyplomowej.		
Literatura: Określana indywidualnie do każdego tematu.		
Wymagania wstępne: znajomość przedmiotów niezbędnych do realizacji pracy dyplomowej.		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 4
Nazwa przedmiotu: <i>Informatyczna pracownia magisterska</i>	Liczba godz. wykładu:	Liczba godz. ćwic./lab.: 30
Cele przedmiotu: Celem przedmiotu jest udostępnienie specjalistycznego sprzętu i oprogramowania niezbędnego podczas pisania pracy magisterskiej. Pomoc nauczyciela akademickiego w rozwiązywaniu problemów napotkanych w trakcie realizacji tematu.		
Zawartość programowa: Praktyczna realizacja zagadnień związanych z pisaniem pracy dyplomowej.		
Literatura: Określana indywidualnie do każdego tematu.		
Wymagania wstępne: znajomość przedmiotów niezbędnych do realizacji pracy dyplomowej.		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 4
Nazwa przedmiotu: <i>Seminarium</i>	Liczba godz. wykładu:	Liczba godz. ćwic./lab.: 30
<p>Cele przedmiotu: Głównym celem zajęć jest przygotowanie i ustalenie zakresu tematyki związanej z pracą dyplomową. Referaty głoszone przez studentów mają również za zadanie przygotowanie studenta do ewentualnej przyszłej pracy dydaktycznej i umiejętności wyboru i przekazywania informacji w wystąpieniach na forum publicznym .</p>		
<p>Zawartość programowa: Zakres materiału prezentowanego podczas referatów i dyskusji związana jest ściśle z tematyką prac magisterskich poszczególnych studentów.</p>		
<p>Literatura: Dobierana do indywidualnej tematyki prac magisterskich.</p>		
<p>Wymagania wstępne: (informacja o powiązaniu z wcześniejszymi przedmiotami) Ogólna wiedza zdobyta przez studenta podczas zajęć dydaktycznych prowadzonych na studiach.</p>		

INFORMATYKA (2-letnie studia II stopnia stacjonarne)	Specjalność:	rok/semestr: II / 4
Nazwa przedmiotu: <i>Specjalistyczna pracownia magisterska</i>	Liczba godz. wykładu:	Liczba godz. ćwic./lab.: 15
Cele przedmiotu: Celem przedmiotu jest udostępnienie specjalistycznego sprzętu i oprogramowania niezbędnego podczas pisania pracy magisterskiej. Pomoc nauczyciela akademickiego w rozwiązywaniu problemów napotkanych w trakcie realizacji tematu.		
Zawartość programowa: Praktyczna realizacja zagadnień związanych z pisaniem pracy dyplomowej.		
Literatura: Określana indywidualnie do każdego tematu.		
Wymagania wstępne: znajomość przedmiotów niezbędnych do realizacji pracy dyplomowej.		