

UNIWERSYTET W SIEDLCACH

INSTYTUT INFORMATYKI

Kierunek INFORMATYKA

INFORMATOR

SYLABUS

Obowiązuje w roku akademickim 2025/26

studia II stopnia

(magisterskie, profil praktyczny)

czas trwania: 3 semestry

Siedlce 2025

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Obliczenia naukowe i metody numeryczne
Nazwa w języku angielskim:	Computational and Numerical Methods in Science	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	2	
Imię i nazwisko koordynatora przedmiotu:		dr Anna Wawrzyńczak-Szaban
Imię i nazwisko prowadzących zajęcia:		dr Anna Wawrzyńczak-Szaban
Założenia i cele przedmiotu:		Zakłada się, że student posiada umiejętność rozwiązywania równań algebraicznych, układów równań liniowych i równań różniczkowych zwyczajnych oraz znajomość podstaw programowania. Celem przedmiotu jest zapoznanie studentów z podstawowymi zagadnieniami z zakresu obliczeń naukowych i metod numerycznych, które znajdują zastosowanie w rozwiązywaniu problemów inżynierskich i informatycznych. Studenci poznają wybrane modele matematyczne oraz metody numeryczne, ucząc się ich praktycznego wykorzystania w pracy informatyka.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	pojęcie matematycznego modelu procesu, etapy jego budowania oraz pojęcie przybliżonego rozwiązania wraz z określeniem błędu przybliżenia i jego własności.	K_W01
W_02	metody numerycznego rozwiązywania równań algebraicznych, układów równań liniowych, metody numeryczne przybliżania wartości pochodnych oraz całek oznaczonych oraz numerycznego rozwiązywania równań różniczkowych zwyczajnych.	K_W01
	UMIEJĘTNOŚCI Student potrafi:	
U_01	skonstruować model matematyczny procesu i wybrać odpowiednie dla danego zagadnienia metody numeryczne oraz je zaimplementować.	K_U01, K_U07
U_02	posługiwać się metodami wbudowanymi MatLab znajdującymi rozwiązanie równań algebraicznych, układów równań liniowych oraz równań różniczkowych zwyczajnych oraz wartości pochodnych oraz całek oznaczonych.	K_U01 K_U07

Forma i typy zajęć:	Studia stacjonarne: wykłady (20 godz.), ćwiczenia laboratoryjne (22 godz.) Studia niestacjonarne: wykłady (12 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:	
Umiejętność rozwiązywania równań algebraicznych, układów równań liniowych i równań różniczkowych zwyczajnych oraz znajomość podstaw programowania.	
Treści modułu kształcenia:	
• Pojęcia matematyki obliczeń naukowych. Matematyczny model procesu. Pojęcie przybliżonego rozwiązania zagadnienia. Etapy modelowania • Błędy obliczeń numerycznych. Błąd względny i bezwzględny. Podstawowe źródła błędów. Ogólna postać błędu. Zapis maszynowy liczby zmiennoprzecinkowej i błędy operacji na tych liczbach. Propagacja błędu. Typy zmiennych języka MatLab. • Metody przybliżonego rozwiązywania równań algebraicznych. Metoda bisekcji. Metoda interpolacji liniowej. Warunki zbieżności metod numerycznych. Metoda iteracji. Metoda Newtona- Rapsona. Porównanie zbieżności metod • Metody numerycznego rozwiązywania układów równań liniowych. Metoda eliminacji Gaussa. Dekompozycja LU. Metoda Crout'a (Thomasa). Funkcje wbudowane MatLab znajdowania pierwiastków równań algebraicznych. Metoda iteracji Jakobiego. Metoda iteracji Gaussa-Seidla. Metoda relaksacji • Interpolacja i aproksymacja zbioru danych eksperymentalnych. Interpolacja za pomocą wielomianów. Wzór interpolacyjny Lagrange'a. Interpolacja Newtona dla węzłów równoodległych. Interpolacja wielomianami sklejanymi.. Zasady doboru odpowiedniej funkcji aproksymującej. Określenie dokładności aproksymacji. Regresja liniowa. Aproksymacja wielomianem. Wbudowane funkcje MatLab. • Metody numeryczne przybliżania pochodnych. Wzór Taylora. Wzór Stirlinga. Różnice centralne. Funkcje wbudowane MatLab obliczania pochodnych funkcji. • Metody numeryczne przybliżania wartości całek oznaczonych. Wzór Prostokątów. Kwadratury Newtona-Cotesa. Metody Simpsona. Funkcje wbudowane MatLab obliczania wartości całek. Metoda Monte Carlo. • Metody jednokrokowe numerycznego rozwiązywania równań różniczkowych zwyczajnych rzędu pierwszego. Metoda Euler'a. Metoda Heun'a. Metoda Rungego - Kuty (rzęd 3,4,5). Metoda Rungego-Kutta-Fehlberga. Funkcje wbudowane MatLab ode45(), ode23(). • Metody wielokrokowe numerycznego rozwiązywania równań różniczkowych zwyczajnych rzędu pierwszego. • Metody predyktor-korektor numerycznego rozwiązywania równań różniczkowych zwyczajnych rzędu pierwszego.	
Literatura podstawowa:	
1. Krzyżanowski Piotr, Obliczenia inżynierskie i naukowe , PWN ,2016 2. Zenon Fortuna, Bohdan Macukow, Janusz Wąsowski , Metody numeryczne, PWN, Warszawa, 2017 3. B. Pańczyk, E. Łukasik, J. Sikora, T. Guziak Metody numeryczne w przykładach, Politechnika Lubelska, 2012	
Literatura dodatkowa:	
1. R. Pratap "MatLab 7 dla naukowców i inżynierów", PWN, 2007. 2. D. Kincaid, W. Cheney "Analiza numeryczna", WNT, 2006. 3. W.Y. Yang, W. Cao, T. Chung, J. Morris "Applied Numerical Methods Using MatLab", Wiley-Interscience, 2005	
Planowane formy/działania/metody dydaktyczne:	
Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko obliczeń naukowych MatLab.	
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
Symbol efektu	Metody weryfikacji efektów uczenia się

W_01	Efekt będzie weryfikowany na kolokwium pisemnym na ostatnim wykładzie.
W_02	Efekt będzie weryfikowany na kolokwium pisemnym na ostatnim wykładzie.
U_01	Efekt sprawdzany będzie na bieżąco, na każdych zajęciach poprzez implementacje w środowisku MatLab algorytmów rozwiązujących określone zadania
U_02	Efekt sprawdzany będzie na bieżąco, na każdych zajęciach poprzez implementacje w środowisku MatLab algorytmów rozwiązujących określone zadania

Forma i warunki zaliczenia:

Przedmiot kończy się zaliczeniem z oceną. Ocena końcowa ustalana jest na podstawie wyników uzyskanych podczas zajęć laboratoryjnych oraz pisemnego kolokwium końcowego z wykładu.

Warunkiem zaliczenia każdej z formy zajęć jest uzyskanie co najmniej 51% możliwych do zdobycia punktów.

Ocena końcowa wyliczana jest jako średnia ważona, w której:

60% stanowi suma punktów uzyskanych w trakcie zajęć laboratoryjnych,

40% stanowią punkty uzyskane z kolokwium pisemnego z wykładu.

Sumarycznie można zdobyć 100 punktów.

Ocena końcowa, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	20 godz.
Udział w ćwiczeniach laboratoryjnych	22 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	3 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do kolokwium z wykładu	2 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	2 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	12 godz.

Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	11 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do kolokwium	9 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	2 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Kierunki rozwoju informatyki
Nazwa w języku angielskim:	Directions of computer science development	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	2	
Imię i nazwisko koordynatora przedmiotu:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni
Imię i nazwisko prowadzących zajęcia:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni
Założenia i cele przedmiotu:		Zakłada się, że student potrafi korzystać z oprogramowania typu Office oraz środowisk typu Matlab, Sphinx, Statistica. Celem zajęć jest zapoznanie studentów z wybranymi kierunkami rozwoju informatyki, w tym informatyki kwantowej, a w szczególności kierunków rozwoju zastosowań informatyki
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	zagadnienia z zakresu rozwoju informatyki, w tym rozwoju zastosowań informatyki w robotyce, gospodarce, wojsku, nauce, rolnictwie, zarządzaniu, administracji, bankowości, itp. Potrafi wyróżniać kierunki rozwoju informatyki.	K_W08
W_02	zagadnienia z zakresu rozwoju komputeryzacji, informatyzacji, automatyzacji i robotyzacji w wybranych obszarach zastosowań informatyki w gospodarce, w szczególności w infrastrukturze technicznej, w tym elektroenergetyce, telekomunikacji, technologiach informacyjnych, itp.	K_W08
W_03	zagadnienia z zakresu rozwoju zastosowań informatyki w robotyce, w szczególności w zakresie robotyki przemysłowej, elastycznych systemów produkcyjnych i fabryk bezludnych	K_W08
W_04	zagadnienia z zakresu potrzeby rozwoju zastosowań nowych technologii informatycznych, w tym komputerów kwantowych, robotów humanoidalnych, robotów inteligentnych, itp.	K_W08

W_05	zagadnienia z zakresu rozwoju komputerowego wspomagania procesów, w tym m.in. CAD, CAM, CAQ, CAE, itp. oraz wykorzystania wielkich środowisk informatycznych (MATLAB, SAP, SAS, STATISTICA, itp.) w prowadzeniu prac typu Badania + Rozwój + Wdrożenia	K_W08
UMIEJĘTNOŚCI Student potrafi:		
U_01	formułować i opisywać inżynierskie kierunki rozwoju zastosowań informatyki	K_U01, K_U03
U_02	wykorzystywać wiedzę dotyczącą nowoczesnych środowisk informatycznych w specyfikacji nowych kierunków rozwoju zastosowań informatyki z punktu widzenia prowadzenia programów badawczych, rozwojowych i wdrożeniowych	K_U01, K_U10
U_03	porównywać i wyciągać wnioski z różnych sposobów rozwoju zastosowań informatyki w robotyce, gospodarce, rolnictwie, wojsku, medycynie, bankowości, energetyce, administracji państwowej, itp.	K_U09, K_U05
U_04	ocenić przydatność różnych środowisk programistycznych do projektowania, modelowania i symulacji nowych systemów, urządzeń, wyrobów, itp.	K_U08
KOMPETENCJE SPOŁECZNE Student jest gotów do:		
K_01	Współdziałania w grupie i ma świadomość wagi zachowania się w sposób profesjonalny (inżynierski) z skłonnością do przejawiania inicjatywy oraz przestrzegania zasad etyki zawodowej.	K_K01
K_02	wykorzystywania nowych kierunków rozwoju zastosowań informatyki w pracy innowacyjnej i twórczej.	K_K02
Forma i typy zajęć:	Studia stacjonarne: wykłady (20 godz.), Zajęcia seminaryjno-laboratoryjne (20 godz.) Studia niestacjonarne: wykłady (12 godz.), Zajęcia seminaryjno-laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
- Umiejętność samodzielnego korzystania z oprogramowania użytkowego typu MS Office. - Umiejętność korzystania z wielkich środowisk programistycznych typu: MATLAB, SPHINX, STATISTICA.		
Treści modułu kształcenia:		
Rozwój robotyki. Generacje robotów, roboty przemysłowe, roboty mobilne, roboty humanoidalne, itp., Rozwój manipulatorów przemysłowych i robotów mobilnych, Elastyczne systemy produkcyjne, Fabryki bezludne, Modelowanie i symulacja robotów, Prawa robotyki, stopnie swobody, itp., Elastyczne stacje i systemy obróbkowe, Elastyczne systemy produkcyjne, Komputerowa integracja wytwarzania, Inteligentne sterowanie, rozwój robotyki, itp. Współczesne metody modelowanie i symulacja sterowania systemami wielkimi. Modelowanie i symulacja systemów funkcjonujących w czasie rzeczywistym, z wykorzystaniem środowiska MATLAB i Simulink, SPHINX, SAS, SAP, itp., Podstawy systemów wielkich takich jak: Parki technologiczne, Centra handlowe, Systemy bezludne (porty lotnicze, porty morskie, centra logistyki kolejowej i drogowej, a także węzły informacyjne, energetyczne, paliwowe, handlowe i rozrywkowe, itp.), Metody sztucznej inteligencji w sterowaniu systemami wielkimi, itp., Od manipulatorów, robotów i elastycznych systemów do fabryk bezludnych. Przykłady współczesnych fabryk bezludnych, Model systemu elektroenergetycznego jako fabryki bezludnej, Ewolucja systemu elektroenergetycznego w Polsce, Problemy przyszłości elektroenergetyki,		

Rozbudowa systemów informatycznych w dziedzinach wytwarzania, przesyłu, odbioru i przetwarzania energii elektrycznej, Smart metering, Smart Power System, Smart Grid, itp.

- **Nowoczesne systemy komputerowego wspomagania.** Sposoby komputerowego wspomagania (projektowania, wytwarzania, działania człowieka, systemu, procesu, obiektu, itp.), Środowiska typu: CAD, CAM, CAE, CAQ, itp., Systemy inteligentne we wspomaganiu działań ludzkich, itp., Inteligentny samochód, Inteligentna maszyna, Inteligentny system, Inteligentny budynek, Inteligentny sprzęt domowy, Inteligentne zabawki, itp.
- **Rozwój systemów informatycznych w medycynie, w tym systemów nanoinformatyki.** Informatyka medyczna, Modele informacyjne jednostek opieki medycznej, Akwizycja danych medycznych, Przetwarzanie danych obrazowych, Medyczne systemy doradcze, Medyczne systemy informacyjne, Sztuczne organy: serce, sztuczna nerka, sztuczna wątroba, sztuczna noga, sztuczna ręka, sztuczne oko, itp., Sterowanie myślą, itp.
- **Współczesne informatyczne systemy w administracji rządowej i samorządowej, w tym w administracji skarbowej i celnej.** Informatyka biurowa, Informatyka transakcyjna, Informatyka edukacyjna, Etapy rozwoju informatyki skarbowej, finansowej i celnej, cechy systemów skarbowych, eDeklaracje, POLTAX, POLTAX 2B, CELINA, ZEFIR, itp.,
- **Perspektywy zastosowań informatyki w gospodarce i w społeczeństwie, w tym w rachunkowości i finansach oraz w bankach.** Informatyka przemysłowa, Informatyka rolna, Informatykę ludyczną, Informatyka rynkowa, Informatyka ekspercka, Informatyka zarządcza, Systemy klasy business intelligence, informatyczne systemy zarządzania, inteligentne systemy transportowe, współczesne systemy obsługi finansowej, Etapy rozwoju informatyki bankowej, Cechy systemów bankowych, Klasyfikacja systemów bankowych, Operacyjne systemy zorientowane na księgowość, produkty, klienta, Architektura bankowych systemów informatycznych, Modele scoringowe, itp.
- **Rozwój komputerów kwantowych.** Informatyka kwantowa, Komputery kwantowe i ich generacje, Komputery jednokubitowe i komputery wielokubitowe, Realizacje praktyczne komputerów kwantowych, Rozwój komputerów kwantowych, Rejestr kwantowy, Bramki kwantowe, Obwody kwantowe, itp., Kubit, w tym kubit, interpretacje kubit, stany systemu i stany kwantowe, kwantowe stany mieszane, superpozycja stanów wykluczających się, pomiar (odczyt) stanu kwantowego, ogólne operacje kwantowe, itp.
- **Wyzwania dla obliczeń kwantowych na bramkach kwantowych.** Przestrzeń Hilberta, w tym macierze unitarne, postulaty informatyki kwantowej, Obliczenia kwantowe z wykorzystaniem bramek kwantowych, Liczby zespolone w obliczeniach kwantowych, Obwody kwantowe w obliczeniach kwantowych, Kwantowy stan splątany, Teleportacja kwantowa, Ewolucja unitarna, rozkład Schmidta stanów kwantowych, itp.
- **Nowoczesna informatyka kwantowa, w tym metody sztucznej inteligencji inspirowane kwantowo.** Kwantowe przetwarzanie informacji, Rola operatorów w informatyce kwantowej, w tym wartości własnych operatorów hermitowskich, Macierz gęstości, w tym wyznaczanie śladu, Funkcja falowa, Obserwable, Synteza obwodów kwantowych, Algorytmy kwantowe, itp., kwantowo inspirowane sztuczne sieci neuronowe, kwantowo inspirowane algorytmy ewolucyjne, kwantowo inspirowane inne metody sztucznej inteligencji, przykłady obliczeniowe, schemat blokowy obliczeń kwantowych w Simulinku, itp.

