

Semestr V

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Platformy Programowania
Nazwa w języku angielskim:		Programing platforms
Język wykładowy:	Polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		Obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		I stopnia
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		Dariusz Mikułowski
Imię i nazwisko prowadzących zajęcia:		Grzegorz Terlikowski
Założenia i cele przedmiotu:		Do realizacji przedmiotu niezbędne jest aby studenci posiadali podstawową wiedzę na temat działania sieci podstawowych języków, technologii i protokołów jakie są w niej wykorzystywane. Studenci powinni znać model sieci OSI, zasadę działania protokołów Internetowych takich jak FTP czy HTTP oraz podstawową znajomość języka HTML. Do zajęć niezbędne są stanowiska komputerowe z zainstalowanym na nim J2EE oraz środowiskiem programistycznym STS Net Beans lub InteliJ Idea oraz środowiskiem programistycznym .NET. Celem zajęć jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z wybranymi platformami programowania: J2EE oraz .NET a także ich wykorzystanie w praktyce
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie strukturę, najważniejsze obiekty i metody aplikacji opartej na technologii serwletów; Wie, co to są i do czego służą obiekty httprequest, httpresponse, ServletConfig i ServletContext; Wie, co to są filtry serwletów i obiekty nasłuchujące;	K_W06, K_W12
W_02	Zna i rozumie elementy, i sposób działania aplikacji JSP; Zna przeznaczenie i sposób użycia znaczników jsp; Wie, co to są wyrażenia EL i zna sposoby ich użycia w kodzie JSP; Zna strukturę i przeznaczenie komponentów Java Beans.	K_W06
W_03	Zna i rozumie trójwarstwowy model aplikacji webowej; Potrafi wymienić i opisać sposoby dostępu do danych stosowane na platformie J2EE; Wie co to	K_W06, K_W12

	są persystencje i obiekty DAO. Zna najważniejsze technologie dostępu do danych platformy J2EE takie jak: JDBC, JTA, Hibernate, JPA,	
W_04	Zna i rozumie zagadnienia związane z paradygmatem MVC, zna i rozumie funkcjonowanie poszczególnych elementów aplikacji zgodnej z tym paradygmatem; Zna i rozumie funkcjonowanie najważniejszych technologii wspierających tworzenie aplikacji MVC; Zna i rozumie strukturę i mechanizmy działania aplikacji budowanej przy pomocy frameworka Spring takie jak wstrzykiwanie zależności	K_W12
W_05	Zna i rozumie podstawowe i zaawansowane struktury języka C# takie jak: typy proste, typy złożone i generyczne, struktury, klasy, konstrukcje, delegaty; Wie, co to jest hermetyzacja klas.	K_W06
W_06	Zna i rozumie strukturę i przeznaczenie głównych elementów aplikacji tworzonych przy pomocy bibliotek Web Forms i Web MVC; Wie, jak można przeprowadzić walidację danych z formularzy przy pomocy atrybutów i własnych klas; Wie co to są obiekty ADO.Net; Zna strukturę i przeznaczenie głównych elementów aplikacji tworzonych przy pomocy bibliotek Web Forms i Web MVC; Wie, jak można przeprowadzić walidację danych z formularzy przy pomocy atrybutów i własnych klas; Wie co to są obiekty ADO.Net;	K_W09, K_W12
W_07	Zna i rozumie strukturę i sposób implementacji aplikacji WinForms; Zna główne właściwości klasy Control; Wie co to jest programowanie zdarzeniowe; Potrafi wymienić i podać przeznaczenie najważniejszych kontrolki;	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi zaimplementować prostą aplikację opartą o technologię Java Servlets posługując się w sposób efektywny środowiskiem programistycznym IntelliJ Idea lub Net Beans.	K_U11, K_U02
U_02	Potrafi zaimplementować aplikację opartą na stronach JSP.	K_U20, K_U10
U_03	Potrafi zaimplementować aplikację o strukturze zgodnej z paradygmatem MVC przy pomocy frameworka Spring MVC korzystającą z dostępu do danych utrwalanych w bazie danych.	K_U20, K_U12
U_04	Potrafi zaimplementować aplikację o strukturze zgodnej z paradygmatem MVC na platformie .NET.	K_U24, K_U13
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
K_02	Jest gotów do formułowania własnych opinii na temat różnych platform programistycznych oraz aktualnie dostępnych technologii do tworzenia aplikacji webowych.	K_K02
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> Umiejętność programowania proceduralnego na poziomie podstawowym; Umiejętność programowania zgodnego z paradygmatem programowania obiektowego; Znajomość podstawowych algorytmów i złożonych struktur danych i umiejętność ich zastosowania w czasie samodzielnego programowania. 		

4. Umiejętność programowania aplikacji działających w sieci wykorzystujących protokoły TCP i UDP.

Treści modułu kształcenia:

1. **Programowanie serwletów Java.** Struktura i pliki konfiguracyjne serwera Tomcat. Działanie serwera www i aplikacji webowej. Deskryptor wdrożeniowy aplikacji. Dynamiczne konfigurowanie serwletów przy pomocy adnotacji. Cykl życia i główne metody serwletu. Wykorzystanie technologii JDBC do łączenia się z bazą danych. Klasy nasłuchujące i filtry serwletów. Mechanizm cookies.
2. **Programowanie aplikacji opartych na technologii serwletów i dynamicznych stronach JSP.** Struktura aplikacji JSP. Rodzaje znaczników. Dyrektywy, skryptlety i standardowe znaczniki JSP. Komponenty Java Beans. biblioteka tagów JSTL. Rodzaje znaczników JSTL. Tworzenie własnych bibliotek znaczników. Wyrażenia JSP i wyrażenia w języku EL. Dokumenty JSP.
3. **Trójwarstwowy model aplikacji webowej.** Warstwy prezentacji, warstwa biznesowa, warstwa integracji danych. Paradygmat MVC. Struktury ramowe implementujące model MVC. Technologia Spring i framework Spring MVC. Mechanizmy odwrócenia kontroli i wstrzykiwania zależności. Specjalne adnotacje w Spring. Kontekst aplikacji Spring. Generowanie widoków - znaczniki Spring. Kontrolery Spring.
4. **Paradygmat MVC na przykładzie MVC Spring.** Rodzaje kontrolerów. Właściwości kontrolerów. Konfiguracja kontrolerów za pomocą adnotacji. Obsługa formularzy w Spring MVC. Walidacja danych z formularzy. Modele w Spring MVC. Kontrolery obsługujące formularze i kreatory. Przesyłanie plików na serwer.
5. **Dostęp do bazy danych w aplikacjach na platformie J2EE.** Obiekty domeny. Mechanizm JDBC. Użycie JDBC w aplikacjach Spring. Framework Hibernate. Otwarta i zamknięta sesja Hibernate. Plik odwzorowań Hibernate. Użycie obiektów DAO. Java Transaction API. Standard EJB3. Persystencje ze standardu EJB3. Deklaratywne konfigurowanie transakcji przy pomocy adnotacji.
6. **Technologie do tworzenia warstwy prezentacji aplikacji J2EE.** Technologie do generowania widoków JSF, Struts, Freemarker, Velocity, ThymeLeaf. Standard JSF. Struktura aplikacji JSF. Kontroler FacesServlet. Znaczniki JSF. Użycie wyrażań UEL. Walidacja danych z formularzy JSF. Internacjonalizacja aplikacji JSF. Technologia szablonów Velocity. Technologia tworzenia widoku ThymeLeaf, Aplikacje w technologii Struts (kafelki). Generowanie widoków przy pomocy makr Spring.
7. **Tworzenie warstwy prezentacji aplikacji za pomocą technologii Thymeleaf.**
8. **Tworzenie aplikacji webowych zgodnych z paradygmatem MVC we frameworku Spring przy pomocy technologii Spring Boot.**
9. **Podstawy platformy .NET.** Narzędzia .NET (CLR, IL, CLS, assembly). Języki platformy .NET: C#, Visual basic. Specyfikacja CLI. Rodzaje aplikacji .NET: konsolowe, webowe, aplikacje MDI i SDI.
10. **Podstawy języka C#.** i platformy .NET Elementy leksykalne języka: komentarze, operatory, wbudowane typy danych, stałe, zmienne, instrukcje. Metody klasy konsoli. Właściwości konsoli. Instrukcje warunkowe. Instrukcje pętli. Instrukcje break i continue. Operator warunkowy. Metody konwersji i rzutowanie typów. Operacje na łańcuchach znakowych. Tablice i listy. Struktury a klasy. Kolekcje. Modyfikatory dostępu. Zmienne statyczne. Akcesory i właściwości.
11. **Zaawansowane możliwości języka C#.** Typy skalarne i referencyjne. Modyfikatory ref i out. Użycie typów generycznych (kolekcje). Przeciążanie metod i konstruktorów. Klasy zamknięte i abstrakcyjne. Hermetyzacja klas. Użycie interfejsów. Indeksery. Obsługa wyjątków w C#. Instrukcje Checked i Unchecked. Obsługa plików i strumienie. Użycie delegat. programowanie oparte na zdarzeniach.
12. **Strony ASP.Net i biblioteka Web Forms.** Języki do tworzenia stron internetowych. Obiekty z biblioteki WebForms. Programowanie wizualne. Kontrolki WebForms. Obiekty Label, Edit i Button. Kontrolka Image. Zapisywanie plików na serwerze. Komponenty do obsługi poczty e-mail. Przechowywanie stanu aplikacji (obiekty Session, Application i Page. Tematy i skórki.
13. **Aplikacje .NET MVC Framework.** Struktura projektu aplikacji. Dostęp do danych (obiekty ADO.NET). Modele w MVC .net Framework. Użycie atrybutów i walidacja danych w modelach. Walidacja danych realizowana po stronie klienta. Zaawansowane mechanizmy walidacji danych. Implementowanie kontrolerów. Widoki w aplikacjach .NET MVC. Mechanizm mapowania ścieżek na metody kontrolerów.
14. **Biblioteka Windows Forms.** Właściwości klasy Control. Metody klasy Control. Klasa Application. Dodawanie i oprogramowywanie kontrolerek formularza. Programowanie zdarzeniowe. Implementacja techniki Przeciagnij i upuść. Właściwości komponentu Menustrip. Kontrolka Toolbar. Kontrolka

TabControl. Kontrolki tekstowe: CompoBox, ListBox i ReachTextBox. Kontrolka Timer. Standardowe i niestandardowe okienka dialogowe. Aplikacje SDI i MDI.

15. Biblioteka Windows Presentation Foundation – tworzenie aplikacji okienkowych dla Windows. Tworzenie interfejsu użytkownika w języku XAML. Implementowanie logiki aplikacji w klasach C#. Dodatkowe biblioteki do tworzenia interfejsu o wyglądzie kafelkowym (ModernUI). Implementowanie obsługi klawiatury w aplikacjach WPF. Umieszczanie elementów graficznych w interfejsie użytkownika.

Literatura podstawowa:

1. Rod Johnson, Juergen Hoeller, Alef Arendsen, Thomas Risberg, Colin Sampaleanu, Spring Framework. Profesjonalne tworzenie oprogramowania w Javie. Helion, Gliwice 06/2006.
2. Jon Galloway, Scott Hanselman, Phil Haack, Scott Guthrie, Rob Conery, Professional ASP.NET MVC 2, Wiley Publishing, Inc. 2010;
3. D. Mikułowski. XML w programowaniu aplikacji internetowych. Akademia Podlaska Siedlce 2009.

Literatura dodatkowa:

1. Alur Deepak, Crupi John, Malks Dan. J2EE. Wzorce projektowe. Wydanie 2 Helion Gliwice 2007.
2. Andrew Troelsen, Pro C# 2010 and the .NET 4 Platform, Apress 2009;
3. Adam Bochenek. Prosty przepis na J2EE: Boss, Eclipse i komponenty EJB. MIKOM Warszawa 2005.
4. J2EE Tutorial <http://java.sun.com/j2ee/1.4/docs/tutorial/>
5. Marty Hall, Larry Brown, Yaakov Chaikin. Java Servlet i Java Server Pages. Wydanie II. Helion Gliwice 2009.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia laboratoryjne na stanowiskach komputerowych. Zamieszczanie na stronach internetowych elementów treści wykładów oraz problemów i zadań laboratoryjnych.

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty U_01, U_02, U_03 i U_04 będą sprawdzane podczas wszystkich zajęć laboratoryjnych w formie zadań do samodzielnego rozwiązania, które studenci będą otrzymywać pod koniec każdego zajęcia. Ponadto, efekty U_01, U_02, U_03, U_04 oraz K_01 i K_02 będą sprawdzane poprzez zrealizowanie przez studenta samodzielnego projektu programistycznego, którego temat student otrzyma na zajęciach w połowie semestru. Efekty W_01, W_02, W_03, W_04, W_05, W_06 i W_07 będą sprawdzane poprzez egzamin w formie pisemnego testu elektronicznego w czasie sesji egzaminacyjnej.

Przykładowe pytanie z kolokwium (test wyboru):

Obiektami i metodami używanymi w implementacji serwletów na platformie J2EE są:

odp 1: HttpRequest i doGet

odp 2: HttpRequest, HttpResponse i doPost

odp 3: ServletFilter i initServlet

odp 4: HttpRequest i JSPRun

Przykładowe zadanie z laboratorium

Utwórz prostą aplikację składającą się z 2 widoków i 2 kontrolerów zgodną z paradygmatem MVC wykorzystując do tego framework Spring MVC.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- Regularne zajęcia – 26 pkt.,
- Obrona zadania indywidualnego – 14 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 13 pkt., obrona indywidualnego zadania – co najmniej 7 pkt. Na tej formie zajęć student może maksymalnie uzyskać 40 pkt.

Egzamin jest egzaminem pisemnym, który może mieć formę testu elektronicznego. Można na nim uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

- 0-50 pkt. ndst (F)
- 51-60 pkt. dst (E)
- 61-70 pkt. dst+ (D)
- 71-80 pkt. db (C)
- 81-90 pkt. db+ (B)
- 91-100 pkt. Bdb (A)

Poprawy:

- Jednorazowa poprawa każdego samodzielnego zadania w trakcie zajęć w semestrze.
- Dwie poprawy dowolnych samodzielnych zadań w sesji egzaminacyjnej, odpowiednio przed drugim i trzecim terminem egzaminu pisemnego.
- Poprawa obrony samodzielnego projektu w sesji egzaminacyjnej.

Bilans punktów ECTS*:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w zajęciach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do zajęć laboratoryjnych	18 godz.
Udział w konsultacjach	2 godz.
Przygotowanie się do samodzielnego zadania indywidualnego (projektu), przygotowanie się do egzaminu i obecność na egzaminie	10 godz.
Sumaryczne obciążenie studenta	75 godz.
Punkty ECTS	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
-----------	---------------------

Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach	2 godz
Przygotowanie się do samodzielnego zadania (projektu) i przygotowanie się do egzaminu i obecność na egzaminie	15 godz.
Sumaryczne obciążenie studenta	75 godz.
Punkty ECTS	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Sztuczna inteligencja	
Nazwa w języku angielskim:	Artificial Intelligence	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	Informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	obowiązkowy	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:	dr hab. inż. Jerzy Tchórzewski, prof. uczelni	
Założenia i cele przedmiotu:	Celem zajęć jest poznanie przez studentów podstawowych metod sztucznej inteligencji: systemów ekspertowych, sztucznych sieci neuronowych, algorytmów ewolucyjnych, systemów rozmytych, itp., a także nabycie przez nich praktycznych umiejętności projektowania systemów rzeczywistych z wykorzystaniem metod sztucznej inteligencji z wykorzystaniem środowiska MATLAB i Simulink z jego toolboxami (np. SIT, DLT, FLS, OPT, CST, itp.) oraz językiem programowania Matlab.	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	zna i rozumie podstawowe zagadnienia sztucznej inteligencji w zakresie tworzenia bazy wiedzy, w tym w szczególności z wykorzystaniem drzewa celów oraz zna mechanizmy wnioskowania wykorzystywane w maszynach wnioskujących, jak np. wnioskowanie do przodu, wnioskowanie do tyłu, wnioskowanie mieszane, itp. i rozumie podstawowe zagadnienia modelowania analitycznego, neuralnego i identyfikacyjnego systemów, procesów, obiektów, itp. na poziomie niezbędnym do: otrzymywania złożonych modeli w środowisku MATLABA i Simulinka	K_W11
W_02	Zna i rozumie podstawowe zagadnienia sztucznej inteligencji w zakresie projektowania systemów ekspertowych, sztucznych sieci neuronowych oraz systemów ewolucyjnych i rozumie podstawowe zagadnienia z zakresu sztucznej inteligencji, w tym	K_W11

	zna i rozumie jej podstawowe metody: systemy ekspertowe, sztuczne sieci neuronowe, algorytmy ewolucyjne, algorytmy immunologiczne, algorytmy mrówkowe, systemy rozmyte, algorytmy kwantowe, metody reprezentacji wiedzy, metody akwizycji wiedzy oraz metody wnioskowania itp.	
W_03	Zna i rozumie podstawowe zagadnienia sztucznej inteligencji w zakresie narzędzi wspomagających tworzenie systemów sztucznej inteligencji takich jak MATLAB i Simulink (w tym język Matlab, System Identification Toolbox, Neural Network Toolbox, Fuzzy Logic Toolbox, Control System Toolbox, Symbolic Math Toolbox, Optimization Toolbox, itp.) i rozumie zagadnienia z zakresu metodyki i technik programowania, najważniejszych paradygmatów programowania w językach bardzo wysokiego poziomu takich jak język systemowy Matlab	K_W11
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi formułować i rozwiązywać złożone i nietypowe problemy za pomocą metod niesymbolicznej reprezentacji wiedzy oraz wykonywać zadania w nie pełni przewidywalnych warunkach poprzez właściwy dobór literatury, baz danych i innych źródeł oraz dokonywać krytycznej ich oceny poprzez odpowiednią interpretację i przygotowanie do wykorzystania w metodach sztucznej inteligencji, a także wyciągać wnioski oraz formułować i uzasadniać opinie co do jakości danych i ich adekwatności do sformułowanego problemu badawczego.	K_U01
U_02	Potrafi projektować i przeprowadzać eksperymenty praktyczne z wykorzystaniem środowiska MATLAB i Simulink, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski.	K_U07
U_03	Potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst zawierający omówienie wyników realizacji tego zadania, w tym dokumentację z realizowanych laboratoriów i instrukcję użytkownika opracowanego własnego systemu, potrafi przygotować drzewo celów oraz zaprogramować system ekspertowy, potrafi przygotować plik uczący i zaprojektować sztuczną sieć neuronową, potrafi przygotować populację początkową i zaprogramować algorytm genetyczny.	K_U08
U_04	Przy identyfikowaniu i formułowaniu specyfikacji zadań inżynierskich oraz przy ich rozwiązywaniu potrafi wykorzystywać metody analityczne, symulacyjne i eksperymentalne, dostrzegać ich aspekty systemowe i pozatechniczne oraz dokonać wstępnej oceny ekonomicznej proponowanych rozwiązań i podejmowanych działań inżynierskich oraz potrafi wykorzystać środowisko MATLABA i Simulinka (j. Matlab, SMT, SIT, NNT, CST, FLT) do projektowania, testowania i symulacji złożonych systemów sztucznej inteligencji.	K_U09
U_05	Potrafi zaprojektować proces testowania implementowanego systemu informatycznego, przeprowadzić diagnozę błędów, wyciągnąć wnioski i wprowadzić niezbędne poprawki w systemie informatycznym.	K_U12

U_06	Potrafi zaprojektować, zaimplementować oraz przygotować proces testowania i go przeprowadzić - zgodnie z zadaną specyfikacją - systemu informatycznego typowego dla kierunku informatyka, a otrzymane wyniki przedstawić w formie liczbowej lub graficznej, a następnie właściwie je zinterpretować i wyciągnąć wnioski.	K_U19
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do podejmowania decyzji, krytycznej oceny działań własnych i grupy laboratoryjnej, w której uczestniczy. Jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki oraz przyjmowania odpowiedzialności za skutki pracy na zajęciach laboratoryjnych.	K_K01
K_02	jest gotów do wypełniania zobowiązań społecznych, współorganizowania działalności na rzecz środowiska społecznego oraz inicjowania działań na rzecz interesu publicznego poprzez przekazywanie informacji i opinii dotyczących osiągnięć informatyki i innych aspektów działalności inżyniera-informatyka; podejmuje starania, aby przekazać takie informacje i opinie w sposób zrozumiały.	K_K02
K_03	Jest gotów do odpowiedzialnego pełnienia roli zawodowej informatyka, w tym do przestrzegania zasad etyki zawodowej i wymagania tego od innych i do dbania o dorobek i tradycję zawodu informatyka.	K_K04
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> 1. Wiedza z podstaw logiki matematycznej, rachunku zdań, rachunku predykatów, algorytmów i złożoności. 2. Znajomość podstaw programowania deklaratywnego, w tym języka Prolog. 3. Znajomość podstaw grafiki i komunikacji człowiek-komputer. 4. Umiejętność samodzielnego programowania w dowolnych środowiskach programistycznych. 5. Znajomość wykorzystywania inżynierii oprogramowania. 		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Wstęp do sztucznej inteligencji. Inteligencja a sztuczna inteligencja, algorytmika a heurystyka, Logika a metalogika, wiedza a baza wiedzy, życie a sztuczne życie, Kod genetyczny a kod informacyjny, Test Turinga, Automatyka, cybernetyka, informatyka a sztuczna inteligencja, Metody, języki i narzędzia sztucznej inteligencji, Projektowanie i implementacja systemów sztucznej inteligencji, Zastosowania systemów sztucznej inteligencji 2. Inżynieria wiedzy i systemy ekspertowe. Istota systemów ekspertowych. Struktura Bazy Wiedzy Systemu Ekspertowego. Projektowanie i programowanie systemów ekspertowych. Metody akwizycji wiedzy. Moduły objaśniania. Języki systemów ekspertowych, Inżynieria wiedzy i architektura systemów ekspertowych. Zastosowania systemów ekspertowych. Metody reprezentacji wiedzy. Systemy ekspertowe czasu rzeczywistego, itp. 3. Metody wnioskowania. Wnioskowanie w przód, Sterowanie wnioskowaniem, Wnioskowanie wstecz, Wnioskowanie mieszane, Wnioskowanie rozmyte, Podstawowe zagadnienia uczenia się maszyn, Metodyka tworzenia i badania poprawności bazy wiedzy. Maszyna Wnioskująca a Baza Wiedzy. 4. Sztuczna Sieci Neuronowe I. Idea sieci neuronowych, Charakterystyka sztucznego neuronu, Proste modele neuronu, Rodzaje sieci neuronowych, Struktura sieci, Funkcjonowanie sztucznej sieci 		

neuronowej, Sieci liniowe i nieliniowe, Sieci jednokierunkowe i sieci rekurencyjne, Sieci komórkowe, Sieci jednowarstwowe i wielowarstwowe, Przykłady sztucznych sieci neuronowych: PERCEPTRON, HOPFIELDA, HAMMINGA, Kohonena, sieci chaotyczne, sieci ontogeniczne, sieci dualne, sieci konwolucyjne, sieci probabilistyczne, itp.

5. **Sztuczne Sieci Neuronowe II.** Reguły uczenia sztucznych sieci neuronowych, Uczenie z nadzorem, bez nadzoru, z krytykiem, Reguły uczenia: Reguła HEBBA, Reguła Perceptronowi, Reguła Delta, Reguła Widrowa – Hoffa, Reguła WTA i WTM. Reguła gwiazdy wyjść. Nowoczesne metody uczenia. Parametry uczenia Sztucznej Sieci Neuronowej
6. **Algorytmy ewolucyjne I.** Klasyczny algorytm genetyczny, Pojęcie krzepkości algorytmów genetycznych, Populacja początkowa i populacja rodzicielska, Operatory genetyczne krzyżowania i mutacji, Metody selekcji, Rola funkcji przystosowania, Zarządzanie populacją.
7. **Algorytmy genetyczne II.** Matematyczne podstawy algorytmów genetycznych, Teoria schematów, Liczność i długość schematu, Przystosowanie schematu, Twierdzenie o schematach, Hipoteza bloków budujących, Zapobieganie przedwczesnej zbieżności. Rodzaje algorytmów ewolucyjnych. Strategie ewolucyjne, Programowanie genetyczne i ewolucyjne, Algorytmy koewolucyjne, Genetyczne systemy uczące się, Modyfikacje algorytmów ewolucyjnych.
8. **Sztuczne systemy immunologiczne.** Detekcja jako dopasowywanie elementów, Pamięć immunologiczna, Funkcjonowanie i metadynamika układu odpornościowego, Typy struktur w systemie, Binarny klasyfikator, Rola strzelców, Samoorganizacja pamięci immunologicznej, Pociski (limfocyty wysyłane przez strzelców), System immunologiczny jako system ewolucyjny.
9. **Metody analizy skupień.** Metody i techniki eksploracji danych. Techniki odkrywania asocjacji. Metody klasyfikacji. Metody grupowania. Metody hierarchiczne i metody niehierarchiczne, itp.
10. **Systemy rozmyte.** Zbiory rozmyte, Logika rozmyta, Rola funkcji przynależności w systemach rozmytych, fuzyfikator, defuzyfikator, Rozmyte systemy ekspertowe, Sztuczne sieci neuronowe o logice rozmytej, itp.
11. **Nowe metody sztucznej inteligencji.** Sztuczne systemy mrówkowe, Fabryki bezludne, Sztuczne życie, Uczenie maszynowe, Algorytmy kwantowe, Rozproszone systemy sztucznej inteligencji, Uczenie głębokie, itp.

Literatura podstawowa:

1. Faliński M.: Wstęp do sztucznej inteligencji. PWN. Warszawa 2018.
2. Rutkowski L.: Metody i techniki sztucznej inteligencji. PWN. Wyd. II zmienione. Warszawa 2020.
3. Tchórzewski J.: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów. Wydawnictwo UPH. Siedlce 2020.
4. Tchórzewski J.: Sztuczna inteligencja i informatyka kwantowa. Cz. 1. Wybrane zagadnienia. Cz. 2. Ćwiczenia laboratoryjne w środowisku MATLAB'a i Simulink'a (Cz. 1 dostępna w postaci print screen-ów prezentacji w Power Point. Ostatnia aktualizacja: wrzesień 2020 r. Cz. 2 dostępna w postaci instrukcji ćwiczeń laboratoryjnych w wersji z września 2020 r.).

Literatura dodatkowa:

1. Arabas J.: Wykłady z algorytmów ewolucyjnych. Wyd. 2. WNT. Warszawa 2016.
2. McIlwraith D., Marmanis H., Babenko D.: Inteligentna sieć. Algorytmy przyszłości. Helion. Warszawa 2017.
3. Osowski S.: Sieci neuronowe do przetwarzania informacji. Wyd. III. OW PW. Warszawa 2013.
4. Tadeusiewicz R., Chaki R., Chaki N.: Exploring Neural Networks with C#. CRC Press. 2014.
5. Wawrzyński P.: Podstawy sztucznej inteligencji. WPO. Warszawa 2015.
6. Wierzchoń S.: Sztuczne systemy immunologiczne. Teoria i zastosowania. AOW EXIT. Warszawa 2001.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Udostępnianie studentom treści wykładów przygotowanych w postaci print screen-ów prezentacji w MS Power Point oraz instrukcji do ćwiczeń laboratoryjnych (wersja pdf). Ćwiczenia laboratoryjne realizowane są w 4-rech blokach tematycznych. Na każdy temat składają się trzy ćwiczenia laboratoryjne:

- przygotowanie przez studenta własnego zadania do zaprogramowania w środowisku MATLAB i Simulink (przygotowanie danych rzeczywistych na zadany temat, sposobu rozwiązania zadania oraz poznanie środowiska programowania),
- zaprojektowanie własnego zadania przy konsultacji z prowadzącym zajęcia w środowisku MATLABA i Simulinka,
- opracowanie sprawozdania z wykonanego samodzielnie zadania, instrukcji obsługi programu oraz zaliczenie tematu (praktyczne i teoretyczne)

Treści tematów:

Blok 1 – SIT, CST (Lab. 1-3): Opracowanie eksperymentu badawczego, danych do badań oraz przeprowadzenie identyfikacji i badań analitycznych systemu w środowisku MATLABA i Simulinka z wykorzystaniem System Identification Toolbox-a (SIT) i Control System Toolbox-a (CST).

Blok 2 – DLT, FLT (Lab 4-6): Przygotowanie pliku uczącego oraz zaprojektowanie sztucznej sieci neuronowej w środowisku MATLAB i Simulink z wykorzystaniem Neural Network Toolbox-a i Fuzzy Logic Toolbox-a.

Blok 3 – OT, GOT, j. Matlab (Lab 7-9): Zaprojektowanie algorytmu ewolucyjnego (AE) na bazie uzyskanych wyników identyfikacji (utworzenie populacji początkowej, opracowanie własnego algorytmu krzyżowania, mutacji, selekcji, funkcji krzepkości, itp.) oraz zaprogramowanie algorytmu genetycznego w j. Matlab w postaci m-pliku oraz jego przetestowanie lub zaprojektowanie algorytmu ewolucyjnego z wykorzystaniem Optimization Toolbox-a oraz Global Optimization Toolbox-a.

Blok 4 – SIM, j. Matlab (Lab 10-12): Zaprojektowanie w Simulinku modelu do symulacji i testowania poprawności uzyskanych rozwiązań, a m.in. do porównywania uzyskanych modeli za pomocą różnych technik programistycznych oraz w odniesieniu do systemu rzeczywistego.

Zadania indywidualne: Każdy zainteresowany student otrzymuje do samodzielnego poznania środowisko programistyczne z zakresu sztucznej inteligencji. Samodzielnie opracowuje przykład, implementuje go w środowisku oraz przygotowuje sprawozdanie z przeprowadzonych badań oraz opracowuje instrukcję obsługi programu. Możliwe do wyboru toolbox-y środowiska MATLAB m.in.: Image Processing Toolbox, Mapping Toolbox, Symbolic Math Toolbox, Signal Processing Toolbox, Machine and Statistics Learning Toolbox, Wavelet Toolbox, itp.

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty **K_W01, K_W05**, sprawdzane są na dwóch kolokwium pisemnych, kolokwium sprawdzającym wiedzę praktyczną oraz kolokwium sprawdzającym wiedzę teoretyczną.

Kolokwium praktyczne dotyczyć będzie praktycznego użycia wybranych metod sztucznej inteligencji oraz związanych z nimi zadań szczegółowych.

Przykładowe zadania:

- Podany jest problem diagnostyczny uszkodzenia samochodu. Należy opracować pięć wyróżników i odpowiadających im pięć poziomów pytań oraz od 15 do 20 konkluzji, a następnie należy opracować drzewo celów i zaprojektować bazę wiedzy z wykorzystaniem j. Matlab.
- Dana jest struktura pliku uczącego. Należy zaprojektować sztuczną sieć neuronową w środowisku MATLAB z wykorzystaniem Neural Network Toolbox-a do rozpoznawania cyfr rzymskich.
- Dany jest model parametryczny postaci $A1(q) y_1(t) = B1(q) u_1(t) + B2(q) u_2(t)$, gdzie: $A1(q) = 1 + 0,2 q^{-1} + 0,3 q^{-2} + 0,4 q^{-3}$, $B1(q) = 0,2 q^{-1} + 0,4 q^{-2}$, $B2(q) = 0,1 q^{-1} + 0,3 q^{-2}$. Należy utworzyć populację początkową dla potrzeb algorytmu ewolucyjnego i napisać program w j. Matlab w postaci m-pliku realizującego algorytm genetyczny zawierający jednopunktowe krzyżowanie, mutację poprzez zmianę znaku genu oraz selekcję turniejową.

- Dany jest plik uczący, struktura sztucznej sieci neuronowej oraz reguła uczenia. Należy przeprowadzić obliczenia uczenia sztucznej sieci neuronowej i przedstawić model sztucznej sieci neuronowej złożony z sumatorów i układów odwzorowujących.
- Dany jest model matematyczny systemu. Należy zbudować schemat blokowy w Simulinku. Zaproponować sposób na przeprowadzenie badań symulacyjnych i badania wrażliwości.