Literatura podstawowa:

1. R. Rudowski: Informatyka medyczna, WN PWN, Warszawa 2020.
2. J. Tchórzewski: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów. Wydawnictwo UPH, Siedlce 2021
3. Programy i raporty państwowe, np. Program Zintegrowanej Informatyzacji Państwa [online:] <https://www.gov.pl/web/cyfrizacja/program-zintegrowanej-informatyzacji-panstwa> [dostęp: 22.06.2025], Raport NIK pt. „Działania organów administracji publicznej na rzecz podnoszenia kompetencji cyfrowych społeczeństwa”, Warszawa 2022.

Literatura dodatkowa (przykłady):

1. Raporty GUS o stanie rozwoju informatyki w Polsce, np. Społeczeństwo informacyjne w Polsce w 2024 r., GUS, Warszawa - Szczecin, 2024
2. Czasopisma informatyczne, np. Studia Informatica. Systems and Information Technology, Theoretical and Applied Informatics, Computer Science, e-Informatica Software Engineering Journal, itp.

3. Raporty Polskiego Towarzystwa Informatycznego, Polskiej Izby Telekomunikacji i informatyki, Ministerstwa Nauki i Szkolnictwa Wyższego, itp.

Planowane formy/działania/metody dydaktyczne

Wykład wspomagany technikami multimedialnymi z aktywną komunikacją ze studentami. Przewiduje się przygotowanie na zaproponowany przez studenta temat z zakresu wiedzy objętej wykładem: konspektów i prezentacji, a następnie po ich wygłoszeniu na zajęciach laboratoryjno-seminaryjnych przygotowanie referatów, a w przypadku ich wyróżnienia możliwość ich publikacji.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia w się
W_01	Efekt będzie sprawdzany jednokrotnie na sprawdzianie pisemnym. Przykładowe pytanie: Scharakteryzować kierunki rozwoju informatyki w robotyce humanoidalnej, wymień przynajmniej trzy projekty dotyczące realizacji robotów humanoidalnych, w tym jeden projekt polski.
W_02	Efekt będzie sprawdzany jednokrotnie na sprawdzianie pisemnym. Przykład pytania: Scharakteryzować pojęcia systemu inteligentnego (smart system) w gospodarce. Omówić funkcjonowanie systemu elektroenergetycznego jako inteligentnej fabryki bezludnej, itp.
W_03	Efekt będzie realizowany jednokrotnie na sprawdzianie pisemnym. Przykład pytania: Wymienić współczesne zagadnienia rozwoju zastosowań informatyki w robotyce. Omówić kierunki rozwoju robotyki przemysłowej.
W_04	Efekt będzie sprawdzany jednokrotnie na sprawdzianie pisemnym. Przykład pytania: Wymienić możliwości inspiracji kwantowych przynajmniej trzech metod sztucznej inteligencji. Opisać koncepcję inspiracji kwantowych sztucznych sieci neuronowych.
W_05	Efekt będzie sprawdzany jednokrotnie na sprawdzianie pisemnym. Przykład pytania: Wymienić trzy środowiska informatyczne z zakresu komputerowego wspomagania procesów. Omówić w tym zakresie możliwości środowiska MATLAB i Simulink.
U_01	Efekt będzie sprawdzany jednokrotnie na zajęciach laboratoryjno-seminaryjnych. Przykład zadania do realizacji: Opracować konspekt własnego zadania badawczego polegającego na określeniu kierunków rozwoju informatyki w uzgodnionym zakresie, np. sformułować wprowadzenie do swoich badań podając: tło badań KRI w przemyśle, potrzeby i możliwości KRI, wyspecyfikować przynajmniej 5 problemów badawczych, sformułować cel, tezę, motywację, opisać treść podstawowych pięciu KRI w przemyśle, itp.
U_02	Efekt będzie sprawdzany jednokrotnie na zajęciach laboratoryjno-seminaryjnych. Przykład zadania do realizacji: Opracować prezentację na temat Kierunków rozwoju informatyki w społeczeństwie zgodną z wcześniej opracowanym konspektem. Sformułować przynajmniej 5 wniosków wynikających z przeprowadzonego krytycznego przeglądu literatury przedmiotu dla realizowanego zadania na bazie przynajmniej 5 artykułów w j. angielskim oraz 5 artykułów w j. polskim z ostatnich pięciu lat.
U_03	Efekt będzie sprawdzany jednokrotnie na zajęciach laboratoryjno-seminaryjnych. Przykład zadania do realizacji: Wygłosić referat z wykorzystaniem wcześniej przygotowanej prezentacji na Seminarium z Kierunków Rozwoju Informatyki, przeprowadzić dyskusję z publicznością oraz wyciągnąć stosowne wnioski w zakresie prowadzenia dalszych badań w tym zakresie.
U_04	Efekt będzie realizowany jednokrotnie na zajęciach laboratoryjno-seminaryjnych. Przykład zadania do realizacji: Opracować artykuł według ustalonych wymagań profesjonalnych do wybranego czasopisma na temat ustalonej tematyki np. Kierunki Rozwoju Informatyki w administracji państwowej. Zwrócić uwagę na ocenę przydatności różnych środowisk programistycznych do projektowania, modelowania i symulacji nowych systemów, urządzeń, wyrobów, itp. opublikowanych w literaturze przedmiotu z ostatnich pięciu lat.

K_01	Efekt będzie sprawdzany przy każdym kontakcie ze studentem. Sprawdzana będzie aktywność studenta na wykładach, zwłaszcza w zakresie powtarzania i podsumowywania wykładów oraz omówienia struktury konspektów, prezentacji i artykułów. Na zajęciach laboratoryjno-seminaryjnych będzie sprawdzana jego umiejętność współdziałania w grupie, w tym w zakresie jego zachowania się w sposób inżynierski (sposób prezentacji własnej tematyki, udział w dyskusji podczas wystąpień innych studentów, itp.
K_02	Efekt będzie sprawdzany podczas każdego kontaktu ze studentem, a w szczególności na podstawie realizacji zadania indywidualnego, w tym jego prezentacji na Seminarium oraz na podstawie napisanego artykułu.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem na ocenę. Ocena końcowa jest średnią z oceny uzyskanej ze sprawdzianu pisemnego przeprowadzonego w postaci testu lub zadania problemowego na ostatnich zajęciach wykładowych oraz z oceny uzyskanej z zajęć laboratoryjnych. Na zaliczenie zajęć laboratoryjnych składają się oceny cząstkowe uzyskane z ustalonej z prowadzącym tematyki samodzielnie przygotowanej przez każdego studenta w postaci pisemnej, z których można uzyskać maksymalną liczbę punktów odpowiednio z: konspektu – 5 pkt., referatu – 15 pkt., tekstu prezentacji – 15 pkt., wygłoszenia prezentacji – 10 pkt i jego obrony na zorganizowanym na zajęciach Seminarium z Kierunków Rozwoju Informatyki – 5 pkt. W przypadku najlepiej opracowanej tematyki studenci zostaną wyróżnieni możliwością przygotowania profesjonalnego artykułu do renomowanych czasopism krajowych i zagranicznych. Ponadto będą dodatkowo punktowane m.in. aktywność studentów na wykładach oraz podejmowana dyskusja na zajęciach seminaryjnych.

Ocena z zajęć seminaryjno-laboratoryjnych oraz z sprawdzianu pisemnego, w zależności od sumy uzyskanych punktów może być następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Zaliczenie zajęć seminaryjno-laboratoryjnych oraz odrębnie sprawdzianu pisemnego następuje w przypadku uzyskania co najmniej 51 pkt. Przewiduje się dodatkowe punkty z tytułu aktywności studenta na wykładach, dyskusji na seminariach, napisania artykułu i wyróżnienie studenta po uzyskaniu co najmniej 101 pkt. z przedmiotu w drodze możliwości przygotowania artykułu pod kierunkiem prowadzącego do druku w czasopiśmie naukowym, popularnonaukowym, bądź w wersji elektronicznej na stronie kół naukowych funkcjonujących na kierunku informatyka na Wydziale Nauk Ścisłych i Przyrodniczych, z adnotacją o wyróżnieniu wygłoszonego referatu na zajęciach z Kierunków rozwoju informatyki.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	20 godz.
Udział w zajęciach laboratoryjno-seminaryjnych	20 godz.
Samodzielne przygotowanie się do zajęć laboratoryjno-seminaryjnych, w tym przygotowanie konspektu, prezentacji, artykułu, itp.	5 godz.
Udział w konsultacjach	1 godz.
Przygotowanie się do sprawdzianu z wykładów i zajęć laboratoryjno-seminaryjnych	4 godz.
Sumaryczne obciążenie pracą studenta	50 godz.

Punkty ECTS za przedmiot	2 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	12 godz.
Udział w zajęciach laboratoryjno-seminaryjnych	15 godz.
Samodzielne przygotowanie się do zajęć laboratoryjno-seminaryjnych, w tym przygotowanie konspektu, prezentacji, artykułu, itp.	15 godz.
Udział w konsultacjach	1 godz.
Przygotowanie się do sprawdzianu	7 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	2 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Modelowanie i analiza systemów informatycznych
Nazwa w języku angielskim:		Modeling and Analysis of Information Systems
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Marek Pilski
Imię i nazwisko prowadzących zajęcia:		dr Marek Pilski
Założenia i cele przedmiotu:		Założeniem przedmiotu jest znajomość przez słuchaczy zagadnień związanych z analizą wymagań na oprogramowanie oraz modelowaniem systemów informatycznych. Celem zajęć jest zapoznanie studentów z wybranymi zagadnieniami związanymi z modelowaniem systemów informatycznych w podejściu strukturalnym i obiektowym.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	czym jest model systemu. Wie po co budować modele systemu. Zna rodzaje modeli oraz kto jest ich odbiorcami. Zna języki opisu modeli.	K_W04, KW_10
W_02	zasady modelowania systemów informatycznych w podejściu strukturalnym i obiektowym.	K_W04
W_03	rolę narzędzi informatycznych oraz ich funkcjonalność we wspomaganiu analizy i modelowania systemów informatycznych.	K_W04
W_04	podstawowe pojęcia w modelowaniu i analizie systemu informacyjnego	K_W04
W_05	sposoby pozyskiwania i zapisywania wymagań dotyczących oprogramowania.	K_W04
	UMIEJĘTNOŚCI Student potrafi:	
U_01	wykorzystać metody podejścia obiektowego do analizy i tworzenia modelu przykładowego systemu informatycznego.	K_U10, K_U11
U_02	wykorzystać metody podejścia strukturalnego do analizy i tworzenia modelu przykładowego systemu informatycznego.	K_U10, K_U11
U_03	wykorzystać odpowiednie narzędzia typu CASE do modelowania systemów informatycznych.	K_U01, K_U11
U_04	zapisać wymagania na oprogramowanie w języku/notacji graficznej.	K_U03, K_U10, K_U11
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	kreatywnej i krytycznej analizy wymagań dotyczących oprogramowania i przekazywania w sposób zrozumiały własnych opinii.	K_01

Forma i typy zajęć:	Studia stacjonarne: wykłady (20 godz.), ćwiczenia laboratoryjne (20 godz.); Studia niestacjonarne: wykłady (12 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:	
Znajomość zagadnień związanych z procesami biznesowymi instytucji gospodarczych, zasad i reguł inżynierii oprogramowania.	
Treści modułu kształcenia:	
Treści dotyczące wykładu:	
- Modelowanie SI. Podstawowe pojęcia: informacja, system, system informacyjny, system informatyczny, model. Czym jest model a czym nie jest. Rola modelu. Rodzaje modeli. Odbiorcy modeli. Języki opisu modeli. - Rodzaje wymagań oprogramowania. Wymagania funkcjonalne. Techniki opisu wymagań funkcjonalnych. Wymagania niefunkcjonalne. Techniki opisu wymagań niefunkcjonalnych. Weryfikacja wymagań. Metody pozyskiwania i dokumentowania wymagań. Szablony dokumentów opisu wymagań. - Podjęcie strukturalne. Modelowanie procesów. Co to jest proces, cele modelowania procesu. Przykłady procesów, podejście do modelowania procesów, komponenty procesu (jednostka organizacyjna, wyzwalacz, wynik, krok procesu, przepływ, składnica), dekompozycja procesów. - Podjęcie strukturalne. Modelowanie wymagań na dane. Cele modelowania związków encji. Diagramy związków encji i ich komponenty, encje, atrybuty, związki. Dziedziny, unikalny identyfikator dla encji, konstrukcje specjalne związków encji. Zasady konstruowania modeli związków encji. Miejsce modeli związków encji w procesie wytwórczym oprogramowania. - Podjęcie strukturalne. Modelowanie przepływu danych. Cele modelowania przepływu danych. Hierarchia modelu. Komponenty modelu przepływu danych (proces przepływ danych, składnica danych, byt zewnętrzny). Zasady modelowania przepływu danych. - Modelowanie hierarchii funkcji. Cele modelowania hierarchii funkcji. Funkcje biznesowe. Hierarchie funkcji. Edytory Diagramów Hierarchii Funkcji. Użycia danych: użycia funkcja-encja, użycia funkcja-atrybut. Granica automatyzacji funkcji. Modelowanie zależności funkcji. Cele modelowania zależności funkcji. Zależności pomiędzy funkcjami. Komponenty diagramu zależności funkcji. Weryfikacja zależności funkcji. - Analiza modelu systemu informatycznego. Macierze CRUD, IRON. Narzędzia i techniki analizy. Wykorzystanie modeli. Sprawdzenie jakości analizy. Analiza a projektowanie. - Podjęcie obiektowe. Język UML. Przegląd diagramów języka UML: struktury i zachowania. systemu informatycznego. Stereotypy w UML. - Metodyki wytwarzania oprogramowania. Podział metodyk (formalne, lekkie). Wybrane metody wytwarzania oprogramowania, rola i miejsce fazy analizy i modelowania SI w kontekście wybranej metodyki. Charakterystyka wybranych metodyk: RUP, Open Up, XP, Scrum i artefakty związane z modelowaniem. - Przegląd narzędzi CASE wspierających analizę i modelowanie systemów informatycznych.	
Treści dotyczące laboratorium:	
- Pozyskiwanie wymagań na oprogramowanie. - Modelowanie procesów.	