Kolokwium teoretyczne dotyczy będzie sprawdzenia znajomości wiedzy teoretycznej i odbędzie się w postaci testu z jednokrotnym wyborem.

Efekty **K_U04**, **K_U05**, **K_U06** sprawdzane są cztery razy, to jest przy zaliczaniu każdego tematu ćwiczeń laboratoryjnych.

Efekty **K_K01**, **K_K02**, **K_K04** będą weryfikowane, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, podczas zaliczania tematów laboratoriów oraz zadania indywidualnego.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie ocen z czterech tematów wykonanych samodzielnie przez studentów na zajęciach laboratoryjnych: pozyskanie danych i wykonanie samodzielne ćwiczenia laboratoryjnego (uzyskanie własnych wyników), obrony wykonanego laboratorium z wykazaniem się wiedzą teoretyczną i praktyczną (obrona wykonanego laboratorium) oraz wykonanie sprawozdania z bloku tematycznego (przedstawienie pisemne uzyskanego rozwiązania i jego interpretacji) oraz z testu pisemnego sprawdzającego wiedzę teoretyczną i praktyczną nabytą na wykładach, laboratorium i podczas studiów indywidualnych (zadania problemowe). Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach laboratoryjnych z nauczycielem akademickim według schematu:

- Regularne zajęcia laboratoryjne (4 bloki tematyczne x 25 pkt) – 100 pkt, przy czym każdy temat oceniany jest w zakresie: wiedzy praktycznej (wykonanie ćwiczenia laboratoryjnego i uzyskanie własnych wyników) – 15 pkt., obrona wykonanego laboratorium – 5p. oraz wykonanie sprawozdania z wykonanego laboratorium – 5pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów ze wszystkich form aktywności studenta, tj. co najmniej 51 pkt. Na tej formie zajęć student może maksymalnie uzyskać 100 pkt.

Za jeden pisemny test można uzyskać maksymalnie 100 pkt. Zaliczenie testu jest możliwe po uzyskaniu co najmniej 51 pkt. Każdy student może uzyskać dodatkowe punkty za zadanie indywidualne oraz za aktywność na zajęciach laboratoryjnych i na wykładach (w częściach podsumowujących laboratoria i wykłady). Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), jest średnią z oceny uzyskanej na laboratorium oraz na wykładach, w zależności od sumy uzyskanych punktów (może przekroczyć 100 pkt.) jest następująca odrębnie w przypadku laboratorium, czy też w przypadku wykładów (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).

W przypadku przekroczenia przez studenta wymaganej liczby maksymalnej 100 pkt. student uzyskuje od prowadzącego wpis oceny bdb z dodatkowym wyróżnieniem, które przez prowadzącego jest interpretowane jako ocena celująca (prowadzący przekazuje studentowi taką ocenę w indywidualnej formie z przedmiotu Sztuczna inteligencja).

Poprawy: Istnieje możliwość jednorazowej poprawy testu w toku trwania semestru oraz dwukrotnej poprawy testu w sesji egzaminacyjnej, odpowiednio przed upływem terminów przewidzianych wpisami oceny do systemu USOS.

Studia stacjonarne

Bilans punktów ECTS:

Aktywność	Obciążenie studenta
Udział w wykładach i studia zagadnień z wykładów	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	13 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Przygotowanie się do dwóch kolokwiów pisemnych (sprawdzianu z umiejętności praktycznych (zadania problemowe) oraz sprawdzianu z wiedzy teoretycznej (testu) i udział w obu kolokwiach (2 x 15 godz.)	15 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Bilans punktów ECTS:

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach z przedmiotu i studia zagadnień z wykładów	2 godz.
Przygotowanie się do dwóch kolokwiów pisemnych (z części teoretycznej – test,	15 godz.

z części praktycznej – zadania problemowe) i udział w obu kolokwiach, przy czym kolokwium z zadań praktycznych odbędzie się w formie elektronicznej zdalnej (2 x 15 godz.)	
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Inżynieria oprogramowania	
Nazwa w języku angielskim:	Software engineering	
Język wykładowy:	Polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	obowiązkowy	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	Trzeci	
Semestr:	Piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:	dr Jarosław Skaruz	
Imię i nazwisko prowadzących zajęcia:	dr Jarosław Skaruz dr Marek Piłski	
Założenia i cele przedmiotu:	Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z wytwarzaniem systemów informatycznych	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia z zakresu inżynierii oprogramowania, w tym zna fazy rozwoju oprogramowania oraz metody podwyższania jakości oprogramowania.	K_W06
W_02	Zna i rozumie zagadnienia z zakresu projektowania i eksploatacji systemów informatycznych.	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów	K_U05

U_02	Potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst zawierający omówienie wyników realizacji tego zadania	K_U08
U_03	Potrafi posłużyć się właściwie dobranymi środowiskami programistycznymi oraz narzędziami komputerowo wspomaganego projektowania do projektowania i weryfikacji systemów informatycznych	K_U11
U_04	Potrafi, zgodnie ze specyfikacją, zaplanować proces realizacji systemu informatycznego; potrafi wstępnie oszacować jego koszty	K_U19
U_05	Potrafi zaprojektować, zaimplementować oraz przetestować prosty system informatyczny	K_U16
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
1 Umiejętność programowania		
Treści modułu kształcenia:		
<p>Wprowadzenie do inżynierii oprogramowania. Kryzys oprogramowania. Złożoność projektu oprogramowania. Modele cyklu życiowego oprogramowania.</p> <p>Modelowanie biznesowe. Cele i efekty modelowania biznesowego. Rodzaje modelowania biznesowego. Notacja.</p> <p>Modelowanie biznesowe II. Notacja BPMN.</p> <p>Metodyka Rational Unified Process. Główne koncepcje metodyki RUP. Zalecane dobre praktyki. Fazy rozwoju oprogramowania. Aktywności projektowe.</p> <p>Wymagania na system. Cele wymagań na system. Wymagania funkcjonalne i нефункционалне. Pozyskiwanie i specyfikowanie wymagań.</p> <p>Projektowanie użytecznych interfejsów użytkownika. Cele usability. Wpływ optymalizacji stron WWW. Optymalizacja elementów WWW: nawigacja, wyszukiwarki, teksty, linki i komunikaty.</p> <p>Analiza i projektowanie I. Cele analizy i projektowania. Forward i backward engineering. Model projektu. Klasy i interfejsy.</p>		

Analiza i projektowanie II. Notacja diagramów klas. Ujęcie diagramów w różnych perspektywach. Prezentacja atrybutów, metod i związków pomiędzy klasami.

Analiza i projektowanie III. Diagramy czynności, sekwencji i pakietów.

Implementacja. Metody efektywnego kodowania: tworzenie i usuwanie obiektów, konstruktory prywatne, unikanie powielania obiektów, wyciek pamięci, interfejsy i klasy abstrakcyjne.

Wzorce projektowe I. Wprowadzenie do wzorców projektowych. Elementy składowe wzorców. Podział wzorców. Wzorzec Startegia

Wzorce projektowe II. Wzorzec Singleton i Dekorator.

Testowanie. Weryfikacja i walidacja. Rodzaje testów. Metody czarnej i białej skrzynki. Inspekcja a testowanie. Framework JUnit.

Zarządzanie projektem. Metodyki zarządzania projektami. Cykl życia projektu. Zarządzanie ryzykiem.

Narzędzia inżynierii oprogramowania. Środowiska programowania., Systemy kontroli wersji. Systemy ciągłej integracji. Systemy raportowania błędów

Literatura podstawowa:

1. FC. Larman, UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji, Helion, 2011
2. E. Freeman, E. Freeman, B. Bates, K. Sierra, Rusz głową ! Wzorce projektowe, Helion, 2011
3. E. Hasted, Sprzedaj swój software, Helion, 2007

Literatura dodatkowa:

1. B. K. Jayaswal, P. C. Patton, Oprogramowanie godne zaufania. Metodologia, techniki i narzędzia projektowania, Helion, 2007
2. D. Pilone, R. Miles, Head First Software Development, Helion, 2008

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia z wykorzystaniem narzędzi CASE. Zamieszczanie na stronach internetowych materiałów dydaktycznych

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W_01 i W_02 sprawdzane będą na egzaminie w sesji egzaminacyjnej.

Efekty U_02-U_04 będą sprawdzane po piątym, dziewiątym i czternastym ćwiczeniu. Studenci realizują zadanie indywidualne polegające na utworzeniu oprogramowania zgodnie z własnoręcznie przygotowaną specyfikacją techniczną i projektem.

Efekty U_01-U_05 będą sprawdzane na każdym zajęciach laboratoryjnych. Studenci pracując w grupach realizują zadania polegające na utworzeniu modeli z wykorzystaniem języka UML np. diagramów przypadków użycia, klas, pakietów, sekwencji, aktywności, wdrożenia, komponentów.

Efekt K1 jest sprawdzany podczas zajęć laboratoryjnych. Studenci zobowiązani są do pracy w grupie, rozwiązując wspólnie zadanie dotyczące zaprojektowania małej części systemu informatycznego z wykorzystaniem języka UML.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem ustnym. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- Regularne zajęcia – 26 pkt.,
- Obrona zadania indywidualnego – 14 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 13 pkt., obrona indywidualnego zadania – co najmniej 7 pkt. Na tej formie zajęć student może maksymalnie uzyskać 40 pkt.

Egzamin jest egzaminem ustnym. Można na nim uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) 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	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	8 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	5 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.

Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	18 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia	
Nazwa przedmiotu/modułu kształcenia:	Podstawy przetwarzania rozproszonego
Nazwa w języku angielskim:	Fundamentals of distributed computing
Język wykładowy:	polski
Kierunek studiów, dla którego przedmiot jest oferowany:	informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia
Rok studiów:	trzeci
Semestr:	piąty
Liczba punktów ECTS:	3
Imię i nazwisko koordynatora przedmiotu:	dr Marcin Stępiak
Imię i nazwisko prowadzących zajęcia:	dr Marcin Stępiak
Założenia i cele przedmiotu:	<p>Studenci przystępujący do tego przedmiotu powinni znać zasady programowania obiektowego, a także treści z przedmiotu „algorytmy i złożoność”.</p> <p>Celem przedmiotu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z przetwarzaniem rozproszonym oraz modelami wykorzystywanymi w projektowaniu i implementacji systemów rozproszonych. Studenci poznają podstawy teorii grafów, modele obliczeniowe i podstawowe algorytmy wykorzystywane w przetwarzaniu rozproszonym. W ramach tego przedmiotu studenci poznają także mechanizmy synchronizacji zegarów oraz algorytmy budowy drzewa rozpinającego, numerowania węzłów i routowania. Dodatkowo poznają algorytmy wyboru przywódcy, mechanizmy wzajemnego wykrywania, a także sposoby wykrywania zakleszczenia, zakończenia i awarii. Na koniec studenci zostają zapoznani z podstawami technologii blockchain.</p>

Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna i rozumie podstawowe standardy, pojęcia i modele przetwarzania rozproszonego.	K_W06, K_W07, K_W10
W_02	Zna i rozumie modele systemów rozproszonych i techniki ich programowania.	K_W06, K_W07
W_03	Zna i rozumie zagadnienia z teorii grafów wykorzystywane w przetwarzaniu rozproszonym. Definicje podstawowych pojęć. Wybrane twierdzenia. Struktury grafów mające zastosowanie w programowaniu rozproszonym.	K_W01
W_04	Zna i rozumie zagadnienia komunikacji zachowującej uporządkowanie wiadomości, konstrukcji spójnego obrazu stanu globalnego, detekcji zakleszczenia rozproszonego oraz detekcji zakończenia przetwarzania rozproszonego.	K_W07
W_05	Zna i rozumie algorytmy przetwarzania w zawodnym środowisku rozproszonym, w tym zagadnienia: modelowania i klasyfikacji awarii, konstrukcji niezawodnych kanałów komunikacyjnych, realizacji detektorów awarii, niezawodnej komunikacji grupowej oraz konsensusu rozproszonego i jego zastosowań.	K_W07
W_06	Zna i rozumie technologię blockchain i jej zastosowania.	K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi zbudować aplikację działającą w środowisku rozproszonym.	K_U01, K_U10, K_U11
U_02	Umie wybrać i zaimplementować odpowiednie do problemu i zgodne ze standardami algorytmy rozproszone.	K_U01, K_U10, K_U11, K_U19
U_03	Potrafi programować algorytmy i systemy rozproszone w środowisku Java.	K_U10, K_U17, K_U22
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań przetwarzania rozproszonego	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.); studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)

Wymagania wstępne i dodatkowe:

Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów:

1. programowanie obiektowe
2. algorytmy i złożoność

lub znajomość literatury obowiązującej w tych przedmiotach.

Treści modułu kształcenia:

1. **Wprowadzenie do przetwarzania rozproszonego.** Podstawowe pojęcia i definicje. Charakterystyka środowiska przetwarzania rozproszonego. Przykłady środowisk rozproszonych.
2. **Podstawy teorii grafów w przetwarzaniu rozproszonym.** Definicje podstawowych pojęć. Wybrane twierdzenia. Struktury grafów mające zastosowanie w programowaniu rozproszonym.
3. **Modele obliczeniowe programów rozproszonych.** Pojęcie modelu obliczeniowego. Model procesów współbieżnych. Cięcia, stan globalny, stan globalny spójny. Algorytm Chandy-Lamporta. Algorytm Lai Yanga. Podnoszenie z awarii. Model obiektów rozproszonych.
4. **Synchronizacja zegarów.** Czas fizyczny a logiczny. Algorytmy czasu logicznego.
5. **Elementarne algorytmy rozproszone.** Podstawowe własności algorytmów. Algorytm fali, przemierzania. Algorytm rozgłaszania.
6. **Budowa drzewa rozpinającego i numerowanie węzłów.** Pojęcie drzewa rozpinającego. Algorytm echo. Algorytm Tarriego. Problem minimalnego drzewa rozpinającego. Zagadnienie enumeracji węzłów. Metoda losowania. Metoda z przywódcą. Algorytm w topologii drzewa.
7. **Rutowanie i wybór przywódcy.** Zadania i kryteria rutowania. Protokoły rutowania. Algorytm Floyda-Warshalla. Algorytm Touega. Algorytm Merlina-Segalla. Algorytm przepychanki. Algorytm potopowy. Algorytm na pierścieniu. Algorytm na drzewie. Algorytm w sieciach anonimowych.
8. **Rozproszone wykrywanie zakończenia i awarii.** Wykrywanie zakończenia. Algorytm Dijkstra'y-Scholten'a. Algorytm na podwójnym pierścieniu. Algorytm energetyczny. Wykrywanie awarii. Metoda ping-ack. Metoda heart-beat.
9. **Wzajemne wykluczanie.** Algorytm centralnego serwera. Algorytm Lamporta. Algorytm głosujący Maekawa. Algorytm Ricarta-Agrawali. Algorytm pierścienia z żetonem. Algorytm Raymonda.
10. **Wykrywane zakleszczenia.** Pojęcie i strategie postępowania z zakleszczeniami. Podejście scentralizowane i rozproszone. Algorytm Chandy-Misra-Haas.
11. **Technologia blockchain.** Omówienie podstawowych zastosowań. Kryptowaluty na przykładzie Bitcoin.

Literatura podstawowa:

1. M. Ben-Ari: Podstawy programowania współbieżnego i rozproszonego, WNT, 2009
2. Karbowski (red.), E. Niewiadomska-Szynkiewicz (red.), Programowanie równoległe i rozproszone, Politechnika Warszawska, 2009
3. Zbigniew Weiss, Tadeusz Gruźlewski, Programowanie współbieżne i rozproszone w przykładach i zadaniach, Wydawnictwa Naukowo-Techniczne, 1994

Literatura dodatkowa:

1. Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing Principles, Algorithms, and Systems, Cambridge University Press, 2008
2. Tel G. Introduction to Distributed Algorithms, Cambridge University Press, 2000
3. A.S. Tanenbaum, M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2001
4. Lynch N. A. Distributed algorithms, Morgan Kaufmann Publishers, 1996
5. Guerraoui R., Rodrigues L. Introduction to Reliable Distributed Programming, Springer-Verlag, 2006
6. Drescher D., Blockchain. Podstawy technologii łańcucha bloków w 25 krokach, Helion, 2018

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko programowania Java. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W_01 – W_06 będą sprawdzane na egzaminie pisemnym.