- Modelowanie związków encji.
- Modelowanie przepływu danych.
- Modelowanie hierarchii funkcji.
- Opracowanie zadania indywidualnego.
- UML. Modelowanie zachowania systemu. Diagramy przypadków i aktywności.
- UML. Modelowanie zachowania systemu. Diagramy sekwencji i komunikacji.
- UML. Modelowanie statyki systemu. Diagramy klas i obiektów.
- UML. Modelowanie statyki systemu. Diagramy pakietów.

Literatura podstawowa:

1. Dąbrowski W., Stasiak A., Wolski M.: Modelowanie systemów informatycznych w języku UML 2.1 w praktyce, PWN, Warszawa 2009
2. Brett D. McLaughlin, Gary Pollice, David West.: Analiza i projektowanie obiektowe, Helion, Gliwice 2010

Literatura dodatkowa:

1. Sharon Allen: Modelowanie danych. Helion, Gliwice, 2006.
2. Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski.: Język UML 2.0 w modelowaniu systemów informatycznych: diagramy języka UML, modelowanie biznesowe, metodyki projektowe oparte na UML, narzędzia CASE. Wydawnictwo Helion, Gliwice 2005
3. Wolski M., Piłski M., Project Estimation Using the Use Case Points Method When Using Enterprise Architect, Studia Informatica. Systems and information technology, volume 1-2(14)2010, Publishing House of University of Podlasie, Siedlce 2010

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratoria wspomagane technikami komputerowymi (wybrane narzędzia CASE). Zamieszczanie na stronach internetowych problemów i zadań laboratoryjnych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Efekt realizowany na egzaminie w formie pisemnej. Pytanie otwarte lub test wiedzy. Przykładowe pytanie: Czym jest model systemu informatycznego i opisz jego rolę.
W_02	Efekt realizowany na egzaminie w formie pisemnej. Pytanie otwarte lub test wiedzy. Przykładowe pytanie: Omów cele i komponenty modelowania procesów.
W_03	Efekt realizowany na egzaminie w formie pisemnej. Pytanie otwarte lub test wiedzy. Przykładowe pytanie: Opisz charakterystyczne cechy narzędzi CASE.
W_04	Efekt realizowany na egzaminie w formie pisemnej. Pytanie otwarte lub test wiedzy. Przykładowe pytanie: Co to jest Encja, podaj przykłady. Co to jest Aktor, podaj przykłady.
W_05	Efekt realizowany na egzaminie w formie pisemnej. Pytanie otwarte lub test wiedzy. Przykładowe pytanie: Wymień i krótko opisz metody pozyskiwania wymagań na oprogramowanie.
U_01	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji. Przykładowe zadanie: Na diagramie

	przypadków użycia przedstaw funkcję „Wypożycz auto”, która opcjonalnie może wymagać realizacji funkcji „Zawrzyj umowę z Klientem” dla aktora: Handlowiec.
U_02	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji. Przykładowe zadanie: Przedstaw w notacji graficznej wymaganie: Każda Linia autobusu musi zatrzymywać się na jednym lub więcej Przystankach natomiast Każdy Przystanek może być przypisany dla jednej lub więcej Linii autobusowych.
U_03	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji. Przykładowe zadanie: Opracuj model związków encji dla podsystemu Sprzedaży Biletów wykorzystując poznane narzędzie typu CASE.
U_04	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji. Przykładowe zadanie: Na diagramie Hierarchii Funkcji przedstaw wymagania funkcjonalne dla podsystemu Sprzedaży Biletów.
K_01	Efekt realizowany w trakcie zajęć laboratoryjnych. Postaw się w roli analityka systemu informatycznego i zapisz w formalny sposób wymagania dla Systemu Zarządzania Transportem Miejskim dla modułu Sprzedaży biletów. Wymagania mają spełniać cechy m.in. cechy możliwości ich weryfikacji i wykonalności.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Ocena z przedmiotu składa się z dwóch ocen częściowych: 1. oceny z zajęć laboratoryjnych, 2. oceny z egzaminu w formie pisemnej. Na ocenę z zajęć laboratoryjnych składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim 5 pkt za każde laboratorium, za które można uzyskać sumarycznie 50 pkt. Zaliczenie zajęć laboratoryjnych możliwe jest po uzyskaniu co najmniej 51% liczby punktów z tej formy zaliczenia. Zaliczenie wykładu w formie egzaminu nastąpi w przypadku uzyskania co najmniej 51% liczby z puli 50 punktów przewidzianej dla tej formy zaliczenia. Sumarycznie można zdobyć 100 punktów. Ocena końcowa, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS): • 0 – 50 pkt: niedostateczna (F), • 51 – 60 pkt: dostateczna (E), • 61 – 70 pkt: dostateczna plus (D), • 71 – 80 pkt: dobra (C), • 81 – 90 pkt: dobra plus (B), • 91 – 100 pkt: bardzo dobra (A).	
Bilans punktów ECTS:	
Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	20 godz.

Udział w ćwiczeniach laboratoryjnych	20 godz.
Przygotowanie się do egzaminu	17 godz.
Udział w egzaminie	2 godz.
Przygotowanie się do zajęć laboratoryjnych	14 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	3
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	12 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Przygotowanie się do egzaminu	23 godz.
Udział w egzaminie	2 godz.
Przygotowanie się do zajęć laboratoryjnych	23 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	3

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Sieci i systemy wirtualne
Nazwa w języku angielskim:	Virtual Networks and Systems	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski
Imię i nazwisko prowadzących zajęcia:		dr Grzegorz Terlikowski, mgr Wojciech Nabiałek
Założenia i cele przedmiotu:		Zakłada się, że student ma podstawowe umiejętności związane z budowaniem sieci komputerowych. Celem kursu jest uzyskanie wiedzy i nabycie przez studentów umiejętności związanych z zarządzaniem sieciami i systemami wirtualnymi oraz z podstawami ich tworzenia
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	zagadnienia dotyczące sieci i systemów wirtualnych.	K_W03, K_W09
W_02	zagadnienia na temat zasad działania różnych protokołów służących do budowy sieci VLAN i VPN oraz zagadnienia z zakresu działania elementów wchodzących w skład infrastruktury służącej do wirtualizacji.	K_W03, K_W07, K_W09
	UMIEJĘTNOŚCI Student potrafi:	
U_01	pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji i krytycznej oceny, a także wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie.	K_U01, K_U10
U_02	dobierać i wykorzystywać oprogramowanie służące do wirtualizacji.	K_U01, K_U07, K_U9, K_U10
U_03	ocenić i wykorzystać protokoły i narzędzia służące do tworzenia sieci wirtualnych	K_U01, K_U07
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01, K_K03
Forma i typy zajęć:		Studia stacjonarne: wykłady (20 godz.), ćwiczenia (20 godz.), Studia niestacjonarne: wykłady (12 godz.), ćwiczenia (15 godz.)

Wymagania wstępne i dodatkowe:

Umiejętność konfigurowania sieci komputerowych.

Treści modułu kształcenia:

Wykłady:

- Wirtualne sieci lokalne. Podstawowe definicje. Metody tworzenia sieci VLAN. Protokół Dot1Q i znakowanie ramek. Różnice w podejściu klasycznym do budowania sieci lokalnych a podejściu bazującym na VLANach.
- Wirtualne sieci lokalne c.d. Protokoły VTP i DTP. Domena VTP. Prywatne sieci VLAN (PVLAN). Dobre praktyki i bezpieczeństwo sieci.
- Prywatne sieci wirtualne. Rodzaje prywatnych sieci wirtualnych. Zasada działania i elementy infrastruktury.
- Prywatne sieci wirtualne c.d. Przegląd protokołów VPN. Protokoły IPSec, PPTP, L2TP, SSTP, OpenVPN i inne.
- Wirtualizacja serwerów i jej typy. Historia maszyn wirtualnych: IBM System/360, IBM System/370. Podstawowe pojęcia: Wirtualizacja, instrukcje uprzywilejowane, hipervisor. Kryteria wirtualizacji. Typy wirtualizacji: natywna, parawirtualizacja, wirtualizacja wspierana sprzętowo, wirtualizacja aplikacji. Zastosowania wirtualizacji.
- Konteneryzacja. Definicja i porównanie z podejściem klasycznym do wirtualizacji. Przegląd oprogramowania: Docker, Kubernetes i inne rozwiązania.
- Pełna wirtualizacja na przykładzie VMware vSphere. Serwery wirtualne VMware ESXi. Usługi vMotion, vStorage, Web Access, Fault Tolerance, High Availability. Zarządzanie infrastrukturą – vCenter Server. Magazyny danych a macierze dyskowe. Przełączniki wirtualne: standardowe i rozproszone przełączniki. Wirtualne porty. Metody znakowania ramek przez wirtualne przełączniki: EST, VST, VGT.
- Platformy wykorzystujące Kubernetes jako orkiestratora kontenerów – VMware vSphere i Kubernetes, Red Hat OpenShift i inne.
- Przegląd pozostałych rozwiązań. HyperV, VirtualBox, QEMU, KVM i inne.
- JVM – Java Virtual Machine. Założenia przyjęte przy tworzeniu maszyny wirtualnej. Fazy uruchamiania programu na maszynie wirtualnej. Faza ładowania: lista czynności wykonywanych przez ClassLoader, Faza linkowania. Kompilacja warstwowa.

Ćwiczenia:

1. Konfigurowanie sieci wirtualnych VLAN. Zarządzanie VLANami, konfiguracja portów, tworzenie łączy TRUNK.
2. Konfigurowanie sieci wirtualnych VLAN cd. Protokół DTP i VTP, konfiguracja usługi DHCP do obsługi VLAN-ów, VLAN zarządzający.
3. Konfigurowanie sieci VPN typu site-2-site na routerach Cisco. Przygotowanie sieci w zadanej topologii, włączenie technologii bezpieczeństwa na wybranych routerach, konfigurowanie routerów do obsługi sieci VPN typu site-2-site, weryfikacja działania sieci VPN.
4. Konfigurowanie sieci VPN typu Remote-access na routerach Cisco. Przygotowanie infrastruktury, aktywowanie usługi AAA na routerach Cisco, konfigurowanie routerów do obsługi sieci VPN typu remote-access, weryfikacja działania sieci VPN.

5. Konfigurowanie VPN typu site-2-site w wariancie GRE over IPSec + NAT. Przygotowanie sieci w zadanej topologii, konfigurowanie routerów do obsługi NAT, konfigurowanie GRE na routerach.
6. Tunele VPN z wykorzystaniem oprogramowania OpenVPN. Tworzenie certyfikatów i kluczy, Konfiguracja OpenVPN w trybie serwera i w trybie klienta,
7. Konteneryzacja Docker. Instalacja środowiska Docker, uruchomienie i konfiguracja gotowego obrazu serwera nginx w środowisku Docker. Mapowanie portów i zasobów dyskowych
8. Docker compose. Budowanie własnych obrazów na bazie istniejących, tworzenie sieci wirtualnej w środowisku Docker, budowanie zależności pomiędzy elementami wirtualnej infrastruktury
9. Wdrażanie aplikacji w K8s . Uruchamianie klastra kubernetes. Konfigurowanie deploymentu. Konfigurowanie usług typu ClusterIP, Node Port i innych.
10. Obrona projektów indywidualnych. Testowanie i weryfikacja poprawności realizacji projektu indywidualnego. Możliwe niewielkie modyfikacje, w celu weryfikacji samodzielności jego wykonania.

Literatura podstawowa:

1. C. Wahl, S. Pantol, VMware dla administratorów sieci komputerowych. Helion 2015.B. Hartpence.
2. M. Serafin. Sieci VPN. Zdalna praca i bezpieczeństwo danych. Wydanie II rozszerzone. Wydawnictwo Helion. 2013.
3. M. Serafin. Wirtualizacja w praktyce. Wydawnictwo Helion. 2012.

Literatura dodatkowa:

1. Wirtualne serwery na bazie oprogramowania VMware GSX/ESX wspomaganego przez VMware VirtualCenter - Dariusz Daniluk , Warszawa, czerwiec 2006r.
2. Routing i switching. Praktyczny przewodnik. B. Hartpence. Wydawnictwo Helion. 2013.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia wspomagane technikami multimedialnymi. Zamieszczanie na stronach internetowych problemów i zadań ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Sprawdzone podczas egzaminu w formie pisemnej. Przykładowe pytania: Podaj definicję wirtualizacji oraz scharakteryzuj jej typy. Wymień i scharakteryzuj kryteria wirtualizacji, zdefiniowane przez Goldberga i Popka.
W_02	Sprawdzone podczas egzaminu w formie pisemnej. Przykładowe pytania: Wymień i scharakteryzuj 5 usług rozproszonych wchodzących w skład infrastruktury VMware vSphere. Wymień i scharakteryzuj elementy infrastruktury budującej sieć VPN typu Client to Site.
U_01	Systematycznie sprawdzane na zajęciach. Zadania na następne laboratorium muszą być dostępne tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować. Przykładowe zadanie: Zapoznaj się z dokumentacją Docker, a następnie uruchom w kontenerze obraz nginx.
U_02	Systematycznie sprawdzane na zajęciach. Przykładowe zadania: Zapoznaj się z oprogramowaniem OpenVPN, a następnie wygeneruj certyfikaty w celu zestawienia połączenia VPN pomiędzy klientem a serwerem trybie TUN a potem TAP.

	Uruchom klaster Kubernetes w Docker Desktop a następnie sprawdź jego status za pomocą narzędzia kubectl.
U_03	Systematycznie sprawdzane na zajęciach. Przykładowe zadanie: Zapoznaj się z protokołami .dot1q, VTP i DTP. Zbuduj prostą sieć zbudowaną z trzech przełączników wykorzystującą te protokoły. Zweryfikuj poprawność działania sieci.
K_01	Weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, podczas zaliczania zadania indywidualnego, a także będą sprawdzane na egzaminie.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i egzaminu. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim.

- Regularne zajęcia – 39 pkt.,
- Obrona zadania indywidualnego – 21 pkt.

W trakcie zajęć laboratoryjnych student może uzyskać maksymalnie 60 pkt. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 20 pkt., obrona indywidualnego zadania – co najmniej 11 pkt. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt. Za egzamin można uzyskać do 40 pkt. Zaliczenie egzaminu jest możliwe po uzyskaniu co najmniej 20 pkt.

Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Poprawy:

- Jednorazowa poprawa 2 wybranych przez studenta laboratoriów w trakcie konsultacji.
- Jednorazowa poprawa projektu indywidualnego w sesji egzaminacyjnej.

Jednorazowa poprawa kolokwium w sesji egzaminacyjnej.