Przykładowe zadania:

- Omów metody synchronizacji zegarów fizycznych w środowiskach rozproszonych.
- W jaki sposób przeprowadzić detekcję zakończenia programu rozproszonego w sieci w topologii pierścienia?
- W jaki sposób można wykrywać zakleszczenia w systemach rozproszonych?
- Jaki jest cel algorytmów rozgłaszania? Podaj przykłady takich algorytmów.
- Znajdź minimalne drzewo rozpinające grafu.

Efekty U_01 – U_03 sprawdzane będą na bieżąco, na każdym zajęciach (poza pierwszymi) poprzez implementacje w środowisku Java algorytmów rozwiązujących zadania praktyczne.

Przykładowe zadanie:

- Napisz system rozproszony w języku Java, który realizuje model producenci-konsumenci, zapewniając poniższe założenia:
 - jeden producent,
 - dwóch konsumentów,
 - bufor ograniczony.

Efekt K_01 będzie weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie wykonywania zajęć laboratoryjnych w grupach. Studenci będą mieli za zadanie wybrać odpowiedni mechanizm do rozwiązania problemu i ocenić jego skuteczność, a także porównać wyniki z innymi studentami z grupy.

Tematyka zajęć laboratoryjnych zostanie podana około tygodnia przed konkretnymi zajęciami.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium.

Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane w trakcie zajęć laboratoryjnych, za które w sumie można uzyskać maksymalnie 110 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu, co najmniej 56 pkt.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 90 pkt. Egzamin będzie zaliczony w przypadku uzyskania, co najmniej 46 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 200pkt. po zaokrągleniu do liczby całkowitej) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 101 pkt: niedostateczna (F),
- 102 – 120 pkt: dostateczna (E),
- 121 – 140 pkt: dostateczna plus (D),
- 141 – 160 pkt: dobra (C),
- 161 – 180 pkt: dobra plus (B),
- 181 – 200 pkt: bardzo dobra (A).

Poprawy:

Uzyskanie poprawkowego zaliczenia laboratoriów możliwe jest w sesji egzaminacyjnej, przed terminem egzaminu pisemnego.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	8 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
-----------	---------------------

Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	28 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Komunikacja i przetwarzanie w chmurze
Nazwa w języku angielskim:		Cloud communications and computing
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		Waldemar Bartyna
Imię i nazwisko prowadzących zajęcia:		Waldemar Bartyna
Założenia i cele przedmiotu:		<p>Studenci przystępujący do tego przedmiotu powinni znać programowanie obiektowe oraz potrafić efektywnie korzystać ze środowiska programistycznego Visual Studio.</p> <p>Celem przedmiotu jest zapoznanie studentów z technologiami związanymi z komunikacją i przetwarzaniem w chmurach; z projektowaniem i implementowaniem aplikacji w chmurze, ich debugowaniem, monitorowaniem i skalowaniem, z przechowywaniem różnego typu informacji w chmurach. Studenci będą również potrafili wykorzystywać poznane technologie i narzędzia do tworzenia aplikacji w chmurze i używanie chmury w celu składowania i zarządzanie różnego rodzaju danymi.</p>
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zasady dotyczące komunikacji i przetwarzania w chmurze, ich rodzaje i zastosowania.	K_W06

W_02	Zna i rozumie sposoby przechowywania różnego rodzaju danych w chmurze.	K_W06
W_03	Zna i rozumie funkcjonowanie narzędzi i metodyk związanych z projektowaniem i programowaniem aplikacji w chmurze i migrowania istniejących aplikacji do chmury.	K_W06
W_04	Zna i rozumie sposoby komunikacji i skalowalności w chmurze.	K_W06
W_05	Zna i rozumie kwestie bezpieczeństwa związane z programowaniem w chmurze.	K_W06
W_06	Zna i rozumie sposoby tworzenia, debugowania, monitorowania i zarządzania aplikacjami w chmurze.	K_W07, K_W12
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi programistycznie wykorzystać różne sposoby korzystania z chmury do przechowywania danych.	K_U02
U_02	Potrafi dostosować aplikacje do możliwości komunikacyjnych i skalowania w chmurze.	K_U02, K_U20
U_03	Potrafi zarządzać, debugować i monitorować aplikacjami w chmurze.	K_U11, K_U12
U_04	Potrafi migrować aplikacje do chmury.	K_U21
U_05	Potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązywania prostych zadań inżynierskich, typowych dla projektowania i programowania aplikacji w chmurze oraz wybierać i stosować właściwe metody i narzędzia związane z zaawansowanymi technikami programistycznymi.	K_U10
U_06	Potrafi zaprojektować, zaimplementować oraz przetestować proste aplikacje w chmurze.	K_U01, K_U02
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy związanej z programowaniem w chmurze.	K_K01
K_02	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów programistycznych oraz do konstruktywnej krytyki powstałych rozwiązań.	K_K01

Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:	
<ol style="list-style-type: none"> 8. Dobra znajomość programowania w językach obiektowych 9. Znajomość środowiska programistycznego Visual Studio 	
Treści modułu kształcenia:	
<ol style="list-style-type: none"> 1. Wprowadzenie do komunikacji i przetwarzania w chmurze 2. Wprowadzenie do Microsoft Azure 3. Azure Storage – Tabele 4. Azure Storage – Blogy 5. Komunikacja i skalowanie w chmurze 6. Aplikacja w chmurze 7. Dane w chmurze 8. Zarządzanie, debugowanie i monitorowanie aplikacji w Microsoft Azure 9. Wprowadzenie do SQL Azure. 10. Migracja aplikacji do chmury 11. Bezpieczeństwo danych i aplikacji w chmurze 	
Literatura podstawowa:	
<ol style="list-style-type: none"> 1. Zbigniew Fryźlewicz, Łukasz Leśniczek, "Usługi Microsoft Azure. Programowanie aplikacji", APN Promise, 2015 2. Guidici Tony, Redkar Tejaswi, "Platforma Windows Azure", Helion, 2013 	
Literatura dodatkowa:	
<ol style="list-style-type: none"> 1. Zbigniew Fryźlewicz, Daniel Nikończuk, "Windows Azure. Wprowadzenie do programowania w chmurze", Helion, 2012 2. Zbigniew Fryźlewicz, Dariusz Parzygnat, Łukasz Przerada, "Serverless na platformie Azure", Helion, 2019 3. Eric D. Boyd, Leonard Lobel, "Microsoft Azure SQL Database. Krok po kroku", APN PROMISE 2017 	
Planowane formy/działania/metody dydaktyczne:	
Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia laboratoryjne wspomagane technikami multimedialnymi. Zamieszczanie na stronach internetowych problemów i zadań ćwiczeniowych.	
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
<p>Efekty W_01 – W_06 będą sprawdzane podczas ćwiczeń i na egzaminie ustnym. Student będzie odpowiadał na pytania dotyczące zagadnień związanych z programowaniem w chmurze:</p> <ul style="list-style-type: none"> • Co to jest chmura? Na czym polega programowanie w chmurze? • Na czym polega migracja aplikacji do chmury? 	

- Wymień najważniejsze kwestie związane z bezpieczeństwem programowania w chmurze.

Studenci zapoznają się z listą wszystkich pytań na egzamin ustny podczas pierwszego wykładu.

Efekt **U_01 - U_06** będą systematycznie sprawdzane podczas ćwiczeń. Zadania na następne ćwiczenia udostępniane będą kilka dni wcześniej. Student, na podstawie wykładu mogą się do nich przygotować.

Efekty **K_01, K_02** będą weryfikowane, w oparciu o odpowiedzi na pytania zadawane w czasie ćwiczeń laboratoryjnych.

Forma i warunki zaliczenia:

Przedmiot kończy się egzaminem ustnym. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie ćwiczeń. Na zaliczenie ćwiczeń składają się oceny częściowe uzyskane podczas ćwiczeń z nauczycielem akademickim, maksymalnie 110 punktów (10 punktów za każde z ćwiczeń oprócz ćwiczeń ostatnich).

Ćwiczenia laboratoryjne będą zaliczone w wypadku uzyskania powyżej połowy punktów z każdych ćwiczeń.

Podczas egzaminu ustnego można uzyskać maksymalnie 40 punktów. Egzamin będzie zaliczony w przypadku uzyskania powyżej połowy punktów za każde pytanie. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 150 pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 75 punktów: niedostateczna (F),
- 76 – 90 punktów: dostateczna (E),
- 91 – 105 punktów: dostateczna plus (D),
- 106 – 120 punktów: dobra (C),
- 121 – 135 punktów: dobra plus (B),
- 136 – 150 punktów: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godzin
Udział w ćwiczeniach laboratoryjnych	24 godziny
Udział w konsultacjach z przedmiotu	2 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	8 godzin
Sumaryczne obciążenie pracą studenta	75 godzin
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Udział w konsultacjach godz. z przedmiotu	1 godzina
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	35 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	9 godzin
Sumaryczne obciążenie pracą studenta	75 godzin
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Technologie Mobilne	
Nazwa w języku angielskim:	Mobile Technologies	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	Informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	fakultatywny	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:	dr Mirosław Szaban	
Imię i nazwisko prowadzących zajęcia:	dr Mirosław Szaban, mgr Kamil Skarżyński	
Założenia i cele przedmiotu:	<p>Celem wykładu jest zapoznanie studentów z problemami przetwarzania mobilnego. W ramach wykładu omówione zostaną najnowsze technologie mobilne i bezprzewodowe. Wskazana zostanie potrzeba stosowania systemów ruchomych, złożoność problemów w nich występujących jak i sposoby rozwiązywania tychże problemów w oparciu o zaadoptowane metody stosowane w innych gałęziach informatyki. W trakcie laboratoriów studenci nabędą umiejętności projektowania i programowania aplikacji mobilnych w systemie Android, z wykorzystaniem środowiska Android Studio.</p>	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna teorię systemów mobilnych oraz z ideę przetwarzania mobilnego.	K_W02, K_W03, K_W05, K_W07

W_02	Zna zasady nawigacji stosowane w systemach mobilnych zarówno na zewnątrz jak i w budynkach. Ponadto zna zintegrowane systemy nawigacyjne oraz systemy nawigacji satelitarnej.	K_W02, K_W03, K_W05, K_W07
W_03	Zna mobilne systemy komórkowe oraz architekturę i działanie systemu GSM	K_W02, K_W03, K_W05, K_W07
W_04	Zna cechy charakterystyczne mobilnych systemów bazodanowych, transakcje mobilne, zapytania i serwisy zależne od pozycji.	K_W03, K_W05, K_W09
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Student potrafi utworzyć aplikację mobilną na urządzenie z systemem Android.	K_U01, K_U06, K_U10, K_U11, K_U12, K_U15, K_U24
U_02	W aplikacji mobilnej potrafi: zaprojektować interfejs użytkownika, pobierać i zapisać dane do bazy danych lokalnej i zamieszczonej na serwerze, przetwarzać dane rejestrowane przez sensory urządzenia mobilnego, zastosować elementy graficzne.	K_U01, K_U06, K_U10, K_U11, K_U12, K_U15, K_U24
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Potrafi formułować opinie na temat zagadnień z zakresu systemów oraz sieci mobilnych i bezprzewodowych, między innymi nt. jakości rozwiązań stosowanych w ich koncepcji i budowie.	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
Umiejętność programowania w C++/Java. Zaliczone przedmioty: 1. Podstawy programowania 2. Algorytmy i złożoność 3. Bazy danych		
Treści modułu kształcenia:		
Wykład: 1. Wprowadzenie – znaczenie systemów mobilnych. Zapoznanie z ideą przetwarzania mobilnego. Historia rozwoju systemów mobilnych i bezprzewodowych. Podstawy systemów mobilnych. Infrastruktura systemów komórkowych. Systemy satelitarne i sieci ad- hoc. Pokazanie rozkwitu dziedziny oraz czynników wpływających na jej rozwój. Przedstawienie mnogości zastosowań,		

olbrzymich korzyści dla klienta końcowego oraz złożonych i nietrywialnych problemów, jakie stoją przed projektantami nowoczesnych systemów mobilnych.

2. **Pojęcia i definicje.** Podstawowe architektury, klasyfikacje terminali. Rozróżnienie między systemami mobilnymi i bezprzewodowymi. Wykazanie cech i elementów złożonego, sieciowego systemu mobilnego.
3. **Pozycjonowanie i nawigacja użytkowników mobilnych.** Podstawowe pojęcia nawigacyjne, określenie jednostek miar. Sposoby wyznaczania pozycji zliczonej i obserwowanej, urządzenia i systemy pozycjonujące. Nawigacja w budynkach i zintegrowane systemy nawigacyjne. Lokalny charakter informacji pozycyjnej oraz strategię jej uaktualniania. **Propagacja fal radiowych a systemy mobilne.** Fale radiowe i ich charakterystyki. Tłumienie fal. Interferencja międzykanałowa. **Systemy nawigacji satelitarnej GLONASS, Galileo, GPS NAVSTAR oraz Compass.** Historia powstania, architektura i zasada działania satelitarnych systemów nawigacyjnych. Budowa satelity Navstar i odbiornika GPS. Błędy w określaniu pozycji i ich korekcja, systemy różnicowe, opis interfejsów komunikacyjnych.
4. **Systemy komórkowe.** Idea i potrzeba stosowania systemów komórkowych. Podstawowe pojęcia i definicje. **Koncepcja komórki.** Parametry komórki. Powtórne wykorzystanie częstotliwości. Formowanie klastrów. Zwalczanie interferencji. Zwiększanie pojemności systemów komórkowych. Omówienie zjawisk typu „roaming” i „handover”. Wady i zalety rozwiązań komórkowych.
5. **Architektura i działanie systemu GSM (1).** Podstawowe komponenty systemu GSM, budowa i rodzaje terminali komórkowych, zespoły stacji bazowych, część centralowa. Utrzymywanie informacji o położeniu terminala, zestawianie połączeń. Bezpieczeństwo w systemie GSM, technologie transmisji danych w telefonii komórkowej.
6. **Architektura i działanie systemu GSM (2). Przydział kanałów.** Przydział statyczny i dynamiczny. Hybrydowe metody przydziału. Przydział kanałów w systemach specjalizowanych. **Mobilne systemy komunikacyjne.** Infrastruktura systemów komórkowych. Rejestracja i przeniesienie połączenia. Roaming.
7. **Architektura i działanie systemu GSM (3). Bezpieczeństwo systemów mobilnych.** Techniki szyfrowania. Uwierzytelnienie. Ściany ogniowe i systemy bezpieczeństwa.
8. **Systemy łączności bezprzewodowej.** Geostacjonarne i niegeostacjonarne satelitarne systemy komunikacyjne. Systemy dyspozytorskie, trunkingowe i przywoławcze. Telefonie bezprzewodowa, łączność w pasmie obywatelskim. Systemy laserowe, podczerwone i ultradźwiękowe. Standardy Bluetooth i IrDA.
9. **Systemy satelitarne.** Typy systemów i ich charakterystyki. Infrastruktura systemu satelitarnego. GPS. **Sieci ad hoc i systemy sensorowe.** Charakterystyki systemów ad hoc. Algorytmy i protokoły routingu. Sieci sensorowe. Zastosowania.
10. Najnowsza wiedza z zakresu przedmiotu. Współczesne technologie mobilne i bezprzewodowe.

Laboratorium:

1. Wprowadzenie do środowiska Android Studio oraz obsługa jego podstawowych narzędzi. Zapoznanie z strukturą projektu.
2. Klasa Activity. Cykl życia aktywności. Utworzenie adaptera. Obsługa podstawowych metod związanych z cyklem życia aplikacji. Przejścia między aktywnościami.
3. Wykorzystanie klasy Intent do współpracy między aplikacjami. Wyjaśnienie roli manifestu oraz uprawnień.
4. Walidacja, internacjonalizacja. Tworzenie wstępnego interfejsu do uwierzytelnienia. (cdn na następnych laboratoriach).
5. Persystancja danych za pomocą lokalnej bazy danych SQLite.

6. Implementacja aplikacji w podejściu REST na platformie .NET w celu późniejszej interakcji z aplikacją.
7. Obsługa po stronie telefonu aplikacji z poprzednich laboratoriów. Wprowadzenie asynchronicznych zachowań z pomocą klasy Task.
8. Powiadomienia.
9. Podstawowe obiekty grafiki 2D i zaawansowane mechanizmy do obsługi grafiki 2D, obsługa zdarzeń interfejsu dotykowego
10. Sensory
11. Fragmenty i usługi w tle
12. Biblioteki zewnętrzne i współczesne technologie mobilne i bezprzewodowe.

Literatura podstawowa:

1. Dmitry Jemerov, Svetlana Isakova: Kotlin w akcji, Helion 2018
2. Nate Ebel: Mastering Kotlin: Learn advanced Kotlin programming techniques to build apps for Android, iOS, and the web, Packt, 2019
3. Android Studio : wygodne i efektywne tworzenie aplikacji / Adam Gerber, Clifton Craig ; [tłumaczenie Rafał Jońca]. - Gliwice : Helion, cop. 2016. - ISBN 978-83-283-2009-3
4. Android : wprowadzenie do programowania aplikacji / Joseph Anzuzi Jr., Lauren Darcey, Shane Conder ; [tłumaczenie Piotr Rajca]. - Gliwice : Wydawnictwo Helion, cop. 2016. - ISBN 978-83-283-2612-5
5. Android: podstawy tworzenia aplikacji / Andrzej Stasiewicz. - Gliwice : Wydawnictwo Helion, cop. 2014 (wznowienie 2015). - ISBN 978-83-246-7006-2

Literatura dodatkowa:

1. D. P. Agrawal, Q. A. Zeng: Introduction to Wireless and Mobile Systems, Thomson, 4th edition, 2014
2. Bieżąca prasa z branży IT
3. Biznes w świecie mobile : jak zaprojektować, wykonać i wypromować aplikację mobilną / Sylwia Żółkiewska, Małgorzata Rycharska, Noemi Gryczko. - Warszawa : Poltext, 2018. - ISBN 978-83-7561-866-2
4. Wydajne aplikacje dla systemu Android : programuj szybko i efektywnie / Doug Sillars ; [tłumaczenie: Andrzej Watrak]. - Gliwice : Wydawnictwo Helion, cop. 2017. - ISBN 978-83-283-2935-5

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia wspomagane technikami multimedialnymi oraz środowiskami umożliwiającymi symulację systemów mobilnych. Zamieszczanie na stronach internetowych problemów i zadań laboratoryjnych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W_01 – W_04 sprawdzane będą na egzaminie pisemnym w sesji egzaminacyjnej, jako zagadnienia teoretyczne. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania:

- Opisz technologię RFID.
- Scharakteryzuj System Mobilny. Wymień jego elementy i krótko opisz.
- Wykonaj rysunek prawidłowego rozmieszczenia stacji bazowych i uzasadnij.
- Scharakteryzuj pojęcia: pasmo, kanały częstotliwościowe i wiązki kanałów.

Efekty U_01 – U_02 sprawdzane będą na bieżąco, na każdym 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 samodzielnie lub korzystając z konsultacji przygotować. Przykładowe zadanie:

Cel ćwiczeń: Zapoznanie się z sposobem w jaki system Android umożliwia komunikację pomiędzy aktywnościami oraz aplikacjami. Przedstawiona zostanie klasa Intent i metody umożliwiające obsługę jej wewnątrz aktywności.

Zadania:

1. Utwórz nowy projekt z aktywnością zawierającą ekran powitalny aplikacji z przyciskiem start,
2. Utwórz drugą aktywność (StudentsListActivity) która będzie wyświetlać dane za pomocą adaptera i będzie posiadać 2 przyciski,
3. Zaimplementuj przejście z aktywności z punktu 1 (MainActivity) po wciśnięciu przycisku start do aktywności z punktu 2,
4. Utwórz kolejną aktywność AddStudentActivity w której będzie możliwe wpisanie danych nowego studenta w postaci 1 pola tekstowego,
5. Dodaj do aktywności z punktu 2 obsługę przycisku „Add” by przenosił do nowej aktywności (AddStudentActivity) w celu uzyskania informacji na temat nowego elementu,
6. Obsłuż otrzymane rezultaty (z AddStudentActivity) i dodaje je do wyświetlanej listy w aktywności StudentsListActivity,
7. W aktywności AddStudentActivity dodaj przycisk umożliwiający pobranie informacji na temat nowego studenta z książki adresowej telefonu i wypełnij nowe pole tekstowe(inputType: numberSigned w edytorze) w tej aktywności,
8. Zmodyfikuj ListView w aktywności StudentsListActivity by wyświetlić bardziej złożone informacje o studencie. Dodaj wyświetlanie numeru telefonu,
9. Zmodyfikuj adapter tak by dawał możliwość usuwania elementów jak na poprzednich laboratoriach.

Efekt K_01 będzie sprawdzany poprzez wykonanie i wygłoszenie prezentacji dotyczącej najnowszych technologii z zakresu systemów mobilnych.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 50 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu co najmniej 25 pkt.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 40 pkt. Osoby, które zaliczyły laboratorium w terminie, na co najmniej 90% możliwych punktów, mogą skorzystać z egzaminu zerowego ustnego.

Egzamin będzie zaliczony w przypadku uzyskania co najmniej 20 pkt. Ponadto, zaliczenie wykładu obejmuje przygotowanie i wygłoszenie prezentacji na zadane tematy dotyczące najnowszych technologii z zakresu systemów mobilnych, za które można uzyskać do 10 pkt. Wykład będzie zaliczony w przypadku uzyskania co najmniej 5 pkt. z prezentacji. Na tej formie zajęć student może maksymalnie uzyskać 50 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	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godz.
Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godz.

Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	30 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Przetwarzanie Mobilne i Komunikacja Ruchoma	
Nazwa w języku angielskim:	Wireless and Mobile Networks	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	Informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	fakultatywny	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:	dr Mirosław Szaban	
Imię i nazwisko prowadzących zajęcia:	dr Mirosław Szaban	
Założenia i cele przedmiotu:	<p>Celem przedmiotu jest nabycie przez studentów bieżącego stanu rozwoju technologii systemów mobilnych i bezprzewodowych, w szczególności sieci telefonii komórkowej - GSM, w tym: główne zasady funkcjonowania systemów bezprzewodowych, sposoby podtrzymywania mobilności systemów, aktualnie istniejące technologie bezprzewodowe, infrastruktura sieci mobilnych oraz nabycie praktycznych umiejętności w tym zakresie. Ponadto nabycie umiejętności samodzielnego rozwiązywania praktycznych problemów z użyciem symulatorów.</p>	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna strukturę sieci mobilnych i bezprzewodowych w szczególności sieci komórkowych. Wie czym jest system GSM, systemy satelitarne, sieci Ad-hoc i systemy sensorowe.	K_W01, K_W02, K_W03, K_W05, K_W07, K_W10

W_02	Zna pojęcie fali radiowej, efektu tłumienia fal, interferencji oraz efekt Dopplera.	K_W01, K_W02, K_W03
W_03	Student wie czym są komórki sieci komórkowej, jak zbudowany jest klaster sieci.	K_W01, K_W02, K_W07, K_W10
W_04	Wie na czym polega statyczny i dynamiczny przydział kanałów w sieciach komórkowych.	K_W01, K_W02, K_W03, K_W07, K_W10
W_05	Zna mechanizmy bezpieczeństwa (szyfrowania i uwierzytelniania) występujące w sieciach bezprzewodowych.	K_W01, K_W02, K_W03, K_W07, K_W10
W_06	Zna metody generowania ruchu, obsługę zdarzeń i system masowej obsługi.	K_W01, K_W02, K_W03, K_W05, K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Posługuje się wybranymi popularnymi aplikacjami, symulatorami odzwierciedlającymi procesy zachodzące w sieciach komórkowych.	K_U01, K_U02, K_U06, K_U07, K_U08, K_U10, K_U12, K_U15, K_U19, K_U24
U_02	Potrafi korzystać z wybranych środowisk programistycznych i ich bibliotek pod kątem ich wykorzystania w projektowaniu symulatorów sieci komórkowych.	K_U01, K_U02, K_U06, K_U10, K_U11, K_U15
U_03	Potrafi wykonać obliczenia wartości parametrów sieci bezprzewodowych i mobilnych, tj. odległość wykorzystania, zysk i moc anten odbiornika i nadajnika, wpływ efektu Dopplera w zależności od kierunku ruchu stacji mobilnej.	K_U01, K_U02, K_U06, K_U11, K_U15, K_U19,
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Potrafi formułować opinie na temat zagadnień z zakresu systemów oraz sieci mobilnych i bezprzewodowych, między innymi nt. jakości rozwiązań stosowanych w ich koncepcji i budowie.	K_K01, K_K02
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		

1. Umiejętność podstaw programowania i programowania obiektowego.
2. Zaliczenie semestru poprzedzającego przedmiot.

Treści modułu kształcenia:

Wykład:

1. **Wprowadzenie.** Historia rozwoju systemów mobilnych i bezprzewodowych. Podstawy systemów mobilnych. Infrastruktura systemów komórkowych. Systemy satelitarne i sieci ad- hoc.
2. **Propagacja fal radiowych a systemy mobilne.** Fale radiowe i ich charakterystyki. Tłumienie fal. Interferencja międzykanałowa.
3. **Koncepcja komórki.** Parametry komórki. Powtórne wykorzystanie częstotliwości. Formowanie klastrów. Zwalczanie interferencji.
4. **Przydział kanałów.** Przydział statyczny i dynamiczny. Hybrydowe metody przydziału. Przydział kanałów w systemach specjalizowanych.
5. **Mobilne systemy komunikacyjne.** Infrastruktura systemów komórkowych. Rejestracja i przeniesienie połączenia. Roaming.
6. **Bezpieczeństwo systemów mobilnych.** Techniki szyfrowania. Uwierzytelnienie. Ściany ogniowe i systemy bezpieczeństwa.
7. **System GSM oraz system IMT-2000.** Zakresy częstotliwości i kanały dla GSM. Numery identyfikacyjne. UMTS.
8. **Systemy satelitarne.** Typy systemów i ich charakterystyki. Infrastruktura systemu satelitarnego. GPS.
9. **Sieci ad hoc i systemy sensorowe.** Charakterystyki systemów ad hoc. Algorytmy i protokoły routingu. Sieci sensorowe. Zastosowania.
10. **Bezprzewodowe sieci MAN, LAN i PAN.** Bezprzewodowe sieci metropolitalne. Bezprzewodowe sieci lokalne. Bezprzewodowe sieci personalne.
11. **Współczesne technologie mobilne i bezprzewodowe.**

Laboratorium:

1. **Wprowadzenie do laboratorium.** Przedstawienie treści poruszanych w trakcie laboratoriów. Warunki zaliczenia.
2. **Teoretyczne podstawy generowania zdarzeń (ruchu).** Zdarzenia dyskretne. Rozkłady zdarzeń dyskretnych.
3. **Algorytmy i generowanie zdarzeń (ruchu).** Generatory o rozkładach dyskretnych oparte na rozkładzie równomiernym. Generatory o rozkładach ciągłych.
4. **Masowa obsługa zdarzeń.** Modele i charakterystyki. Systemy kolejkowe z pojedynczym kanałem obsługi. Systemy kolejkowe z wielokrotnym kanałem obsługi.
5. **System masowej obsługi (SMO) M/M/S/S.** Analiza procesów zachodzących w SMO. Charakterystyki SMO. Modelowanie systemu.
6. **Koncepcja komórki.** Parametry komórki. Powtórne wykorzystanie częstotliwości. Formowanie klastrów. Zwalczanie interferencji.
7. **Stacja bazowa w sieci komórkowej.** Zadania stacji bazowej. Procesy zachodzące w sieci komórkowej w zasięgu stacji bazowej. Symulator – analiza.
8. **Propagacja fal radiowych w systemach mobilnych.** Tłumienie fal. Efekt Dopplera. Interferencje.
9. **Modelowanie topologii sieci komórkowej.** Powtórne wykorzystanie częstotliwości. Zwalczanie interferencji. Modelowanie systemu sieci komórkowej.

10. **Przydział częstotliwości i zapożyczanie kanałów w sieciach komórkowych.** Opis problemu. Koncepcja rozwiązania problemu. Optymalizacja z użyciem algorytmu genetycznego.
11. **Rozmieszczenie stacji raportujących i nieraportujących.** Opis problemu. Koncepcja rozwiązania. Zastosowanie algorytmu genetycznego i Tabu Search.
12. **Współczesne technologie mobilne i bezprzewodowe.**

Literatura podstawowa:

1. D. P. Agrawal, Q. A. Zeng: Introduction to Wireless and Mobile Systems, Thomson, 2016
2. K. Smith: Mobile Communications and Networks, Larsen and Keller Education, 2017
3. J. Januszewski. Systemy satelitarne GPS, Galileo i inne. PWN, 2010.
4. Bieżąca prasa z branży IT

Literatura dodatkowa:

1. S. Basagni et al.: Mobile Ad Hoc Networking, Wiley, 2013

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:

Efekty W_01 – W_06 oraz K_01 sprawdzane będą na egzaminie jako zagadnienia teoretyczne z wykładu. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania:

- Jaka jest zaleta podziału komórki na sektory?
- Jaka jest różnica pomiędzy interferencją sąsiedniego kanału (adjacent channel interference) a interferencją współkanałową?
- Co oznacza przeniesienie połączenia (handoff) i obszar przeniesienia (handoff region). Wyjaśnij pojęcia przy pomocy odpowiedniego rysunku.
- Jaka jest różnica pomiędzy szybkim i wolnym tłumieniem sygnału (fast and slow fading)?

Efekty U_01 – U_03 sprawdzane będą na bieżąco, na każdym 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 samodzielnie lub korzystając z konsultacji przygotować. Przykładowe zadanie:

Na podstawie teorii modelowania systemu (M/M/S/S), dokonaj implementacji symulatora stacji bazowej.

Parametry:

- Liczba kanałów,
- λ - parametr natężenia ruchu w rozkładzie Poissona,
- N – średnia długość rozmowy, wartość oczekiwana w rozkładzie Gaussa,
- σ - odchylenie standardowe w rozkładzie Gaussa,

- Min - minimalna długość rozmowy,
- Maks - maksymalna długość rozmowy,
- Długość kolejki,
- Czas symulacji.

Kontrolki:

- Graficzne przedstawienie połączeń kanałach + liczba obsłużonych + czas obsługi bieżącego połączenia,
- Czas symulacji.

Wyniki:

- Wykresy: ρ - Intensywność ruchu, Q - średnia długość kolejki, W – średni czas oczekiwania.
- Plik: Parametry symulacji, ρ , Q, W poniżej w kolumnach.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 50 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu co najmniej 25 pkt.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 40 pkt. Osoby, które zaliczyły laboratorium w terminie, na co najmniej 90% możliwych punktów, mogą skorzystać z egzaminu zerowego ustnego. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 20 pkt. Ponadto, zaliczenie wykładu obejmuje przygotowanie i wygłoszenie prezentacji na zadane tematy dotyczące najnowszych technologii z zakresu systemów mobilnych, za które można uzyskać do 10 pkt. Wykład będzie zaliczony w przypadku uzyskania co najmniej 5 pkt. z prezentacji. Na tej formie zajęć student może maksymalnie uzyskać 50 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

21 godz.

Udział w ćwiczeniach laboratoryjnych

24 godz.

Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godz.
Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godz.
Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	30 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Zintegrowane systemy informatyczne zarządzania
Nazwa w języku angielskim:		Integrated Management Information Systems
Język wykładowy:	Polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Ewa Szczepanik
Imię i nazwisko prowadzących zajęcia:		dr Marek Piłski
Założenia i cele przedmiotu:		Celem kursu jest przedstawienie zagadnień związanych z zintegrowanymi systemami zarządzania, min. to: struktura, architektura, standardy oraz nabycie praktycznych umiejętności związanych z ich zarządzaniem oraz projektowaniem
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie rolę systemu informatycznego w przedsiębiorstwie oraz sposoby (etapy, metody) wprowadzania TI do korporacji.	K_W07, K_W10
W_02	Zna i rozumie zagadnienia z zakresu typologii, standardów, struktury i architektury typowych systemów informatycznych zarządzania.	K_W07, K_W10
W_03	Zna i rozumie zagadnienia z zakresu złożoności realizacyjnej SI oraz zna sposoby (scenariusze) realizacji zintegrowanych systemów informatycznych zarządzania.	K_W07, K_W13
W_04	Zna i rozumie zagadnienia z zakresu wdrażania do przedsiębiorstwa zintegrowanych systemów informatycznych zarządzania.	K_W07, K_W13
W_05	Zna i rozumie zagadnienia z zakresu zastosowania niektórych branżowych systemów informatycznych zarządzania oraz systemu SAP.	K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego

U_01	Potrafi na podstawie literatury formułować wnioski dotyczące najnowszych rozwiązań systemów komputerowych i wskazywać sposób ich powiązania z ogólnymi zasadami realizacji pracy systemu.	K_U01
U_02	Posiada umiejętność samokształcenia się, m.in. w celu podnoszenia kompetencji zawodowych	K_U06
U_03	Potrafi scharakteryzować systemy informatyczne ze względu na zadane kryteria funkcjonalne, użytkowe i ekonomiczne.	K_U13
U_04	Potrafi przygotować prezentację dotyczącą charakterystyki i właściwości funkcjonalno-użytkowych wybranych SIz	K_U01, K_U02
U_05	Potrafi wskazać rodzaj systemu informatycznego do określonych zastosowań w praktyce zarządzania przedsiębiorstwem.	K_U20, K_U23
U_06	Potrafi wykorzystać wybrany SI do realizacji prostych zadań biznesowych.	K_U21
U_07	Potrafi zaprojektować, zaimplementować oraz przetestować - zgodnie z zadaną specyfikacją - system informatyczny typowy dla kierunku Informatyka	K_U19
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
K_02	jest gotów do wypełniania zobowiązań społecznych, współorganizowania działalności na rzecz środowiska społecznego oraz inicjowania działań na rzecz interesu publicznego poprzez przekazywanie informacji i opinii dotyczących osiągnięć informatyki i innych aspektów działalności inżyniera-informatyka; podejmuje starania, aby przekazać takie informacje i opinie w sposób zrozumiały	K_K02
K_K03	jest gotów do myślenia i działania w sposób przedsiębiorczy w zakresie systemów informatycznych zarządzania	K_K03
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
1. Umiejętność wykorzystania podstaw inżynierii tworzenia systemów informatycznych oraz znajomość ogólnych problemów społecznych i zawodowych informatyki		
Treści modułu kształcenia:		

1. **Rola zasobów informacyjnych w przedsiębiorstwie.** Systemy organizacji. Zasoby informacyjne w przedsiębiorstwie. Procesy informacyjno-decyzyjne. Informacja a procesy podejmowania decyzji
2. **Model zastosowań technologii informatycznej w organizacji.** Zastosowanie pojedynczych aplikacji IT. Wewnętrzna integracja. Reorganizacja procesów biznesowych. Reorganizacja sieci gospodarczej. Zmiana zakresu działalności gospodarczej
3. **Typologia i ewolucja informatycznych systemów zarządzania i przegląd wybranych informatycznych systemów zarządzania.** Systemy informatyczne zarządzania. Typologia SIZ. Standardy SIZ
4. **Struktury i architektura SIZ.** Struktury SIZ. Architektury SIZ
5. **Podejście procesowe w zarządzaniu i jego wpływ na kierunki rozwoju SIZ.** Istota podejścia procesowego w zarządzaniu. Związek podejścia procesowego z koncepcjami współczesnych SIZ. Podejście procesowe w systemach jakości
6. **Złożoność realizacyjna SIZ i scenariusze realizacji SIZ.** Czynniki występujących podczas realizacji systemu. Składniki systemu jako produktu końcowego. Tworzenie ZSI od podstaw przez służby informatyczne przedsiębiorstwa. Tworzenie ZSI od podstaw przez zewnętrzne firmy informatyczne. Wybór, zakup i wdrożenie wyrobu gotowego
7. **Procedura wyboru gotowego ZSI.** Kryteria wyboru ZSI. Opis procedury. Fazy procedury wyboru gotowego ZSI
8. **Organizacja prac wdrożeniowych ZSI.** Model i etapy wdrażania ZSI. Realizacja ZSI. Działania integrujące (integrator wdrożeniowy). Przykładowy model integracji działań
9. **Współczesne zintegrowane systemy informatyczne zarządzania.** Klasyfikacja i podział współczesnych zintegrowanych SIZ. Przegląd dostępnych na rynku polskim SIZ. Tendencje rozwojowe współczesnych SIZ
10. **Zapoznanie z działaniem wybranych systemów informatycznych zarządzania.** Systemy branżowe: Abra Gold firmy AKTIS i CDN OPT!MA firmy Comarch. Praca z systemem SAP.
11. **Przegląd wybranych informatycznych systemów zarządzania.** Prezentacja wybranej aplikacji (systemu) przez grupę studentów - wspomaganie zarządzania instytucją oświatową, administracji państwowej, administracji gospodarczej, służby zdrowia itp.

Literatura podstawowa:

1. Barczak A., Florek J., Sydoruk T.: *Projektowanie zintegrowanych systemów informatycznych zarządzania*. Wydawnictwo Akademii Podlaskiej, Siedlce 2006
2. Jabłoński W., Bartkiewicz W.: *Systemy informatyczne zarządzania: klasyfikacja i charakterystyka systemów*, Bydgoszcz, 2006
3. Kisielnicki J., Sroka H.: *Systemy informacyjne biznesu. Informatyka dla zarządzania*. Agencja Wydawnicza PLACET, Warszawa 2005.
4. Isajew J.G.: *Systemy informacyjne*. Expertus, Warszawa 2001
5. Adamczewski P.: *Zintegrowane systemy informatyczne w praktyce*. MIKOM, Warszawa 2000

Literatura dodatkowa:

1. Chmielarz W.: *Systemy informatyczne wspomagające zarządzanie. Aspekt modelowy w budowie systemów*. Elipsa, Warszawa 1996
2. Vivek K.: *SAP R/3. Przewodnik dla menadżerów*. HELION, Gliwice 2001
3. Mazzullo J., Wheatley P.: *SAP R/3. Podręcznik użytkownika*. HELION, Gliwice 2006.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratoria wspomagane technikami komputerowymi i przykładowymi systemami informatycznymi zarządzania. Zamieszczanie na stronach internetowych problemów i zadań laboratoryjnych.

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty W_01 – W_04, W_05, U_03, U_05 sprawdzane będą na semestralnym kolokwium.

Przykładowe pytania z kolokwium:

1. Przedstaw i krótko omów 5-cio fazowy model pozwalający świadomie stosować IT dla potrzeb biznesu .
2. Podaj podział typologiczny Systemów Informatycznych Zarządzania
3. Przedstaw architekturę SIZ na podstawie siatki Zamana.
4. Wymień i krótko opisz scenariusze realizacji ZSIZ w przedsiębiorstwie.
5. Podaj właściwości funkcjonalne systemu SAP

Efekt U_01, U_02, U_06, U_07 będą systematycznie sprawdzane na zajęciach laboratoryjnych w trakcie wykonywania prostych zadań. Przykładowe zadania.

1. Omów pojęcia: „zadanie krytyczne” i „ścieżka krytyczna” w kontekście planowania przedsięwzięć gospodarczych
2. Zdefiniuj nowy zasób typu „praca” i przydziel go do wybranego zadania projektowego w wymiarze 50% dostępnych jednostek w narzędziu MS Project
3. Przedstaw dostępne typy relacji pomiędzy zadaniami projektowymi w harmonogramach Gantta i podaj stosowne przykłady zadań pasujące do omawianych typów relacji
4. Podaj przykład zadania cyklicznego i dodaj je w MS Project
5. Wykonaj bilans otwarcia dnia z 100 sztukami wskazanego towaru w narzędziu Optima firmy Comarch
6. Sprzedaj dany towar nowemu kontrahentowi wystawiając fakturę płatną w dwóch równych ratach w narzędziu Optima firmy Comarch
7. Zatrudnij nowego pracownika w wymiarze ½ etatu, ustal mu miesięczne wynagrodzenie oraz premie w narzędziu Optima firmy Comarch

Zadania na każde następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować.

Efekty U_04 oraz K_01, K_02 i K_03 weryfikowane, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych oraz podczas zaprezentowanej przez studenta prezentacji.

Przykładowe zadania

1. Przedstaw funkcjonalność wybranego narzędzia z sektora służby zdrowia, pokaż realizację dwóch wybranych/kluczowych funkcji aplikacji
2. Przedstaw funkcjonalność wybranego narzędzia z branży IT, pokaż realizację dwóch wybranych/kluczowych funkcji aplikacji

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i jednego kolokwium pisemnego przeprowadzonego na ostatnim wykładzie. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- Regularne zajęcia – 39 pkt.,
- Obrona zadania indywidualnego – 21 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 10 pkt. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt.

Za pisemne kolokwium można na nim uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu co najmniej 20 pkt. Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) 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 kolokwium w trakcie zajęć w semestrze. Dwie poprawy oraz jedno przedstawienie prezentacji w sesji egzaminacyjnej.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do kolokwium pisemnego i obecność na kolokwium	10 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do kolokwium pisemnego i obecność na kolokwium	15 godz.
Sumaryczne obciążenie pracą studenta	75 godz.

Punkty ECTS za przedmiot	3 ECTS
--------------------------	--------

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Komputerowe systemy wspomaganie decyzji
Nazwa w języku angielskim:		Computer-Aided Decision System
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Anna Wawrzyńczak-Szaban
Imię i nazwisko prowadzących zajęcia:		dr Anna Wawrzyńczak-Szaban
Założenia i cele przedmiotu:		Celem kursu jest zapoznanie studentów z zagadnieniami oraz systemami wspomagającymi podejmowanie decyzji w nowoczesnych przedsiębiorstwach.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia z zakresu standardów i nowych technologii informacyjnych zarządzania, rolę decyzji w nowoczesnym przedsiębiorstwie i zakres i horyzont czasowy podejmowania decyzji.	K_W10, K_W03
W_02	Zna i rozumie modele podejmowania decyzji oraz zasady i zakres analitycznego modelowania sytuacji decyzyjnej i modelowania preferencji decydenta indywidualnego, grupowego oraz zespołów decyzyjnych.	K_W13
W_03	Zna i rozumie definicję, istotę oraz elementy składowe komputerowych systemów wspomaganie decyzji i ich związki z współczesnymi systemy informacyjne MIS, EIS, ES itd.	K_W06
W_04	Zna i rozumie podstawowe definicje oraz metody rozwiązywania zadań analizy wielokryterialnej, a także metody reprezentacji niepewności reprezentowane przez modele probabilistyczne, logikę rozmytą i zbiory rozmyte i przybliżone.	K_W01
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi przeprowadzić analizę wypracowanych rozwiązań decyzyjnych i przygotować przekonujące ich uzasadnienia.	K_U01, K_U02, K_U19, K_U20, K_U23, K_U13

U_02	Potrafi wskazać rodzaj systemu informatycznego do określonych zastosowań w procesie analizy i eksploracji danych i wspomagania decyzji zarządczych.	K_U17
U_03	Potrafi wykorzystać wybrane narzędzie (MatLab, Pakiet MS Access, ExpertChoice i AHPproject) do analizy danych i wspomagania decyzji.	K_U15
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
1. Umiejętność rozwiązywania równań algebraicznych, układów równań liniowych oraz znajomość podstaw rachunku prawdopodobieństwa i podstaw programowania.		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> Podstawy wspomagania decyzji. Rola decyzji w nowoczesnym przedsiębiorstwie, Charakterystyka sytuacji decyzyjnej. Zakres podejmowania decyzji. Horyzont czasowy podejmowanych decyzji. Charakter problemów, decydent. Schematy podejmowania decyzji i analityczne podejście do podejmowania decyzji Informacyjne systemy zarządzania i modele podejmowania decyzji. Pojęcie informacyjnego systemu zarządzania. Informacyjne systemy zarządzania i poziomy organizacyjne. Decyzje podejmowane w zależności od poziomu zarządzania. Modelowanie rzeczywistości, baza modelu Wprowadzenie do komputerowych systemów wspomagania decyzji. Systemy wspomagania decyzji a współczesne systemy informacyjne MIS, EIS, ES itd. Definicja systemu wspomagania decyzji. Struktura systemu wspomagania decyzji. Charakterystyka elementów składowych Analityczne modelowanie sytuacji decyzyjnej. Sytuacja decyzyjna a model. Modelowanie analityczne sytuacji decyzyjnej. Elementy składowe modelu sytuacji decyzyjnej. Model rzeczowy sytuacji decyzyjnej Modelowanie preferencji decydenta. Koncepcja specyfikacji preferencji za pomocą funkcji realizacji celu. Decydenci indywidualni i grupowi, tworzenie efektywnych zespołów. Wspomaganie komputerowe decydenta indywidualnego i grupowego. Przykłady modelowania sytuacji decyzyjnej. Metody rozwiązywania zadań analizy wielokryterialnej. Analiza wielokryterialna. Podstawowe definicje. Przegląd metod rozwiązywania zadań analizy wielokryterialnej Metody reprezentacji niepewności. Modele probabilistyczne. Logika rozmyta i zbiory rozmyte. Zbiory rozmyte i przybliżone Wydobywanie wiedzy z danych. Modele danych, analiza wielowymiarowa. Eksploracja i analiza danych jako narzędzia wspomagania decyzji Systemy ekspertowe. Baza wiedzy i metody reprezentacji wiedzy w bazie wiedzy. Maszyna wnioskująca. Algorytmy wnioskowania (do przodu, wstecz, mieszane, heurystyki). Przykład systemu ekspertowego w systemie wspomagania decyzji. Wykorzystanie wybranych narzędzi do analizy danych i wspomagania decyzji. Pakiet MatLab, ExpertChoice i AHPproject Przegląd wybranych informatycznych systemów zarządzania. Prezentacje (grupowe) studenckie wybranych SWD: Statistica, SPSS Clementine, Oracle Data Mining, Oracle Text , Insightful Miner, Weka, Oracle Discoverer, Cognos /Applix/ TM1, SAS 		
Literatura podstawowa:		
<ol style="list-style-type: none"> Kwiatkowska A. M.: <i>Systemy wspomagania decyzji</i>, PWN, Warszawa 2007. Wierzbicki A.P., Granat J.: <i>Optymalizacja we wspomaganiu decyzji</i>, Preskrypt, Instytut Automatyki i Informatyki Stosowanej, Politechnika Warszawska, 2003 L. Knopik , W. Bojar , K. Rostek <i>Systemy wspomagania decyzji</i>, PWE, 2014 		

Literatura dodatkowa:

1. Sirpińska M., Jachna T.: *Metody podejmowania decyzji finansowych*, PWN, Warszawa 2007.
2. Czerminski J., *Systemy wspomaganie decyzji w zarządzaniu przedsiębiorstwem*, Dom Organizatora, TNOiK, Torun-Gdansk 2002
3. Bolesta-Kukułka K.: *Decyzje menedżerskie*, Wyd. PWE Warszawa 2003
4. Sroka H.: *Inteligentne systemy wspomaganie decyzji w zarządzaniu*, AE Katowice 1996
5. Turban, E.: *Decision Support and Expert Systems*. Prentice-Hall. London. 1995

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko obliczeń naukowych MatLab, ExpertChoice i AHPproject. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty W_01÷ W_04 będą weryfikowane kolokwium pisemnym na ostatnim wykładzie.

Przykładowe pytania:

- Scharakteryzuj następujące pojęcia: sytuacja decyzyjna i problem decyzyjny
- Omów metody wydobywania wiedzy z danych.
- Podaj przykład min. 2 komputerowych systemów decyzyjnych i opisz szczegółowo jeden z nich
- Czym charakteryzuje się aktywny System Wspomaganie Decyzji ?

Efekty U_01÷ U_04 sprawdzane będą na bieżąco, na każdych zajęciach poza pierwszym i ostatnim, a także podczas zaliczania zadania indywidualnego.