Bilans punktów ECTS:

Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	20 godz.
Udział w ćwiczeniach	20 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	16 godz.
Samodzielna praca nad projektem indywidualnym	6 godz.

Samodzielne przygotowanie się do egzaminu	11 godz.
Udział w egzaminie	1 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	12 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	21 godz.
Samodzielna praca nad projektem indywidualnym	10 godz.
Samodzielne przygotowanie się do egzaminu	15 godz.
Udział w egzaminie	1 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Metody inteligentnej optymalizacji
Nazwa w języku angielskim:	Intelligent optimization methods	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Anna Wawrzyńczak-Szaban
Imię i nazwisko prowadzących zajęcia:		dr Anna Wawrzyńczak-Szaban dr Mirosław Szaban
Założenia i cele przedmiotu:		Zakłada się, że student zna podstawy programowania, programowania obiektowego i sztucznej inteligencji.. Celem przedmiotu jest nabycie przez studentów aktualnego stanu wiedzy dotyczącej metod inteligentnych w optymalizacji także systemów agentowych i wieloagentowych w szczególności ich inteligentnych zachowań realizowanych przez algorytmy inspirowane Naturą. Podczas laboratoriów studenci nabędą umiejętności projektowania, implementacji i stosowania algorytmów inspirowanych Naturą w stopniu wystarczającym do stosowania ich do poszukiwania rozwiązań w dużych przestrzeniach danych.
Symbol efektu	Efekty uczenia się	
	WIEDZA Student zna i rozumie:	
W_01	standardy i różne rodzaje algorytmów inspirowanych procesami zachodzącymi w Naturze. Zna standardy i problemy optymalizacyjne oraz transportowe, w których może być zastosowany agent z inteligencją opartą na algorytmach inspirowanych Naturą.	K_W01, K_W02, K_W03, K_W07
W_02	definicję agenta, systemów agentowych i wieloagentowych, inteligentnych systemów agentowych, ich rolę, budowę i zastosowania we współczesnych zadaniach, tj: w poszukiwaniu, przetwarzaniu, zarządzaniu informacją oraz w modelowaniu systemów.	K_W01, K_W02, K_W03, K_W07
	UMIEJĘTNOŚCI Student potrafi:	
U_01	korzystać z wybranych środowisk programistycznych i ich bibliotek pod kątem ich wykorzystania w projektowaniu systemów agentowych i algorytmów inspirowanych Naturą.	K_U01, K_U05, K_U06, K_U07, K_U09, K_U011
U_02	zaimplementować podstawowe algorytmy inspirowane Naturą związane z optymalizacją współczesnych problemów. Potrafi rozwiązywać problemy optymalizacyjne i transportowe z użyciem algorytmów inspirowanych procesami zachodzącymi w Naturze: zakodować instancję	K_U01, K_U05, K_U06, K_U07, K_U09, K_U011

	problemu, dobierać operatory i parametry algorytmów, dokonać analizy otrzymanych wyników.	
Forma i typy zajęć:	Studia stacjonarne: wykłady (20 godz.), ćwiczenia laboratoryjne (20 godz.) Studia niestacjonarne: wykłady (12 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
Umiejętność podstaw programowania, programowania obiektowego i sztucznej inteligencji.		
Treści modułu kształcenia:		
1. Wprowadzenie do algorytmów inspirowanych przez Naturę. Optymalizacja funkcji i optymalizacja kombinatoryczna. Problemy NP-trudne. Przegląd algorytmów inspirowanych przez Naturę. Reprezentacja i populacja początkowa. Kodowanie osobnika. Funkcja oceny.		
2. Algorytmy Ewolucyjne I: (Klasyczny) Algorytm Genetyczny (AG). Podstawowe założenia. Pojęcia genetyczne ich znaczenie. Rodzaje algorytmów genetycznych. Funkcja oceny i operatory genetyczne. Parametry i wyniki obliczeń. Podstawowe założenia. Modyfikacje klasycznego AG. Algorytm Genetyczny (Modyfikowany). Rodzaje selekcji, krzyżowania i mutacji. Dobór parametrów selekcji, krzyżowania i mutacji. Porównanie wyników różnych selekcji, krzyżowania i mutacji. Wpływ rodzaju selekcji, krzyżowania i mutacji na uzyskane wyniki działania algorytmu ewolucyjnego.		
3. Algorytmy Ewolucyjne II: Analiza algorytmu genetycznego. Rząd i długość schematu w AG. Twierdzenie o schematach. Rozwiązywanie problemów przy pomocy algorytmu genetycznego. Dobór operatorów. Dobór parametrów. Poszukiwanie rozwiązania (uruchamianie algorytmu genetycznego). Analiza i dokumentacja wyników. Stosowanie algorytmu genetycznego do rozwiązania problemu TSP (biblioteki standardowe).		
4. Algorytmy Ewolucyjne III. Strategie Ewolucyjne (SE). Założenia i operatory używane w SE. Rodzaje algorytmów SE. Zastosowania. Programowanie ewolucyjne (PE). Założenia i operatory PE. Zastosowania. Programowanie genetyczne (PG). Założenia i operatory PG. Zastosowania. Ewolucja różnicowa (DE). Założenia i operatory DE.		
5. Algorytmy Ewolucyjne IV: Równoległe algorytmy genetyczne. Modele równoległych algorytmów genetycznych - klasyfikacja. Algorytmy równoległe - wyspowe. Algorytmy równoległe - dyfuzyjne. Modele hybrydowe.		
6. Algorytmy optymalizacyjne lokalnego przeszukiwania (jednego rozwiązania). Algorytm Optymalizacji Ekstremalnej (GEO). Algorytm największego wzrostu (HillClimbing). Algorytm symulowanego wyżarzania (Simulated Annealing). Algorytm Tabu (Tabu Search).		
7. Algorytmy stadne, optymalizacyjne globalnego przeszukiwania. Rój cząsteczek (Partial Swarm). Budowa cząsteczki. Funkcja prędkości cząsteczki. Konstrukcja algorytmu. Testy i analiza wyników. Algorytm mrówkowy (Ant Colonies). Opis problemu. Kodowanie osobnika. Feromony i ich znaczenie.		
8. Zagadnienia optymalizacji drogi – Travelling Salesman Problem (TSP). Założenia. Zastosowanie algorytmu mrówkowego do rozwiązywania problemów transportowych TSP (biblioteki standardowe). Zastosowanie AG w rozwiązywaniu TSP, odpowiednie konstrukcje operatorów genetycznych.		
9. Systemy agentowe. Definicja agenta, agenta programowego. System agentowy. System wieloagentowy. Skąd wziąć inteligencję agenta? Inteligentny system agentowy (inteligencja agenta).		
10. Najnowsze osiągnięcia w zakresie zagadnień przedmiotu. Przegląd bieżącej tematyki konferencji naukowych w zakresie przedmiotu.		
Literatura podstawowa:		

1. Z. Michalewicz, Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT, 2010
2. L. Rutkowski, Metody i techniki sztucznej inteligencji, PWN, 2016
3. Z. Michalewicz, D. B. Fogel, Jak to rozwiązać czyli nowoczesna heurystyka, WNT, 2006

Literatura dodatkowa:

1. David A. Goldberg, Algorytmy genetyczne i ich zastosowania, WNT, 2003J.
2. Arabas, Wykłady z algorytmów ewolucyjnych, WNT, 2001
3. Materiały z corocznych międzynarodowych konferencji: GECCO, CEC, PPSN,...

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowiska i aplikacje programistyczne. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01- W_02	Egzamin pisemny. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania: • Podaj twierdzenie o schematach w AG. (W_01) • Czym są inteligentne systemy agentowe? (W_02)
U_01- U_02	Na bieżąco, na każdych zajęciach poza pierwszym i ostatnim w postaci zadań praktycznych. Tematyka następnego laboratorium będzie podana tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich przygotować samodzielnie lub korzystając z konsultacji. Przykładowe zadanie (sprawdza efekty: U_01, U_02): Znaleźć z dokładnością 0,001 taki argument (x), dla którego funkcja: $F(x) = \cos(20 * \pi * x) - \sin(x)$ przyjmuje maksimum w przedziale $[-4; 12]$. Utworzyć aplikację, która bazując na algorytmie roju cząsteczek rozwiąże problem postawiony w zadaniu. Zmodyfikować algorytm w taki sposób, aby poprawić jego skuteczność w rozwiązywaniu problemów z wieloma rozwiązaniami lokalnymi. Parametry programu: • Liczba cząsteczek, • Liczba kroków czasowych (iteracji): T, • Wartości wag: c1, c2, c3, • Rozmiar sąsiedztwa – gdy bgt jest najlepszą cząsteczką w sąsiedztwie a nie globalnie. Sąsiedztwo (%) oznacza ile procent najbliższych położonych cząsteczek w populacji należy do sąsiedztwa, Utworzyć animację pokapującą położenie cząsteczek w każdej iteracji, oraz najlepszego bieżącego rozwiązania.

Forma i warunki zaliczenia:

Przedmiot kończy się egzaminem. Ocenę końcową zależy od liczby uzyskanych punktów w stosunku 60% z laboratorium oraz 40% wykład (egzamin).

Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 31 pkt. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt.

Z egzaminu można uzyskać maksymalnie 40 pkt. Wykład będzie zaliczony w przypadku uzyskania z egzaminu co najmniej 21 pkt.

Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),
- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	20 godz.
Udział w ćwiczeniach laboratoryjnych	20 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	17 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Przygotowanie się do egzaminu	16 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	12 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	23 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu	22 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	50 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Praktyka zawodowa I
Nazwa w języku angielskim:	Apprenticeship I	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		drugiego stopnia
Rok studiów:	pierwszy	
Semestr:	pierwszy	
Liczba punktów ECTS:	10	
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski
Imię i nazwisko prowadzących zajęcia:		Osoba delegowana z firmy/instytucji
Założenia i cele przedmiotu:		Cele praktyki zawodowej I: Pogłębienie wiedzy i umiejętności zawodowych zdobytych podczas praktyk na studiach inżynierskich, zwłaszcza w zakresie analizy, i rozwoju istniejących systemów oraz projektowania i wytwarzania nowych aplikacji. Zastosowanie wiedzy i umiejętności w praktyce Praca w grupie, zarządzanie niewielkim zespołem
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	zagadnienia umożliwiające wytwarzanie oprogramowania oraz jego analizę i rozwijanie.	K_W02, K_W03, K_W04, K_W08
W_02	kwestie związane z rynkiem pracy oraz relację między wymaganiami pracodawców a wiedzą zdobytą w trakcie zajęć.	K_W02, K_W03
W_03	rolę systemów informatycznych w zarządzaniu firmą, potrafi administrować oraz analizować systemy informatyczne	K_W02, K_W03
W_04	zna i rozumie ogólne zasady wykorzystania technologii informatycznych w tworzeniu i rozwoju form indywidualnej przedsiębiorczości poprzez obserwację procesów wdrożeniowych i analitycznych w organizacji, w której odbywa praktyk	K_W11
	UMIEJĘTNOŚCI Student potrafi:	
U_01	zastosować wiedzę teoretyczną w praktyce oraz odpowiednio dobrać i wykorzystać narzędzia informatyczne do realizacji określonych celów w pracy indywidualnej i grupowej.	K_U01, K_U05, K_U12

U_02	zarządzać infrastrukturą informatyczną, projektować i implementować aplikacje i systemy informatyczne, analizować i testować te systemy, oraz planować dalsze kierunki ich rozwoju.	K_U11, K_U12, K_U13
U_03	pracować w grupie oraz zarządzać niewielkim zespołem	K_U04, K_U03
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	dostrzegania potrzeby zdobywania nowej wiedzy, umiejętności i doświadczeń w celu podnoszenia kompetencji zawodowych. Orientuje się w wymaganiach rynku pracy.	K_K01
K_02	dotrzymywania terminów, dba o wysoką jakość efektów pracy, przestrzega przepisów prawa oraz zasad etyki zawodowej.	K_K04, K_K05
Forma i typy zajęć:		praktyka (320 godz.)
Wymagania wstępne i dodatkowe:		
1. Wiedza i umiejętności związane z projektowaniem, programowaniem oraz analizą systemów informatycznych. 2. Doświadczenie zdobyte na praktykach w ramach studiów inżynierskich		
Treści modułu kształcenia:		
Szczegółowe treści w zależności od miejsca wykonywania praktyk zawodowych. Podczas praktyki mogą one obejmować między innymi: 1. Zasady bezpieczeństwa i higieny pracy na danym stanowisku oraz uwarunkowania prawne i etyczne stosownie do wykonywanych obowiązków. 2. Specyfika działania przedsiębiorstwa, w którym jest odbywana praktyka. 3. Rozpoznanie obszarów działalności firmy wspomaganych komputerowo. 4. Ocena istniejącej infrastruktury i wykorzystywanych technologii informatycznych w przedsiębiorstwie pod kątem zgodności ze standardami oraz możliwości rozwoju i współpracy z innymi rozwiązaniami. 5. Poznanie zastosowanych metod zarządzania infrastrukturą informatyczną oraz strategii jej rozwoju w firmie. 6. Ocena aktualnego stanu oraz przyszłych potrzeb systemów informatycznych. 7. Współudział w projektowaniu nowych i ulepszaniu istniejących systemów informatycznych, biorąc pod uwagę: Wymagania i cele stawiane przed systemem informatycznym, Polityka bezpieczeństwa oraz procedury organizacyjne dotyczące wykorzystania infrastruktury informatycznej, Praktyka projektowania zastępczego i rola pełniona przez kierownika przedsiębiorstwa w procesie projektowania, Napotykanne ograniczenia techniczne i biznesowe, Zagadnienia związane ze zwrotem kosztów inwestycji. 8. Udział w projektach informatycznych oraz, w miarę możliwości, zarządzanie niewielkim zespołem. 9. Prowadzenie dokumentacji przebiegu praktyk.		
Literatura podstawowa:		

Według zalecenia w miejscu odbywania praktyki.

Literatura dodatkowa:

Regulamin praktyk

Planowane formy/działania/metody dydaktyczne:

Cykl spotkań informacyjnych odnośnie celów i zakresu praktyki, wymaganych dokumentów i terminów oraz indywidualne konsultacje.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
W_02	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
W_03	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
W_04	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
U_01	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
U_02	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
U_03	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
K_01	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.
K_02	Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji.

Forma i warunki zaliczenia:

Podstawą zaliczenia modułu jest ocena wystawiona studentowi w instytucji przyjmującej na praktykę i może być weryfikowana przez opiekuna praktyk na podstawie rozmowy lub arkusza hospitacyjnego. Ocena ta obejmuje efekty wykonania przydzielonych zadań, jak również sposób organizacji pracy i podejmowane działania (od 0 do 50 punktów). Ponadto oceniana jest dokumentacja praktyk zarówno pod kątem merytorycznym, jak i formalnym, m.in. kompletność dokumentacji i dotrzymywanie terminów(od 0 do 50 punktów).

Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 50 pkt: niedostateczna (F),
- 51 – 60 pkt: dostateczna (E),
- 61 – 70 pkt: dostateczna plus (D),
- 71 – 80 pkt: dobra (C),

- 81 – 90 pkt: dobra plus (B),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki	320 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	12 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki	320 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	10 ECTS