Przykładowe zadania:

- W pewnej firmie zauważono, że istnieje duża zależność pomiędzy zmianą ceny a zmianą sprzedaży w sztukach. Dane po kwartałach dla lat 2008-2012 przedstawione są w tabeli. Przeanalizuj wskazane zależności i określ, jaka może być sprzedaż w kolejnym roku w przy rozważanych zmianach cen. Zapoznaj się z przygotowanym zadaniem optymalizacji portfolio oraz przeprowadź przykładowe obliczenia i przedyskutuj otrzymane wyniki.

Tematyka zajęć laboratoryjnych zostanie podana, co najmniej tydzień przed zajęciami.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i jednego kolokwium pisemnego przeprowadzonego na ostatnim wykładzie. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego polegającego na samodzielnym opracowaniu rozwiązania wybranego problemu decyzyjnego. według schematu:

Regularne zajęcia – 39 pkt.,

Obrona zadania indywidualnego – 21 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 10 pkt. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt.

Za pisemne kolokwium można na nim uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu, co najmniej 20 pkt. Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) 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:

Uzyskanie poprawkowego zaliczenia laboratoriów oraz wykładu możliwe jest w trakcie sesji egzaminacyjnej..

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do kolokwium	10 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do kolokwium	15 godz.
Sumaryczne obciążenie pracą studenta	75 godz.

Punkty ECTS za przedmiot	3 ECTS
--------------------------	--------

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Modelowanie i wizualizacja grafiki 3D
Nazwa w języku angielskim:		Modeling and Visualisation 3D Graphics
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		Fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		Mirosław Barański
Imię i nazwisko prowadzących zajęcia:		Mirosław Barański Mirosław Szaban
Założenia i cele przedmiotu:		Założono, że studenci poznają ważniejsze (i wybrane) aspekty związane z modelowaniem grafiki 3D, wykorzystywane w praktyce środowiska, ich możliwości. Ma to być wstęp do dalszych samodzielnych studiów w tym zakresie. Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z metodami i narzędziami modelowania i wizualizacji grafiki 3D oraz zdobycie praktycznych umiejętności w implementacji tych algorytmów i metod oraz poznanie wybranych narzędzi informatycznych wspomagających modelowanie i wizualizację.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia z zakresu modelowania obiektów podstawowych obiektów graficznych: krzywe, powierzchnie, figury geometryczne i ich przetwarzania	K_W08
W_02	Zna i rozumie zagadnienia związane z określaniem własności obiektów graficznych	K_W08

W_03	Zna obecny stan wiedzy oraz orientuje się w najnowszych trendach rozwojowych z zakresu modelowania i wizualizacji grafiki 3D	K_W08
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł na temat zagadnień związanych z modelowaniem i wizualizacją; a następnie integrować uzyskane informacje, dokonywać ich interpretacji i wyciągać wnioski oraz formułować i uzasadniać opinie	K_U01
U_02	posiada umiejętność samokształcenia się, m.in. w celu podnoszenia kompetencji zawodowych	K_U06
U_03	potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązywania zadań związanych z modelowaniem i wizualizacją oraz wybierać i stosować właściwe metody i narzędzia	K_U09, K_U10
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy z zakresu rozpoznawania obrazów	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia (24 godz), Studia niestacjonarne: wykłady (15 godz.), ćwiczenia (15 godz),	
Wymagania wstępne i dodatkowe:		
1. Umiejętność programowania, 2. Umiejętność posługiwania się pojęciami geometrii obliczeniowej z podstawowym zakresie.		
Treści modułu kształcenia:		

1. Modelowanie krzywych i powierzchni w modelowaniu obiektów grafiki 3D, metody reprezentacji. Własności, rodzaje, zastosowania. Wyznaczanie powierzchni widocznych.
2. Bryły w modelowaniu obiektów grafiki 3D i metody ich reprezentacji.
3. Wybrane zagadnienia dotyczące światła i barw. Fizyka światła i jej wpływ na modele kolorów. Podstawowe zagadnienia związane z percepcją postrzegania barw przez człowieka.
4. Modele kolorów: Model RGB i CMY, Model HSV, Obliczenia w przestrzeni barw.
5. Modelowanie scen. Podstawowe elementy sceny: obiekty, obserwator i oświetlenie. Proste modele scen. Modele scen uwzględniające wybrane efekty oświetlenia: własne oświetlenie, odbicie, odbicie zwierciadlane, załamanie.
6. Modelowanie scen - metoda ray-tracingu. Podstawy matematyczne metody ray-tracingu, uwzględnianie wybranych efektów geometrycznych w metodzie ray-tracingu. Podstawy metody energetycznej.
7. Tekstura - podstawowe zagadnienia. Pojęcie tekstury, metody generowania tekstury.
8. Modelowanie i wizualizacja grafiki 3D w HTML5.
9. Systemy grafiki komputerowej implementujące wybrane metody modelowania scen - aplikacja 3dsMax, Efekty spacje dostępne w 3ds max.
10. Programowanie scen w aplikacji 3ds max - ogólna charakterystyka języka programowania. Podstawowe i złożone obiekty w 3ds max. Operacje na obiektach. Tekstury w programie 3dsmax. Efekty specjalne w programie 3ds max.
11. Przegląd wybranych systemów wspomagających modelowanie scen. Modelowanie scen za pomocą: SketchUp, Blender, Unity.

Literatura podstawowa:

1. Zabrodzki J. i inni : Grafika komputerowa, metody i narzędzia. WNT 1994.
2. Michał Jankowski: Elementy grafiki komputerowej. WNT 2006.
3. James D. Foley: Wprowadzenie do grafiki komputerowej. WNT 2001.
4. Tony Parisi, Aplikacje 3D. Przewodnik po HTML5, WebGL i CSS3, Helion 2014.

Literatura dodatkowa:

1. Kelly L. Murdock, 3ds Max 8. Biblia, Helion 2007,
2. Bogdan Bociek, Blender. Podstawy modelowania, Helion 2014,
3. Mike Geig, Unity. Przewodnik projektanta gier. Wydanie III, Helion 2019,
4. Aleksandra Tomaszewska, Google SketchUp. Ćwiczenia praktyczne, Helion 2009.

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekt W_01, W_02 będą weryfikowane na egzaminie w postaci indywidualnych zadań do wykonania. Przykładowe zadania dotyczyć będą obliczania krzywych lub powierzchni dla konkretnych danych.

Efekt W_03 będzie weryfikowany poprzez opracowanie dotyczące wybranego problemu grafiki 3D związanego z modelowaniem i wizualizacją na podstawie samodzielnych studiów literaturowych.

Efekty U_01, U_02 i U_03 będą weryfikowane na zajęciach laboratoryjnych w postaci zadań. Zadania będą udostępniane co najmniej tydzień wcześniej w postaci strony WWW.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Na zaliczenie składają się: kolokwium ustne i zajęcia laboratoryjne. Na ocenę laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 60 pkt. Zaliczenie zajęć laboratoryjnych jest możliwe po uzyskaniu co najmniej 30 pkt.

Na kolokwium ustnym można uzyskać do 40 pkt. Kolokwium będzie zaliczone w przypadku uzyskania co najmniej 20 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) 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 każdego zajęcia laboratoryjnego. W przypadku obecności usprawiedliwionej można na poprawie uzyskać maksymalnie 100% punktów, a w przypadku obecności nieusprawiedliwionej maksymalnie 80% punktów.

Uwaga: Istnieje możliwość zwolnienia z egzaminu pisemnego lub ustnego studentów wyróżniających się na zajęciach laboratoryjnych. Warunkiem koniecznym zwolnienia z kolokwium ustnego jest uzyskanie 90% punktów możliwych do zdobycia w trakcie regularnych zajęć laboratoryjnych. Decyzję o ewentualnym zwolnieniu podejmuje osoba przeprowadzająca egzamin po zasięgnięciu opinii (np. poprzez rozmowę) osób prowadzących zajęcia. Decyzję o zwolnieniu prowadzący wykład przekazuje studentom nie później niż 2 tygodnie przed końcem semestru.

Bilans punktów ECTS:**Studia stacjonarne**

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach	24 godz.
Samodzielne przygotowanie się do ćwiczeń	30 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach	15 godz.
Samodzielne przygotowanie się do ćwiczeń	40 godz.
Udział w konsultacjach godz. z przedmiotu	4 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	26 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Rozpoznawanie obrazów
Nazwa w języku angielskim:		Image recognition
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:		Wydział Nauk Ścisłych i Przyrodniczych
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Andrzej Salamończyk
Imię i nazwisko prowadzących zajęcia:		dr Andrzej Salamończyk dr Mirosław Barański
Założenia i cele przedmiotu:		Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z algorytmami i metodami rozpoznawania obrazów oraz zdobycie praktycznych umiejętności w implementacji tych algorytmów i metod
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia związane z metodami i algorytmami rozpoznawania obrazów opartymi o metody probabilistyczne i metody wzorców	K_W08
W_02	Zna i rozumie zagadnienia związane z metodami i algorytmami rozpoznawania obrazów opartymi o metody aproksymacyjne	K_W08
	Zna i rozumie zagadnienia związane z metodami i algorytmami rozpoznawania obrazów opartymi o metody sztucznej inteligencji	K_W08
	Zna i rozumie zagadnienia związane z metodami i algorytmami rozpoznawania obrazów opartymi o metody ciągowe	K_W08

	Zna i rozumie zagadnienia związane z wykorzystaniem funkcji graficznych html5 w rozpoznawaniu obrazów	K_W08
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi pozyskiwać informacje dotyczące z literatury, baz danych i innych źródeł, potrafi integrować pozyskane informacje z innymi zadaniami	K_U01
U_02	Potrafi samodzielnie planować i realizować uczenie się przez całe życie, m.in. w celu podnoszenia kompetencji zawodowych	K_U06
	Potrafi wykorzystać właściwie dobrane środowiska programistyczne i narzędzia komputerowe do projektowania i weryfikacji systemów informatycznych związanych z przetwarzaniem obrazów	K_U11
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy z zakresu rozpoznawania obrazów	K_K01
K_02	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów praktycznych i poznawczych z zakresu rozpoznawania obrazów	K_K01
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
1. Znajomość analizy matematycznej, algebry, podstaw programowania, technologii WWW oraz zagadnień sztucznej inteligencji.		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Wstęp. Prezentacja dziedziny. Podstawowe wiadomości i pojęcia związane z obrazami: formy przetwarzania danych obrazowych, grafika, przetwarzanie obrazów, rozpoznawanie obrazów, formy danych obrazowych, grafika wektorowa i rastrowa, przekształcenia form danych obrazowych, 2. Algorytmy NN: Algorytmy oparte o rozpoznanie i analizę cech badanych obiektów 3. Metody wzorców: Algorytmy oparte o badanie przynależności danego obrazu do danej klasy. 4. Metody aproksymacyjne: Algorytmy związane z rozwijanie funkcji przynależności w szereg względem wybranych funkcji bazowych, 5. Sieci neuronowe: Wykorzystanie sieci neuronowych w rozpoznawaniu obrazów, 6. Inne metody sztucznej inteligencji: Wykorzystanie metod i narzędzi sztucznej inteligencji do rozpoznawania obrazów, 		

7. Metody probabilistyczne: Metody oparte o warunkowe rozkłady prawdopodobieństwa,
8. Metody ciągowe: kody łańcuchowe Freemana,
9. Języki opisów obrazów: Język Shawa, języki opisu kształtów.

Literatura podstawowa:

3. R.Tadeusiewicz, M.Fłasiński, Rozpoznawanie obrazów, WNT, 1991.
4. J.Koronacki, J.Ćwik, Statystyczne systemy uczące się, WNT, Warszawa, 2005,
5. K. Stąpor, Automatyczna klasyfikacja obrazów, EXIT, 2005

Literatura dodatkowa:

4. A. Kaehler, G.Bradszi, OpenCV 3: Komputerowe rozpoznawanie obrazów w C++ przy użyciu biblioteki OpenCV, Helion 2017.
5. M.Domański, Obraz cyfrowy, WKŁ 2009.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Zamieszczanie na stronach internetowych zadań i materiałów do laboratoriów.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W_01 – W_04 będą sprawdzane na kolokwium ustnym. Zadania będą dotyczyły wybranych problemów przetwarzania obrazów, przykładowe zadania:

- Omów metodę wzorców, podaj przykłady jej wykorzystania.
- Omów metodę kodów łańcuchowych Freemana, Jakie algorytmy są wykorzystywane w tej metodzie.

Efekty U_01 - U_02 będą systematycznie sprawdzane na zajęciach. Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować.

Efekt U_03 będzie sprawdzany systematycznie na zajęciach laboratoryjnych, przykładowe zadanie:

- Dany jest obraz zapisany w formacie BMP. Stosując algorytm NN znajdź je.

Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami.

Efekty K_K01, K_K02 będą sprawdzane na kolokwium ustnym. Przykładowe zadanie:

- W jaki sposób kody łańcuchowe Freemana można wykorzystać do rozpoznawania pisma.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i egzaminu pisemnego. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na laboratoriach oraz z samodzielnie wykonanego zadania indywidualnego:

- Oceniane laboratoria (10 zajęć po 5 pkt.)

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów tj. co najmniej 51pkt Ostateczny wynik punktowy modułu oblicza się wg wzoru:

$$P=L+50,$$

gdzie P-końcowy wynik punktowy (maksymalnie 100pkt.) , L-punkty uzyskane z części laboratoryjnej (maksymalnie 50), E-punktowy wynik egzaminu (maksymalnie 50)

Ocena z zajęć zależy od końcowego wyniku punktowego i wyznacza się w następujący sposób.

- 0-50 punktów – 2
- 51-60 punktów – 3
- 61-70 punktów - 3,5
- 71-80 punktów – 4
- 81-90 punktów – 4,5
- 91-100 punktów – 100

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godzin
Udział w ćwiczeniach laboratoryjnych	24 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Przygotowanie się do egzaminu i obecność na egzaminie	23 godzin
Sumaryczne obciążenie pracą studenta	100 godzin
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	35 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny

Przygotowanie się do egzaminu i obecność na egzaminie	33 godziny
Sumaryczne obciążenie pracą studenta	100 godzin
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Technologie Programistyczne - Systemy Internetowe
Nazwa w języku angielskim:		Programming Technologies in Internet Systems
Język wykładowy:	Polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		Fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Trzeci	
Semestr:	Piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		Dr Dariusz Mikułowski
Imię i nazwisko prowadzących zajęcia:		Dr Dariusz Mikułowski
Założenia i cele przedmiotu:		<p>Do realizacji przedmiotu niezbędne jest aby studenci:</p> <ul style="list-style-type: none"> • posiadali podstawową wiedzę na temat działania sieci podstawowych języków, technologii i protokołów jakie są w niej wykorzystywane. • Studenci powinni znać model sieci, zasadę działania protokołów Internetowych takich jak FTP czy HTTP oraz podstawową znajomość języka HTML. • Studenci powinni znać wzorzec projektowy MVC i 3-warstwową architekturę aplikacji webowej. • Do zajęć niezbędne są stanowiska komputerowe z zainstalowanym na nim J2EE oraz środowiskiem programistycznym STS, NetBeans lub IntelliJ Idea. • Celem kursu jest zapoznanie studentów z zagadnieniami związanymi z tworzeniem aplikacji webowych zwłaszcza technologiami służącymi do implementowania front-endowej części takich aplikacji.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie podstawowe i zaawansowane konstrukcje programistyczne języków Java, JavaScript oraz bibliotek takich jak JQuery służących do programowania aplikacji webowych.	K_W06
W_02	Zna technologie i frameworki do tworzenia front-endowej i back-endowej części aplikacji webowej takie jak: ThymeLeaf, Angular.js, Bootstrap,	K_W12, K_W10

	Node.js, Bazy danych MongoDB, React, VUE i inne, zna uniwersalny format do przesyłania obiektów pomiędzy komponentami aplikacji JSON.	
W_03	Zna i rozumie technologie takie jak standard JavaServer Faces oraz Spring Web Flow wykorzystywane do tworzenia warstwy prezentacji.	K_W06, K_W12
W_04	Zna i rozumie technologie wspomagającą tworzenie aplikacji webowych Spring Boot.	K_W06, K_W12
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowe go
U_01	Potrafi zaprojektować i zaimplementować z poziomu programu dokumenty HTML i XML wykorzystując model dokumentu DOM i biblioteki takie jak JQuery czy DWR Ajax.	K_U24, K_U11, K_U22
U_02	Potrafi napisać, skompilować i uruchomić na serwerze aplikacje wykorzystujące technologie JSF, DWR, Spring Web Flow, Spring Boot.	K_U02, K_U11, K_U24, K_U22
U_03	Potrafi zaprojektować i zaimplementować aplikację webową z użyciem technologii i frameworków do tworzenia Front-endu takich jak Bootstrap, Angular.js, Thymeleaf, React.	K_U24, K_U11
U_04	Potrafi zaprojektować i utworzyć aplikację webową przy użyciu frameworka Node.js z dostępem do bazy danych zrealizowanym np. przy pomocy bazy MongoDB i użyciem standardowego formatu przesyłania obiektów JSON.	K_U24, K_U11
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowe go
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
K_02	Ma świadomość odpowiedzialności za pracę własną oraz gotowość ponoszenia odpowiedzialności za wspólnie realizowane zadania.	K_K02
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> Umiejętność programowania w języku Java, projektowania obiektowego i paradygmatu MVC. Znajomość HTML5 i podstaw języka JavaScript. 		
Treści zajęć		

1. **Skryptowy język programowania po stronie klienta JavaScript i biblioteka jQuery:** manipulowanie drzewem dokumentu DOM, zaawansowana obsługa zdarzeń, tworzenie animacji, obsługa asynchronicznych zapytań AJAX,
2. **Mechanizmy zdalnego wywoływania procedur w Javie z poziomu JavaScript:** obsługa asynchronicznych zapytań z wykorzystaniem biblioteki Direct Web Remoting
3. Technologie do tworzenia front-endowej części aplikacji webowej Angular.js, Bootstrap
4. Technologie do tworzenia aplikacji webowych – framework Node.js i baza danych MongoDB.
5. **Technologie do tworzenia aplikacji webowych: np. React, VUE Amber.**
6. **Technologie do tworzenia warstwy prezentacji w aplikacji webowej:** tworzenie komponentów interfejsu użytkownika oraz powiązań między nimi w standardzie JavaServer Faces.
7. **Przepływ informacji w aplikacji webowej:** zapoznanie z technologią Spring Web Flow
8. **Koncepty umiędzynarodawiania aplikacji:** umiędzynarodowienie i inkulturacja w języku Java, JSP, Javascript
9. **Przegląd alternatywnych rozwiązań do tworzenia systemów internetowych JHipster, SpringBoot Serwisy REST itp.**

Literatura podstawowa:

1. Rod Johnson, Juergen Hoeller, Alef Arendsen, Thomas Risberg, Colin Sampaleanu, *Spring Framework. Profesjonalne tworzenie oprogramowania w Javie*. Helion, Gliwice 06/2006.
2. Giulio Zambon, Michael Sekler, *Beginning JSP, JSF & Tomcat Web Development: From Novice to Professional*, APRESS, 2007
3. D. Mikułowski XML w programowaniu aplikacji internetowych, Oficyna Wydawnicza Akademii Podlaskiej, Siedlce 2009.

Literatura dodatkowa:

1. Direct Web Remoting <http://directwebremoting.org/dwr/index.html>
2. jQuery <http://jquery.com>

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów kształcenia osiąganych przez studenta:

Podczas ćwiczeń laboratoryjnych oraz realizacji zadania indywidualnego (projektu) sprawdzane będą efekty U_01, U_02, U_03 i U_04 oraz K_S01 i K_S-2. Podczas kolokwium zaliczeniowego sprawdzane będą efekty W_01, W_02, W_03, W_04 i W_05.

Przykładowe pytanie z kolokwium: (test wyboru):

Funkcjami dostępnymi w bibliotece jQuery są:

odp 1: click(), text(), html(), val().

odp 2: toggle(), append(), attr(), removeClass().

odp 3: draw(), addElement(), prepend(), hide().

odp 4: resizeWindow(), insert(), before(), after().

Przykładowe zadanie z laboratorium:

Utwórz stronę internetową składającą się z 3 widoków o nazwach: kontakt, więcej o mnie i moja praca wykorzystując mechanizm routingu dostępny we frameworku Angular.js.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie ocen częściowych uzyskanych na każdych zajęciach laboratoryjnych, oceny z zadania indywidualnego (projektu) i kolokwium (testu w formie elektronicznej). Za zajęcia laboratoryjne można uzyskać maksymalnie 120pkt, za zadanie indywidualne (projekt) można uzyskać maksymalnie 50 pkt a za kolokwium - maksymalnie 30 pkt. Daje to łącznie 200pkt. Zaliczenie zajęć następuje w przypadku uzyskania co najmniej 61 pkt za zadania wykonywane podczas laboratorium 25 pkt za zadanie indywidualne oraz co najmniej 16 punktów za kolokwium.

Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 200pkt.) może być następująca (w nawiasach ocena wg skali ECTS):

- 0-102 pkt. ndst (F)
- 103-121 pkt. dst (E)
- 122-141 pkt. dst+ (D)
- 142-161 pkt. db (C)
- 162-181 pkt. db+ (B)
- 182-200 pkt. Bdb (A)

Warunek uzyskania zaliczenia przedmiotu: co najwyżej dwie nieusprawiedliwione nieobecności na ćwiczeniach i spełnienie każdego z niżej opisanych warunków

1. uzyskanie co najmniej 61 punktów z ćwiczeń
2. uzyskanie co najmniej 25 punktów z zadania indywidualnego (projektu)
3. uzyskanie co najmniej 16 punktów z kolokwium(testu)

Sposób uzyskania punktów:

1. ćwiczenia laboratoryjne: 120 pkt. (za każde z 12 zajęć do 4,5 pkt.)
2. zadanie końcowe (projekt): 50 pkt
3. kolokwium / test elektroniczny: 30 pkt

Poprawy:

- Jednorazowa poprawa kolokwium (e-testu) w sesji.
- Obrony do 4 nie wykonanych ćwiczeń laboratoryjnych w trakcie semestru (za obronione zadanie można otrzymać maksymalnie 50% punktów.
- Możliwość ponownego przystąpienia do obrony zadania końcowego (projektu).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych i wykonanie zadania końcowego (projektu)	35 godz.

Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	15 godz.
Sumaryczne obciążenie pracą studenta:	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	38 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	30 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:	Technologie programistyczne - sieciowe systemy biznesowe	
Nazwa w języku angielskim:	Programming Technologies - Network Business Systems	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:	informatyka	
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):	fakultatywny	
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):	pierwszego stopnia	
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:	Dr Anna Wawrzyńczak-Szaban	
Imię i nazwisko prowadzących zajęcia:	Dr Anna Wawrzyńczak-Szaban	
Założenia i cele przedmiotu:	Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami funkcjonowania systemów sieciowych systemów biznesowych opartych o architekturę SOA	
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
W_01	Student zna i rozumie koncepcję systemów w architekturze opartej o usługi (SOA) oraz przesłanki biznesowe i technologiczne do stosowania tej architektury.	K_W10, K_W12, K_W13
W_02	Zna i rozumie zasady projektowania SOA, ich warstwy i standardy architektury.	K_W12
W_03	Zna i rozumie fazy w cyklu życia SOA oraz wzorce rozwiązań dla scenariuszy tworzenia serwisów i identyfikacji ich produktów.	K_W12
W_04	Zna i rozumie zasady bezpieczeństwa w środowisku SOA .	K_W12, K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrąfi wyspecyfikować serwis SOA, wybrać odpowiedni scenariusz i zidentyfikować wzorzec z użyciem JK Enterprise.	K_U01, K_U19
U_02	Potrąfi modelować proces z użyciem WebSphere Business Modeler, przeprowadzić symulację i przeanalizować jej wyniki.	K_U01, K_U11, K_U19
U_03	Potrąfi debuggować moduł procesu biznesowego, dodawać reguły i zadania biznesowe.	K_U01, K_U19, K_U24

U_04	Potrafi rozmieszczać i zarządzać procesem biznesowym i usługami mediacyjnymi w WebSphere Process Server	K_U01, K_U20
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
K_02	Jest gotów formułować opinie na temat problematyki architektury zorientowanej na usługi (Service-Oriented Architecture, SOA) i przekazywać informacje i opinie w sposób zrozumiały	K_K02
Forma i typy zajęć:	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
Wymagania wstępne i dodatkowe:		
Umiejętność programowania oraz znajomość technologii sieciowych i zagadnień związanych z zarządzaniem i projektowaniem systemów informatycznych.		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 1. Wprowadzenie do architektury SOA. Koncepcja systemów w architekturze opartej o usługi. Przesłanki biznesowe do stosowania architektury SOA. Przesłanki technologiczne do stosowania architektury . 2. Warstwy architektury SOA. Warstwy tworzące architekturę zorientowaną na usługi. Korzyści z wdrożenia SOA. Cykl życia SOA 3. Zasady projektowania SOA. Charakterystyka architektury zorientowanej na usługi. Zasady projektowania stosowane podczas budowy serwisów. Jak SOA jest powiązane z tradycyjnymi stylami architektonicznymi. Definiowanie SOI (service-oriented integration) i wyjaśnienie roli ESB w ustanawianiu SOI. Studium przypadku z na przykładzie JK Enterprise. 4. Standardy w architekturze SOA. Usługi sieciowe (Web services) i ich podstawowe standardy. Wyjaśnienie związków między WS a SOA. Budowanie rozwiązań interoperacyjnych. Lista standardów i identyfikacja ról w budowaniu rozwiązań z wykorzystaniem WS. Modelowanie procesu z użyciem WebSphere Business Modeler. 5. Wsparcie narzędziowe IBM dla cyklu rozwojowego opartego o SOA. Identyfikacja zmiennych i standardów stosowanych w poszczególnych fazach cyklu życiowego SOA. Faza modelowania: Rational RequisitePro , Rational Software ,WebSphere Business Modeler . Faza montażu: WebSphere Integration Deweloper, WebSphere Developer for System z, Rational Application Deweloper 6. Wsparcie narzędziowe IBM dla cyklu rozwojowego opartego o SOA. Faza Konfiguracji: WebSphere Application Serwer ,WebSphere Process Serwer, WebSphere Portal, WebSphere DataPower. Faza zarządzania : WebSphere Business Monitor and Tivoli Performance Viewer, ITCAM for WebSphere and ITCAM for SOA 7. Zarządzanie w SOA. Wsparcie dla zarządzania SOA. Lista faz w cyklu życiowym SOA. Identyfikacja oferty zarządzania cyklem życiowym w SOA 8. Założenia i modelowanie. Korzyści z modelowania w SOA. RUP dla modelowania architektury zorientowanej na usługi (RUP for SOMA). Profile UML 2.0 dla tworzenia oprogramowania usług 9. Założenia i modelowanie. Specyfikacja ról w modelowaniu biznesowym (modelowanie procesów, analiza i optymalizacja). Identyfikacja przepływów prac, aktywności i przewodników oferowanych w RUP dla SOMA (identyfikacja serwisów, ich specyfikacja i realizacja). Wsparcie narzędzi CASE dla projektowania SOA 10. Scenariusz. Definiowanie punktów wejść do tworzenia scenariuszy dla SOA. Realizacje scenariuszy szybkiego rozwoju projektów SOA 		

11. Wzorce. Wzorce rozwiązań dla scenariuszy tworzenia serwisów i identyfikacja ich produktów (do realizacji tego wzorca). Wzorce rozwiązań dla połączeń serwisów i identyfikacji ich produktów (do realizacji tego wzorca). Wzorzec scenariusza dla BPM i identyfikacja jego produktów
12. Założenia montażu. Kreowanie potrzeb dla programowania w modelu SOA. Definiowanie komponentów architektury zorientowanej na usługi (Service Component Architecture - SCA). Komponenty SCA: moduły serwisu, pośrednicy, komponenty serwisu i komponenty pośrednika
13. Założenia konfiguracji. Modele programowania klienta. Integracja platform architektonicznych. Integracja narzędzi do wytwarzania serwisów
14. Zarządzanie serwisów. Środowisko zarządzania w SOA. Odkrywanie wymagań w zakresie zarządzania. Identyfikacja oferty dostarczenia tych możliwości. Rozmieszczanie i zarządzanie procesem biznesowym i usługami mediacyjnymi w WebSphere Process Server.
15. Bezpieczeństwo serwisów. Bezpieczeństwo w środowisku SOA. Cykl życia dla budowania aspektu bezpieczeństwa serwisów. Oferta dostarczenia wymaganych cech serwisu w zakresie jego bezpieczeństwa

Literatura podstawowa:

1. Thomas Erl, "SOA Principles of Service Design", Prentice Hall Service-Oriented Computing Series, 2007
2. Thomas Erl, „Service-Oriented Architecture (SOA): Concepts, Technology, and Design”, Prentice Hall Service-Oriented Computing Series, 2005

Literatura dodatkowa:

1. Norbert Bieberstein, Robert G. Laird, Keith Jones, Tilak Mitra „Executing SOA: A Practical Guide for the Service-Oriented Architect”, IBM Press, 2008
2. Jim Brennan „IBM WebSphere DataPower SOA Appliance Handbook”, IBM Press, 2009.

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wsparcie narzędziowe IBM dla cyklu rozwojowego opartego o SOA. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty U_01 – U_4 sprawdzane będą na bieżąco, na każdym zajęciach poza pierwszym i ostatnim.

Przykładowe pytania sprawdzające efekt U_01:

- Zaproponuj specyfikację serwisu opartego na SOA
- Dobierz odpowiedni scenariusz i wzorzec dla usługi webowej wykorzystując JK Enterprise

Przykładowe pytania sprawdzające efekt U_02:

- Utwórz model procesu biznesowego np. zamówienie i dostarczenie produktu oraz przeprowadź jego symulację w WebSphere Business Modeler
- Omów wyniki symulacji procesu biznesowego

Przykładowe pytania sprawdzające efekt U_03:

- Zaproponuj i dodaj nowe reguły oraz zadania biznesowe do wskazanego procesu biznesowego

Przykładowe pytania sprawdzające efekt U_04:

- Omów usługi mediacyjne dostępne z WebSphere Process Server

Efekty W_01 – W_04 oraz K_01 i K_02 sprawdzane będą na kolokwium na ostatnim wykładzie.

Przykładowe pytania sprawdzające efekt W_01:

- Wymień i opisz charakterystyczne cechy systemów opartych na paradygmacie SOA
- Wymień i opisz przesłanki biznesowe i technologiczne do stosowania SOA

Przykładowe pytania sprawdzające efekt W_02:

- Wyjaśnij rolę i znaczenie ESB
- Wymień i opisz najpopularniejsze standardy stosowane usługach sieciowych

Przykładowe pytania sprawdzające efekt W_03:

- Scharakteryzuj profile UML 2.0 dla tworzenia oprogramowania usług

Przykładowe pytania sprawdzające efekt W_04:

- Omów aspekty związane z bezpieczeństwem usług sieciowych
- Scharakteryzuj wymagane cechy serwisu w zakresie jego bezpieczeństwa

Przykładowe pytania sprawdzające efekt K_01 i K_02:

- Wymień i opisz charakterystyczne cechy systemów opartych na paradygmacie SOA
- Scharakteryzuj przesłanki biznesowe i technologiczne do stosowania SOA

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i jednego kolokwium pisemnego przeprowadzonego na ostatnim wykładzie. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 60pkt. Zaliczenie zajęć laboratoryjnych następuje w przypadku uzyskania co najmniej 30pkt.

Za pisemne kolokwium można na nim uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu co najmniej 20 pkt. Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) 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).

Poprawy:

Uzyskanie poprawkowego zaliczenia laboratoriów oraz wykładu możliwe jest w trakcie sesji egzaminacyjnej.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność

Obciążenie studenta

Udział w wykładach

21 godz.

Udział w ćwiczeniach laboratoryjnych

24 godz.

Samodzielne przygotowanie się do ćwiczeń laboratoryjnych

35 godz.

Udział w konsultacjach godz. z przedmiotu

5 godz.

Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	43 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS