



Sylabus przedmiotu / modułu kształcenia			
Nazwa przedmiotu/modułu kształcenia:		Przedsiębiorczość indywidualna	
Nazwa w języku angielskim:		Personal entrepreneurship	
Język wykładowy:	Język polski		
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka	
Jednostka realizująca:	Wydział Nauk Społecznych		
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:	2		
Imię i nazwisko koordynatora przedmiotu:		Dr inż. Stanisław Szarek	
Imię i nazwisko prowadzących zajęcia:		Dr inż. Stanisław Szarek	
Założenia i cele przedmiotu:		<ul style="list-style-type: none"> <li>- przyswojenie przez studentów wiedzy z zakresu uwarunkowań przedsiębiorczości we współczesnym świecie,</li> <li>- kształcenie przedsiębiorczych postaw, docenienie roli innowacji w gospodarce</li> <li>- ukierunkowanie studentów do samodzielnego pogłębiania wiedzy, doskonalenia umiejętności i bycia świadomym uczestnikiem rynku i społeczeństwa</li> </ul>	
Symbol efektu	Efekty kształcenia		Symbol efektu kierunkowego
	WIEDZA Student:		
W_01	Zna zasady funkcjonowania gospodarki rynkowej, identyfikuje rolę państwa i sektora finansowego w gospodarce		K_W13
W_02	Rozpoznaje atrybuty człowieka przedsiębiorczego, zna sposób zakładania i formy prowadzenia działalności gospodarczej		K_W13, K_W03
		UMIEJĘTNOŚCI Student:	
U_01	Potrafi przedstawić ograniczenia ludzi i firm związane z ich funkcjonowaniem na rynku		K_U09, K_U07, K_U05
U_02	Wybiera najlepszą formę do prowadzenia różnych rodzajów działalności gospodarczej		K_U09, K_U15
U_03	Potrafi współdziałać w grupie, przyjmując w niej różne role		K_U09,
U_04	Potrafi sporządzić i skomercjalizować innowacyjne rozwiązania z zakresu informatyki		K_U09, K_U15

<b>KOMPETENCJE SPOŁECZNE</b>		
<b>Student:</b>		
<b>K_01</b>	Rozpoznaje przyczyny i skutki podejmowania różnych decyzji przez jednostki	K_K03
<b>K_02</b>	Wykazuje się kreatywnością i angażuje się w poszukiwanie rozwiązania problemów	K_K01, K_K02
<b>Forma i typy zajęć:</b>	Studia stacjonarne: ćwiczenia (30 godz), Studia niestacjonarne: ćwiczenia (18 godz),	
<b>Wymagania wstępne i dodatkowe:</b>		
Znajomość podstawowych pojęć z zakresu ekonomii		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li>1. Zasady funkcjonowania gospodarki rynkowej.</li> <li>2. Czynniki produkcji w gospodarce (ziemia, praca , kapitał i zarządzanie).</li> <li>3. Uwarunkowania prawne podejmowania działalności gospodarczej.</li> <li>4. Źródła finansowania działalności gospodarczej i projektów innowacyjnych.</li> <li>5. Innowacje w przedsiębiorczości.</li> <li>6. Określenie indywidualnych predyspozycji do prowadzenia działalności gospodarczej i uczestnictwa w projektach innowacyjnych: <ul style="list-style-type: none"> <li>• typ inteligencji wielorakiej,</li> <li>• określenie przydatności w zespole i w grupach projektowych (lider, myśliciel, innowator, praktyk itp.)</li> <li>• ocena zdolności przedsiębiorczych.</li> </ul> </li> <li>7. Tworzenie zespołów projektowych na podstawie przeprowadzonych testów.</li> <li>8. Przygotowanie projektu praktycznego z zakresu innowacyjnych rozwiązań informatycznych/uczestnictwo w symulacjach biznesowych.</li> </ol>		
<b>Literatura podstawowa:</b>		
<ol style="list-style-type: none"> <li>1. Markowski W., ABC small businessu, Wyd. XIII, Marcus s.c., Łódź 2010</li> <li>2. Cieślik J., Przedsiębiorczość dla ambitnych. Jak uruchomić własny biznes, Wydawnictwa Akademickie i Profesjonalne, Warszawa 2006</li> </ol>		
<b>Literatura dodatkowa:</b>		
<ol style="list-style-type: none"> <li>1. Kapusta F., Przedsiębiorczość – teoria i praktyka, Wydawnictwo Forum Naukowe, Poznań – Wrocław 2006</li> <li>2. Moczydłowska J., Pacewicz I., Przedsiębiorczość, Wyd. Oświatowe „Fosze”, Rzeszów 2007</li> <li>3. Dolna-Ciemniakowska M., A. Wesołowska, Zakładamy firmę, Wyd. Difin, Warszawa 2007</li> <li>4. Laszczak M., Kierowanie małą firmą-tajniki przedsiębiorczości, Wyd. Poltex, Warszawa 2004</li> </ol>		
<b>Planowane formy/działania/metody dydaktyczne:</b>		
Ćwiczenia z prezentacją multimedialną; przygotowanie i prezentacja projektu praktycznego		
<b>Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:</b>		
Weryfikacja efektów kształcenia w zakresie wiedzy następuje na kolokwium, a umiejętności i kompetencji społecznych w trakcie przygotowywania i prezentacji projektu praktycznego.		
<b>Forma i warunki zaliczenia:</b>		

Ćwiczenia - zaliczenie z oceną. Na zaliczenie przedmiotu składa się uczestnictwo na ćwiczeniach oraz uzyskanie co najmniej oceny dostatecznej (2,51). Wiedzę sprawdza praca pisemna. Czas pisania odpowiedzi - 60 minut. Liczba pytań zamkniętych wynosi 30. Umiejętności i kompetencje sprawdza przygotowanie i prezentacja projektu, za który można uzyskać 20 pt.; dodatkowe 10 pt można uzyskać za aktywność.

Na końcową ocenę składa się suma punktów uzyskana z prezentacji projektu innowacyjnego i wyniku testu. Uzyskać można maksymalnie 60 pt. (30 pt. test, 20 pt. projekt, 10 pt. aktywność)

Kryterium oceny dla zaliczenia:

- 25-30 pt – ocena dst,
- 31-35 pt – ocena dst plus,
- 36-40 pt – ocena db,
- 41-45 pt – ocena db plus,
- >45 pt – ocena bdb.

#### Bilans punktów ECTS\*:

##### Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w ćwiczeniach, kolokwium	30 godz.
Czytanie zadanych wybranych fragmentów literatury	5 godz.
Przygotowanie do testu	5 godz.,
Przygotowanie projektu	8 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>50 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>2 ECTS</b>

##### Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w ćwiczeniach, kolokwium	18 godz.
Czytanie zadanych wybranych fragmentów literatury	17 godz.
Przygotowanie do testu	5 godz.,
Przygotowanie projektu	8 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>50 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>2 ECTS</b>

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Bezpieczeństwo systemów komputerowych
Nazwa w języku angielskim:		Computer security
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 Piotr Świtalski
Imię i nazwisko prowadzących zajęcia:		dr Piotr Świtalski
Założenia i cele przedmiotu:		Celem przedmiotu jest zaznajomienie studentów z problematyką bezpieczeństwa systemów komputerowych. Przedstawione zostaną podstawowe pojęcia i techniki związane z bezpieczeństwem komputerowym. Przewiduje się zajęcia praktyczne z użyciem środowiska narzędziowego, podczas których studenci nabędą umiejętności posługiwania się podstawowymi aplikacjami w zakresie bezpieczeństwa. Przedmiot ma również usystematyzować wiedzę w zakresie obecnych zagrożeń i przeciwdziałania im w systemach komputerowych.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie podstawowe techniki szyfrowania danych oraz zagadnienia z zakresu tworzenia i weryfikacji podpisu cyfrowego	K_W05
W_02	Zna i rozumie mechanizmy uwierzytelniania użytkownika w systemach komputerowych	K_W05, K_W07
W_03	Ma wiedzę w zakresie ataków wymierzonych w aplikacje oraz systemy operacyjne	K_W05, K_W11

<b>W_04</b>	Zna i rozumie zasady zabezpieczania sieci komputerowych oraz wykrywania anomalii w tych sieciach	<b>K_W07</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
<b>U_01</b>	Potrafi sprawnie wyszukiwać w literaturze informacje związane z bezpieczeństwem systemów komputerowych, potrafi wyszukać informacje na temat nowych podatności wykrytych w systemach komputerowych	<b>K_U01</b>
<b>U_02</b>	Potrafi sprawnie wykorzystać narzędzia bezpieczeństwa systemów komputerowych	<b>K_U15, K_U17</b>
<b>U_03</b>	Potrafi dobrać adekwatne zabezpieczenia w stosunku do zagrożenia w systemie komputerowym	<b>K_U15, K_U17, K_U25</b>
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
Warunkiem uczestnictwa w zajęciach jest uprzednie zaliczenie następujących przedmiotów: „Architektura systemów komputerowych”, „Systemy operacyjne”, „Podstawy technologii WWW”, „Sieci komputerowe” lub znajomość literatury obowiązującej w tych przedmiotach.		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li><b>Koncepcja bezpieczeństwa komputerowego.</b> Podstawowe zasady. Właściwości i klasyfikacja koncepcji bezpieczeństwa: integralność, dostępność, poufność, niezaprzeczalność, odpowiedzialność. Podstawowe pojęcia. Model bezpieczeństwa sieciowego. Minimalne standardy bezpieczeństwa. Polityka bezpieczeństwa.</li> <li><b>Ataki w systemach komputerowych.</b> Specyfika systemów informatycznych. Klasyfikacja zagrożeń. Szkodliwe oprogramowanie. Sieci botnet. Ataki na współczesne procesory. Statystyka typowych zagrożeń. Przystępstwa ujawnione i nieujawnione. Obiekty, typy i sprawcy przestępstw. Piractwo komputerowe, sabotaż, wywiad gospodarczy, szpiegostwo, przestępstwa bankowe. Inżynieria społeczna. Phising. Inżynieria odwrotna. Organizacje przeciwdziałające przestępczości komputerowej.</li> <li><b>Szkodliwe oprogramowanie.</b> Problemy związane z nieautoryzowanym dostępem do systemów komputerowych. Infekcje systemów komputerowych. Bezpieczny system operacyjny. Podatność systemów operacyjnych. Charakterystyka szkodliwego oprogramowania: tylne drzwi, bomby logiczne, konie trojańskie, wirusy, robaki, exploity i rootkity, keyloggery. Przeciwdziałanie szkodliwemu oprogramowaniu.</li> <li><b>Klasyczne techniki szyfrowania.</b> Dziedzina kryptografii i podstawowe pojęcia. Podstawowe techniki szyfrowania: technika podstawieniowa, szyfr Cezara, szyfry mono i polialfabetyczne, szyfr Playfaira, szyfr Vigenère'a. Techniki przestawieniowe, szyfr zygzakowy, maszyny wirnikowe. Szyfrowanie symetryczne. Ataki siłowe przeprowadzane na algorytmy szyfrowania.</li> <li><b>Współczesne szyfry komputerowe.</b> Szyfry strumieniowe. Szyfr strumieniowy RC4. Szyfry blokowe. Struktura i szyfr Feistela. Standard DES, efekt lawiny. Algorytm AES. Tryby operacyjne szyfrów blokowych. Szyfrowanie asymetryczne. Algorytm RSA.</li> </ol>		

6. **Podpis cyfrowy.** Idea podpisu cyfrowego. Wymagania stawiane podpisom cyfrowym. Mechanizm uwierzytelniania komunikatów. Kody uwierzytelnienia komunikatów MAC. Podpis cyfrowy ElGamal. Standard DSS. Algorytm DSA. Kryptograficzne funkcje haszujące. Algorytm SHA-512. Paradoks urodzin.
7. **Uwierzytelnianie.** Pojęcie uwierzytelniania. Uwierzytelnianie przez hasło. Strategie wyboru haseł. Inne metody uwierzytelniania. Protokoły uwierzytelniania: protokół challenge and response. Atak „człowiek pośrodku”. Dowód z wiedzą zerową. Uwierzytelnianie dwuskładnikowe. Hasło jednorazowe. Generowanie haseł jednorazowych – protokół S/KEY.
8. **Kontrola dostępu.** Zasady kontroli dostępu. Podmiot, obiekt i prawa dostępu. Kontrola dostępu uznaniowa (DAC). Kontrola dostępu bazująca na rolach (RBAC). Kontrola dostępu bazująca na atrybutach (ABAC). Zarządzanie tożsamością. Zarządzanie uwierzytelnianiem. Zarządzanie dostępem. Platformy zaufane.
9. **Protokoły i standardy bezpieczeństwa w Internecie.** Bezpieczeństwo poczty elektronicznej. S/MIME. Protokoły SSL/TLS. Ataki na protokoły TLS. Protokół HTTPS. Nagłówki w protokole HTTP. Architektura IPsec.
10. **Zapory sieciowe (firewalle).** Model ogólny zapory sieciowej. Charakterystyka firewalli. Ograniczenia firewalli. Firewall filtrujący pakiety. Firewall filtrujący pakiety z badaniem stanu pakietu. Brama aplikacyjna, brama transmisyjna. Implementacja firewalla. Strefa zdemilitaryzowana (DMZ). Przykładowa konfiguracja firewalla z DMZ.
11. **Systemy wykrywania intruzów.** Zachowania intruzywne. Wzorce zachowań intruzów. Wykrywanie intruzów. Statystyczna analiza zachowania. Wykrywanie intruzów w oparciu o reguły. Systemy IDPS. Audyt w systemach IDPS. Pułapki (Honeypoty).
12. **Bezpieczeństwo aplikacji internetowych cz. 1.** Ataki w warstwie aplikacji. Niewłaściwa kontrola dostępu do aplikacji. Błędy kryptograficzne. Wstrzykiwanie kodu. Ataki SQL Injection. Atak XSS. Atak CSRF.
13. **Bezpieczeństwo aplikacji internetowych cz. 2.** Niebezpieczne projektowanie. Błędy w konfiguracji oprogramowania. Błędy w uwierzytelnianiu użytkownika i zarządzania jego sesją. Niewłaściwe zabezpieczenie wrażliwych danych. Awarie oprogramowania. Niewystarczające logowanie i monitorowanie aplikacji. Atak SSRF.
14. **Bezpieczeństwo systemów mobilnych.** Model izolowania procesów. Piaskownica. Uprawnienia Androida. Android Package. Manifest aplikacji i integralność pakietu. Weryfikacja manifestu. Zagrożenia i podatności w aplikacjach mobilnych. Ewolucja złośliwego oprogramowania w systemach mobilnych. Ataki na sprzęt. Bezpieczeństwo urządzeń mobilnych. Technologie zarządzania urządzeniami mobilnymi.
15. **Zapewnianie dostępności danych.** Utrzymanie ciągłości zasilania. Sposoby zapobiegania problemom zasilania, zasilacze awaryjne UPS. Ochrona danych przed utratą. Systemy macierzowe RAID. Kopie bezpieczeństwa.

#### Literatura podstawowa:

1. Stallings W., Brown L.: Bezpieczeństwo systemów informatycznych. Zasady i praktyka, Wydanie IV. Tom 1, Wyd. Helion, Gliwice, 2023.
2. Khawaja G.: Kali Linux i testy penetracyjne. Biblia. Wyd. Helion, Gliwice, 2022.

#### Literatura dodatkowa:

1. Stallings W., Brown L.: Bezpieczeństwo systemów informatycznych. Zasady i praktyka, Wydanie IV. Tom 2, Wyd. Helion, Gliwice, 2023.
2. Janca t.: Alicja i Bob. Bezpieczeństwo aplikacji w praktyce. Wyd. Helion, Gliwice, 2021.

3. Tevault D. A.: Bezpieczeństwo systemu Linux. Hardening i najnowsze techniki zabezpieczania przed cyberatakami, Wyd. Helion, Gliwice, 2024.

#### Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany jest technikami multimedialnymi. Ćwiczenia laboratoryjne – zajęcia praktyczne z wykorzystaniem wybranych narzędzi programowych. Na stronie internetowej prowadzącego zamieszczane są materiały z problemami i zadaniami laboratoryjnymi.

#### Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W\_01 do W\_04 weryfikowane będą poprzez egzamin pisemny, a także w toku weryfikacji przygotowania do kolejnych zajęć laboratoryjnych. Na egzaminie pisemnym pytania będą dotyczyły poznanych technik ataków i sposobów zabezpieczania systemów komputerowych. Egzamin będzie również obejmował treści związane z kryptografią. Przykładowe pytania:

- Przedstaw schemat działania podpisu cyfrowego.
- Na czym polega uwierzytelnianie dwuskładnikowe?
- W jaki sposób przeprowadzany jest atak DDoS?
- Na czym polega szyfrowanie symetryczne?
- Czym jest dowód z wiedzą zerową?

Przed egzaminem studenci będą mieli dostęp do przykładowych pytań na egzamin.

Efekty U\_01 do U\_03 będą sprawdzane systematycznie na zajęciach laboratoryjnych. Przykładowe zadania:

- Wygeneruj i zaimplementuj certyfikat SSL w wybranym serwerze HTTP.
- Zabezpiecz usługi systemu operacyjnego Linuks przy pomocy wybranej zapory sieciowej.

Materiały na następne laboratorium będą dostępne na dwa dni przed zajęciami.

#### Forma i warunki zaliczenia:

Ocena z przedmiotu składa się z dwóch ocen cząstkowych:

- oceny z zajęć laboratoryjnych,
- oceny z egzaminu końcowego.

Na ocenę z zajęć laboratoryjnych składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać sumarycznie 50 pkt. Zaliczenie zajęć laboratoryjnych możliwe po uzyskaniu co najmniej 51% liczby punktów z tej formy zaliczenia

Egzamin jest egzaminem pisemnym. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Można na nim uzyskać maksymalnie 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 51% liczby punktów z tej formy zaliczenia. 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).



<b>Bilans punktów ECTS:</b>	
Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Przygotowanie się do egzaminu	8 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	5 godz.
Udział w konsultacjach z przedmiotu	2 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Przygotowanie się do egzaminu	20 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	20 godz.
Udział w konsultacjach z przedmiotu	5 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

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):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski
Imię i nazwisko prowadzących zajęcia:		dr Grzegorz Terlikowski
Założenia i cele przedmiotu:		Celem przedmiotu jest zapoznanie studentów z platformami programistycznymi J2EE/Spring Framework i .NET/.NET Core.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie trójwarstwowy model aplikacji webowej; Potrafi wymienić i opisać sposoby dostępu do danych stosowane na platformie J2EE; Wie co to są persystencje i obiekty DAO. Zna najważniejsze technologie dostępu do danych platformy J2EE takie jak: JDBC, JTA, Hibernate,	K_W03, K_W11
W_02	Zna i rozumie zagadnienia związane z paradygmatem MVC, zna i rozumie funkcjonowanie poszczególnych elementów aplikacji zgodnej z tym paradygmatem; Zna i rozumie strukturę i mechanizmy działania aplikacji budowanej przy pomocy frameworka Spring takie jak wstrzykiwanie zależności	K_W03, K_W11
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi zaimplementować prostą aplikację opartą o Spring Boot. Potrafi posługiwać się w sposób efektywny środowiskiem programistycznym IntelliJ Idea Ultimate.	K_U01, K_U10, K_U11

<b>U_02</b>	Potrafi opracować i zaimplementować aplikację o strukturze zgodnej z paradygmatem MVC przy pomocy frameworka Spring MVC korzystającą z dostępu do danych utrwalanych w bazie danych.	<b>K_U10, K_U11, K_U12, K_U18</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
<b>K_01</b>	Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań projektowych z zakresu implementacji aplikacji multimedialnych	<b>K_K01, K_K03</b>
<b>K_02</b>	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu zadań projektowych z zakresu aplikacji wirtualnych i multimedialnych.	<b>K_K01</b>
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
Umiejętność wykorzystania języków: HTML(5), Java Script, Java, C#.		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li><b>Przegląd platform programistycznych.</b> Historia i architektury platform J2EE, Spring Framework. .NET i .NET Framework.</li> <li><b>Wprowadzenie do Spring Framework, Spring Boot i Spring MVC.</b> Przegląd modułów i możliwości Spring Framework/Spring Boot. Omówienie paradygmatu MVC.</li> <li><b>Spring MVC a programowanie reaktywne w Web Flux.</b> Przepływ sterowania. Kontrolery i mapowania żądań.</li> <li><b>Obsługa formularzy i walidacja danych.</b> Integracja aplikacji z Bean Validation API. Walidatory w Spring Framework. Przepływ sterowania.</li> <li><b>Spring IoC.</b> Kontener odwrócenia kontroli. Wstrzykiwanie zależności. Konfiguracje..</li> <li><b>Dostęp do danych w Spring Framework.</b> Wzorce DAO i Repozytorium. Spring JDBC, JPA, Encje, Mapowanie obiektowo-relacyjne. Obsługa transakcji.</li> <li><b>Spring Data.</b> JPQL. Hierarchia interfejsów. Sposoby tworzenia zapytań o dane. Wzorzec otwartej sesji w widoku. Zagadnienie audytu i sposoby jego rozwiązania.</li> <li><b>Bezpieczeństwo aplikacji z wykorzystaniem Spring Security.</b> Elementy architektury. Przepływ sterowania związany z uwierzytelnianiem i autoryzacją. Możliwości frameworka i możliwości integracyjne.</li> <li><b>Wprowadzenie do REST.</b> Formaty wiadomości, mapowania żądań, obsługa sytuacji wyjątkowych. Spring Data REST.</li> <li><b>Podstawy platformy .NET.</b> Narzędzia .NET/.NET Core (CLR, IL, CLS, assembly. Języki platformy .NET: C Specyfikacja CLI. Rodzaje aplikacji .NET: konsolowe, webowe.</li> <li><b>Możliwości języka C#.</b> Elementy leksykalne języka: komentarze, operatory, wbudowane typy danych, stałe, zmienne, instrukcje. Metody klasy konsoli. Właściwości konsoli. Instrukcje warunkowe. Instrukcje pętli. 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.</li> <li><b>Aplikacje .NET MVC Framework.</b> 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.</li> </ol>		
<b>Literatura podstawowa:</b>		
1. David R. Heffelfinger. Java EE 8 Application Development (ebook). 2017.		

- Craig Walls. **Spring W Akcji**. Wydanie V. Helion 2019.
- Krzysztof Żydzik, Tomasz Rak. C# 6.0 i MVC 5. Tworzenie nowoczesnych portali internetowych. Helion Gliwice 2018.

#### Literatura dodatkowa:

- Ranga Karanam. Mastering Spring 5.0. Publish Pack June 2017
- D. Mikułowski. XML w programowaniu aplikacji internetowych. Akademia Podlaska Siedlce 2009.

#### Planowane formy/działania/metody dydaktyczne:

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

#### Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W\_01 – W\_02 będą sprawdzane podczas pisemnego zaliczenia wykładu na ocenę. Część pytań może mieć charakter otwarty, a część zamknięty (test).

Przykładowe pytania o charakterze otwartym:

- Czym jest Encja? Przedstaw cykl życia encji (diagram stanów) w standardzie JPA.
- Wymień i scharakteryzuj poziomy caschowania danych w silniku Hibernate.
- Opisz krok po kroku jak jest obsługiwane żądanie HTTP przez silnik Spring MVC.

Przykładowe pytania o charakterze zamkniętym:

- Który serwer **nie jest** kontenerem serwletów?
  - GlassFish,
  - Apache Tomcat,
  - Jetty,
  - Netty.
- Domyślnym zakresem (ang. scope) Spring beanów jest...
  - Session
  - request
  - application
  - singleton.
- Środowiskiem uruchomieniowym aplikacji bazujących na .NET Framework jest...
  - CLR,
  - CL,
  - CTS,
  - CIL.

Efekt U\_01 - U\_02 oraz K1 - K2 będą systematycznie sprawdzane na zajęciach laboratoryjnych. 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.

Przykładowe zadanie:

- Opracuj encję Book i połącz ją relacją ManyToOne z encją CoverType oraz relacją ManyToMany z Category.

#### 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 według schematu:

- Regularne zajęcia – 35 pkt.,
- Obrona zadania indywidualnego – 15 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 18 pkt., obrona indywidualnego zadania – co najmniej 8 pkt. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt.

Egzamin jest egzaminem pisemnym, który może mieć formę testu elektronicznego. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 26 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:

1. Jednorazowa poprawa 2 wybranych przez studenta laboratoriów w trakcie konsultacji.
2. Jednorazowa poprawa projektu indywidualnego 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 egzaminu i obecność na egzaminie	10 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

##### Studia niestacjonarne

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

Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	25 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	13 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

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:	4		
Imię i nazwisko koordynatora przedmiotu:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni	
Imię i nazwisko prowadzących zajęcia:		dr hab. inż. Jerzy Tchórzewski, prof. uczelni, Mgr Dariusz Ruciński	
Założenia i cele przedmiotu:		Celem zajęć jest zapoznanie studentów ze sztuczną inteligencją, w tym zwłaszcza podstawowymi metodami sztucznej inteligencji: systemami ekspertowymi, sztucznymi sieciami neuronowymi, algorytmami ewolucyjnymi, sztucznymi systemami immunologicznymi i innymi stosowanymi do programowania, i identyfikacji systemów sztucznej inteligencji. Także celem zajęć jest poznanie przez studentów środowiska MATLAB i Simulink w tym wybranych toolbox-ów oraz j. Matlab w zakresie wspomaganie implementacji metod i systemów sztucznej inteligencji.	
Symbol efektu	Efekty uczenia się		Symbol efektu kierunkowego
	WIEDZA		
W_01	Zna i rozumie metody tworzenia bazy wiedzy, w tym w szczególności z wykorzystaniem drzewa celów oraz zna metody 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 MATLAB-a i Simulink-a.		K_W10
W_02	Zna i rozumie metody projektowania systemów ekspertowych, sztucznych sieci neuronowych oraz systemów ewolucyjnych i rozumie podstawowe zagadnienia z zakresu sztucznej inteligencji, w		K_W10

	tym 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, metody wnioskowania, itp.	
<b>W_03</b>	Zna i rozumie funkcjonowanie takich środowisk programistycznych wspomagających tworzenie systemów sztucznej inteligencji jak MATLAB i Simulink (w tym język Matlab, Simulink oraz takie toolbox-y jak m.in.: System Identification Toolbox, Deep Learning Toolbox, Fuzzy Logic Toolbox, Control System Toolbox, Optimization Toolbox, Global Optimization Toolbox, itp.) oraz rozumie zagadnienia z zakresu metodyki i technik programowania w językach bardzo wysokiego poziomu takich jak język Matlab oraz j. Python.	<b>K_W10</b>
<b>UMIEJĘTNOŚCI</b>		
<b>U_01</b>	Potrafi projektować i przeprowadzać eksperymenty badawcze systemów rzeczywistych z wykorzystaniem środowiska MATLAB i Simulink, w tym potrafi pozyskać dane do eksperymentu, przygotować eksperyment praktyczny z wykorzystaniem środków informatycznych, przeprowadzić eksperyment oraz zinterpretować uzyskane wyniki i wyciągać wnioski.	<b>K_U07</b>
<b>U_02</b>	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 programu, potrafi przygotować dane do identyfikacji i przeprowadzić identyfikację prowadzącą do uzyskania modelu systemu, potrafi przygotować plik uczący i zaprojektować Sztuczną Sieć Neuronową, potrafi przygotować populację początkową i zaprogramować Algorytm Ewolucyjny, potrafi przygotować model symulacyjny systemu rzeczywistego i zaprogramować model symulacyjny w Simulinku.	<b>K_U05</b>
<b>U_03</b>	Przy identyfikowaniu i formułowaniu specyfikacji zadań inżynierskich oraz przy ich rozwiązywaniu potrafi wykorzystywać metody analityczne, identyfikacyjne, neuronalne, ewolucyjne oraz 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, w tym: j. Matlab, Simulink oraz toolbox-y takie jak m.in.: System Identification Toolbox, Deep Learning Toolbox, Fuzzy Logic Toolbox, Control System Toolbox, Optimization Toolbox, Global Optimization Toolbox, Simulink, itp. do projektowania, testowania i symulacji złożonych systemów sztucznej inteligencji.	<b>K_U06</b>
<b>KOMPETENCJE SPOŁECZNE</b>		
<b>K_01</b>	Jest gotowy do podejmowania decyzji, krytycznej oceny działań własnych i studentów z grupy laboratoryjnej, w której uczestniczy. Jest gotowy do przyjmowania odpowiedzialności za skutki własnej pracy wykonywanej na zajęciach laboratoryjnych.	<b>K_K02</b>
<b>K_02</b>	Jest gotowy do uznania znaczenia wiedzy w rozwiązywaniu problemów oraz jest gotowy do konstruktywnej krytyki w stosunku do działań swoich i innych osób. Ma świadomość znaczenia	<b>K_K03</b>



	zachowania się w sposób profesjonalny, konieczności przejawiania inicjatywy oraz przestrzegania zasad etyki zawodowej i inżynierskiej.	
<b>K_03</b>	Jest gotowy do odpowiedzialnego pełnienia roli zawodowej i społecznej informatyka, w tym do przestrzegania zasad etyki zawodowej i etyki społecznej oraz wymagania tego od innych oraz potrafi dbać o dorobek i tradycję zawodu informatyka.	<b>K_K04</b>
<b>Forma i typy zajęć:</b>	Studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
<ol style="list-style-type: none"> <li>1. Wiedza z podstaw logiki matematycznej, rachunku zdań, rachunku predykatów, algorytmów i złożoności.</li> <li>2. Znajomość podstaw programowania deklaratywnego, w tym języka Prolog oraz programowania w językach sztucznej inteligencji, jak m.in. j. Python, j. Sphinx.</li> <li>3. Znajomość podstaw grafiki i komunikacji człowiek-komputer.</li> <li>4. Umiejętność samodzielnego programowania w dowolnych środowiskach programistycznych.</li> </ol>		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li>1. <b>Wstęp do sztucznej inteligencji.</b> 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 jako sterowanie i komunikacja wewnętrzna w systemie, informatyka a sztuczna inteligencja, metody, języki i narzędzia sztucznej inteligencji, projektowanie i implementacja systemów sztucznej inteligencji, zastosowania systemów sztucznej inteligencji.</li> <li>2. <b>Metody reprezentacji wiedzy i systemy ekspertowe.</b> Projektowanie i programowanie systemów ekspertowych. Metody akwizycji wiedzy. Języki systemów ekspertowych, inżynieria wiedzy i architektura systemów ekspertowych, Zastosowania systemów ekspertowych, systemy ekspertowe czasu rzeczywistego, metody reprezentacji wiedzy: rachunek zdań, stwierdzenia, regułowa reprezentacja wiedzy, rachunek predykatów, sieci semantyczne, reprezentacja wiedzy za pomocą ram, inne metody reprezentacji wiedzy, metody akwizycji wiedzy, moduły objaśniania wiedzy, itp.</li> <li>3. <b>Metody wnioskowania.</b> wnioskowanie w przód, sterowanie wnioskowaniem, wnioskowanie wstecz, wnioskowanie mieszane, wnioskowanie rozmyte, podstawowe zagadnienia uczenia się maszyn, metodyka tworzenia i badania poprawności bazy wiedzy, itp.</li> <li>4. <b>Sztuczne sieci neuronowe I.</b> Idea sztucznych sieci neuronowych, Charakterystyka sztucznego neuronu oraz sztucznej sieci neuronowej, modele neuronu, rodzaje sieci neuronowych, struktura sieci neuronowych, podstawowe metody uczenia sieci, reguły uczenia sieci neuronowych, metody uczenia sztucznych sieci neuronowych (uczenie z nadzorem, bez nadzoru, z krytykiem), reguły uczenia: Reguła Hebba, Reguła Perceptronowa, Reguła Delta, Reguła Widrowa – Hoffa, Reguła WTA i WTM, Reguła gwiazdy wyjść, nowoczesne metody uczenia, parametry uczenia sztucznych sieci neuronowych, itp.</li> <li>5. <b>Sztuczne sieci neuronowe II .</b> Funkcjonowanie Sztucznej Sieci Neuronowej, sieci liniowe i nieliniowe, sieci jednokierunkowe i sieci rekurencyjne, sieci komórkowe, sieci jednowarstwowe i wielowarstwowe, przykłady sztucznych sieci neuronowych: SSN PERCEPTRON, SSN HOPFIELDA, SSN HAMMINGA, SSN Kohonena, sztuczne sieci neuronowe chaotyczne, sztuczne sieci neuronowe ontogeniczne, sztuczne sieci neuronowe dualne, głębokie sztuczne sieci neuronowe, itp.</li> <li>6. <b>Sztuczne sieci neuronowe III.</b> Sztuczne sieci neuronowe w zastosowaniach praktycznych, w tym: <b>uczenie głębokie, uczenie rozmyte, pamięci asocjacyjne</b>, Metody uczenia głębokiego, metody kognitywistyczne, systemy uczące się, wnioskowanie wartości funkcji logicznej z przykładów, uczenie drzew decyzyjnych, uczenie Bayesowskie, pamięci asocjacyjne, w tym heteroasocjacyjne, itp.</li> <li>7. <b>Algorytmy genetyczne i ewolucyjne I.</b> Klasyczny algorytm genetyczny, algorytmy ewolucyjne, pojęcie krzepkości algorytmów genetycznych, populacja początkowa, operatory genetyczne krzyżowania i mutacji, metody selekcji, rola funkcji przystosowania, zarządzanie populacją, itp.</li> </ol>		

8. **Algorytmy genetyczne i ewolucyjne II.** Matematyczne podstawy algorytmów genetycznych, teoria schematów, liczebność i długość schematu, przystosowanie schematu, twierdzenie o schematach, hipoteza bloków budujących, zapobieganie przedwczesnej zbieżności, itp., strategie ewolucyjne, programowanie genetyczne i programowanie ewolucyjne, algorytmy koewolucyjne podpopulacyjne i komórkowe, genetyczne systemy uczące się, modyfikacje algorytmów ewolucyjnych, itp.
9. **Systemy kognitywne.** Inteligencja kognitywna, systemy jako układy pamiętające, język maszyny a dialog człowieka z maszyną, widzenie komputerowe, mimika a naturalne odruchy twarzy humanoidalnej, problematyka umysłu a możliwości modelowania umysłu, sens poznania, rozumienie mowy i obrazu w psychologii i neurobiologii, podejście do systemu kognitywnego w filozofii, informatyce, lingwistyce i antropologii, itp.
10. **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, itp.
11. **Sztuczne systemy transportowe,** w tym sztuczne systemy rojowe (algorytmy mrówkowe, algorytmy kolonii pszczół, algorytmy ruchu, algorytmy grawitacyjne, itp.).
12. **Metody eksploracji danych i metody analizy skupień,** w tym hierarchiczne i niehierarchiczne metody analizy skupień, miary odległości, metody i techniki eksploracji danych, techniki odkrywania asocjacji, metody klasyfikacji, metody grupowania, itp.
13. **Metody maszynowego uczenia się,** w tym uczenie ze wzmocnieniem, wnioskowanie wartości funkcji logicznej z przykładów, uczenie drzew decyzyjnych, uczenie Bayesowskie, uczenie z przykładów, uczenie się zbioru reguł, analityczne uczenie, połączenie indukcyjnego i analitycznego uczenia, uczenie przez wzmacnianie, itp.
14. **Inspiracje kwantowe metod sztucznej inteligencji,** w tym inspiracje kwantowe sztucznych sieci neuronowych i algorytmów ewolucyjnych, kwantyzacja i dekwantyzacja, obliczenia kwantowe z wykorzystaniem algebry liniowej i rachunku wektorowo-macierzowego, bramki i obwody kwantowe, itp.
15. **Nowe metody sztucznej inteligencji.** Fabryki bezludne, sztuczne życie, statystyczne systemy uczące się, rozproszone oraz równoległe i rozmyte systemy sztucznej inteligencji, itp.

#### Literatura podstawowa:

1. M. Flasiński: Wstęp do sztucznej inteligencji. PWN. Warszawa 2018.
2. R. Prata: MATLAB dla naukowców i inżynierów. PWN. Warszawa 2015.
3. L. Rutkowski: Metody i techniki sztucznej inteligencji. PWN. Wyd. II zmienione. Warszawa 2020.
4. J. Tchórzewski: Sztuczna inteligencja i informatyka kwantowa. Wydawnictwo UPH, Siedlce 2021.
5. R. Tadeusiewicz: Archipelag sztucznej inteligencji. OW EXIT, Warszawa 2021.

#### Literatura dodatkowa:

1. R. A. Kosiński: Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos. PWN. Warszawa 2020.
2. M. Lawrence: Sztuczna inteligencja i uczenie maszynowe dla programistów. Praktyczny przewodnik po sztucznej inteligencji. HELION, Warszawa 2021.
3. S. Osowski: Cyfrowe przetwarzanie sygnałów z zastosowaniem MATLABA. OW PW. Warszawa 2016.
4. J. Patterson, A. Gipson: Deep Learning. Praktyczne wprowadzenie. Helion. Gliwice 2018.
5. H. de Ponteves: Sztuczna inteligencja. HELION, Warszawa 2021.
6. M. Sawerwain, J. Wiśniewska: Informatyka kwantowa. Wybrane obwody i algorytmy. PWN. Warszawa 2015.
7. M. Szeliga: Praktyczne uczenie maszynowe. PWN, Warszawa 2019.
8. P. Wawrzyński: Podstawy sztucznej inteligencji. OW PW, Warszawa 2019.
9. S. Wierzchoń, M. Kłopotek M: Algorytmy analizy skupień. WNT. Warszawa 2015.

#### Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi. Udostępnianie studentom treści wykładów w postaci Print Screen-ów prezentacji przygotowanych w MS Power Point oraz instrukcji do ćwiczeń laboratoryjnych przygotowanych w wersji pdf. Ćwiczenia laboratoryjne realizowane są w 5-ciu blokach tematycznych. Na każdy blok tematyczny składają się trzy ćwiczenia laboratoryjne:

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

Dodatkowym zadaniem jest opracowanie projektu indywidualnego poza zajęciami i jego implementacja w wybranym środowisku implementacji systemów sztucznej inteligencji (m.in. w j. Matlab, j. Python, w j. Sphinx, itp. na uzgodniony z prowadzącym zajęcia temat.

#### **Treści bloków zadań laboratoryjnych:**

**Zadanie 1:** Opracowanie eksperymentu badawczego, pozyskanie danych do badań oraz przeprowadzenie identyfikacji systemu w środowisku MATLABA i Simulinka z wykorzystaniem System Identification Toolbox-a i Control System Toolbox-a e celu uzyskania modelu parametrycznego systemu z elementami sztucznej inteligencji.

**Zadanie 2:** Przygotowanie pliku uczącego oraz zaprojektowanie Sztucznej Sieci Neuronowej w środowisku MATLAB-a i Simulink-a z wykorzystaniem Deep Learning Toolbox-a lub Neural Network Toolbox-a oraz Fuzzy Logic Toolbox-a w celu uzyskania modelu neuralnego systemu.

**Zadanie 3:** Zaprojektowanie Algorytmu Ewolucyjnego na bazie uzyskanego modelu identyfikacyjnego lub modelu neuralnego (utworzenie Populacji Początkowej, opracowanie uzgodnionego z prowadzącym zajęcia algorytmu krzyżowania, algorytmu mutacji, algorytmu selekcji, algorytmu funkcji krzepkości/przystosowania, itp.) oraz implementacja Algorytmu Ewolucyjnego w ustalonym z prowadzącym zajęcia języku programowania systemów sztucznej inteligencji, m.in. w j. Matlab w postaci m-pliku lub z wykorzystaniem System Optimization Toolbox-a (lub Global Optimization Toolbox-a), języku Python i jego bibliotek, języku Sphinx, itp., a także przetestowanie zaimplementowanego Algorytmu Ewolucyjnego.

**Zadanie 4:** Zaprojektowanie i implementacja w Simulink-u modelu symulacyjnego z elementami systemu sztucznej inteligencji na bazie uzyskanych modeli: parametrycznego, neuralnego, ewolucyjnego, itp. oraz wykorzystanie ich do przeprowadzenia badań symulacyjnych, komparatystycznych i testujących poprawność uzyskanych modeli, a m.in. do ich porównywania między sobą oraz w odniesieniu do systemu rzeczywistego z wykorzystaniem błędu bezwzględnego i błędu względnego.

**Zadanie 5:** Każdy student otrzymuje do samodzielnego poznania środowisko programistyczne z zakresu sztucznej inteligencji i robotyki humanoidalnej lub mobilnej. Samodzielnie opracowuje przykład, implementuje go z wykorzystaniem uzgodnionego z prowadzącym zajęcia środowiska programowania oraz przygotowuje sprawozdanie z przeprowadzonych badań, a także instrukcję obsługi programu. Możliwymi do wykorzystania są m.in.:

- j. Matlab z takimi bibliotekami jak m.in.: Bioinformatics Toolbox, Control System Toolbox, Deep Learning Toolbox, Fuzzy Logic Toolbox, Global Optimization Toolbox, Image Processing Toolbox, Mapping Toolbox, Optimization Toolbox, Signal Processing Toolbox, Statistics and Machine Learning Toolbox, Symbolic Math Toolbox, Wavelet Toolbox, itp.
- j. Python z dostępnymi bibliotekami,
- j. Sphinx z dostępnymi bibliotekami.

Wszyscy studenci samodzielnie poznają podstawy implementacji systemów z wykorzystaniem jednego z ww. języków programowania i ich bibliotek.

#### **Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:**

Efekty **K\_W01 - K\_W03**, sprawdzane są na czterech zajęciach podsumowujących bloki tematyczne na zajęciach laboratoryjnych sprawdzającym wiedzę praktyczną w odniesieniu do wiedzy teoretycznej.

Pytania dotyczyć będą praktycznego użycia wybranych metod sztucznej inteligencji oraz związanych z nimi zadań szczegółowych.

#### **Przykładowe zadania:**

- Dana jest struktura pliku uczącego. Należy zaprojektować Sztuczną Sieć Neuronową w środowisku MATLABA z wykorzystaniem Deep Learning 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 oraz napisać program w: j. Matlab w postaci m-pliku, j. Python, j. Sphinx, itp. Z implementacją Algorytmu Ewolucyjnego zawierającego jednopunktowe krzyżowanie, mutację poprzez zmianę znaku genu oraz selekcję koła ruletki.
- Dany jest plik uczący, struktura systemu rzeczywistego oraz reguła uczenia. Należy przygotować i przeprowadzić obliczenia uczenia Sztucznej Sieci Neuronowej jako modelu systemu rzeczywistego oraz przedstawić uzyskany model neuronalny złożony z odpowiednich sumatorów i układów odwzorowujących.
- Dany jest model matematyczny systemu, np. w postaci modelu parametrycznego arx. Należy zbudować model symulacyjny w Simulinku. Zaproponować sposób na przeprowadzenie badań symulacyjnych, komparatystycznych oraz badania wrażliwości.

Sprawdzenie wiedzy teoretycznej dokona się jednokrotnie na ostatnim wykładzie w postaci testu z jednokrotnym wyborem lub w postaci zadań problemowych.

Efekty **K\_U01, K\_U02, K\_U03** sprawdzane są cztery razy, to jest przy zaliczaniu każdego zadania ćwiczeń laboratoryjnych.

Efekty **K\_K01, K\_K02** będą weryfikowane, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, a także podczas zaliczania tematów laboratoriów oraz zadania indywidualnego.

#### **Forma i warunki zaliczenia:**

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie ocen z pięciu zadań laboratoryjnych wykonanych samodzielnie przez studentów na zajęciach laboratoryjnych (dopuszcza się realizację zadanie 5, tj. projektu w zespołach dwuosobowych). Z całości zajęć (wykłady oraz laboratoria) przewidywany jest egzamin pisemny sprawdzający wiedzę teoretyczną i praktyczną w postaci sprawdzianu lub alternatywnie zadań problemowych. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim, w tym projektu wykonanego samodzielnie (zadanie 5), według schematu:

- regularne zajęcia (5 tematów x 20 pkt) – 100 pkt, przy czym każde zadanie oceniane jest w zakresie: wiedzy praktycznej (wykonanie ćwiczenia laboratoryjnego) – do 10 pkt., obrony wykonanego zadania i związanej z nim wiedzy teoretycznej – do 5 pkt. oraz sprawozdania z wykonanego zadania (zadania 1-4) oraz z przygotowanej instrukcji korzystania z odpowiedniego toolbox-a (zadanie 5) – do 5 pkt.,

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania ze wszystkich form aktywności studenta na zajęciach laboratoryjnych co najmniej 51 pkt. Na tej formie zajęć student może maksymalnie uzyskać do 100 pkt.

Za egzamin pisemny w postaci testu lub zadań problemowych można uzyskać do 100 pkt. Zaliczenie egzaminu jest możliwe po uzyskaniu co najmniej 51 pkt. Każdy student może uzyskać dodatkowe punkty m.in. za aktywność na zajęciach laboratoryjnych i na wykładach (w części podsumowującej wykłady). Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), jest średnią oceną z laboratorium oraz z egzaminu, przy czym zarówno z ćwiczeń laboratoryjnych jak też z egzaminu, w zależności od sumy uzyskanych punktów (suma ze względu na dodatkową ocenę aktywności studentów może przekroczyć 100 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).

W przypadku przekroczenia przez studenta wymaganej liczby maksymalnej 100 pkt. student może uzyskać od koordynatora przedmiotu dyplom wyróżnienia z metod sztucznej inteligencji.

**Poprawy:** Istnieje możliwość jednorazowej poprawy egzaminu (przeprowadzonego w postaci testu sprawdzającego wiedzę teoretyczną oraz wiedzę praktyczną lub w postaci zadań problemowych) w trakcie trwania sesji egzaminacyjnej. Ćwiczenia laboratoryjne praktyczne w każdym bloku tematycznym można jednokrotnie dodatkowo zaliczać w trybie poprawkowym na konsultacjach w trakcie semestru.

### Bilans punktów ECTS:

#### Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych, w tym przygotowanie danych liczbowych do eksperymentów praktycznych, zaprojektowanie/zaprogramowanie i wykonanie eksperymentów praktycznych, sprawdzenie poprawności uzyskanych wyników, zaliczenie poszczególnych zadań i projektu indywidualnego	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych – zapoznanie się z instrukcją laboratoryjną oraz wykonanie sprawozdań z laboratoriów (4 bloki tematyczne x 2 godz.), a także wykonanie projektu indywidualnego ze sprawozdaniem/instrukcją obsługi (1 blok tematyczny x 7 godz.)	15 godz.
Udział w konsultacjach z przedmiotu oraz w studiach samodzielnych, w tym webinarach obejmujących zagadnienia z wykładów	5 godz.
Przygotowanie się do egzaminu z wiedzy teoretycznej (10 godz.) i wiedzy praktycznej (10 godz.)	20 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

#### Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych – zapoznanie się z instrukcją laboratoryjną, przygotowanie danych liczbowych do	30 godz.

eksperymentów praktycznych oraz sprawozdania z zajęć (4 bloki tematyczne x 5 godz.), a także wykonanie projektu indywidualnego ze sprawozdaniem i instrukcja obsługi (1 blok x 10 godz.)	
Udział w konsultacjach z przedmiotu oraz w studiach samodzielnych, w tym webinarium obejmujących zagadnienia z wykładów	10 godz.
Przygotowanie się do egzaminu z wiedzy teoretycznej (15 godz.) i wiedzy praktycznej (15 godz.)	30 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

Sylabus przedmiotu / modułu kształcenia			
Nazwa przedmiotu/modułu kształcenia:		Programowanie w sieciach komputerowych	
Nazwa w języku angielskim:		Programming in Computer 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 hab. Stanisław Ambroszkiewicz Prof.uczelni	
Imię i nazwisko prowadzących zajęcia:		S. Ambroszkiewicz dr Grzegorz Terlikowski	
Założenia i cele przedmiotu:		Celem modułu jest zaznajomienie studentów z ws[zaawansowanymi protokołami i technologiami sieci komputerowych, oraz wprowadzeniem studentów do programowania z użyciem tych protokołów	
Symbol efektu	Efekty uczenia się		Symbol efektu kierunkowego
	WIEDZA		
W_01	Student zna i rozumie zasadę działania współczesnych zaawansowanych sieciowych protokołów komunikacyjnych		K_W07
W_02	Zna i rozumie zasady działania architektury CNAApp		K_W07
W_03	Zna i rozumie zasadę projektowania i działania sieci SDN.		K_W07
W_04	Zna i rozumie wybrane aspekty bezpieczeństwa w sieciach komputerowych.		K_W07
W_05	Zna i rozumie wybrane elementy programowania sieciowego (stosowane w CNAApp oraz service Mesh) w oparciu o język Java.		K_W07
UMIEJĘTNOŚCI			
U_01	Potrafi projektować realizować aplikacje sieciowe CNAApps na mikroserwisach w architekturze chmury obliczeniowej		K_U10, K_U11, K_U12, K_U19
U_02	Potrafi projektować i zrealizować elastyczność i niezawodność CNAApp		K_U10, K_U21
U_03	Potrafi zaimplementować Service Mesh do zarządzania CNAApp		K_U10, K_U21
U_04	Potrafi zaprojektować i zaimplementować bezpieczną komunikację pomiędzy użytkownikami a API Gateway na podstawie protokołu PGP		K_U10, K_U23
KOMPETENCJE SPOŁECZNE			
K_01	Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu problemów w istniejących sieciach komputerowych uwzględniając istniejące standardy sieciowe		K_K01

<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
<b>Wymagania wstępne i dodatkowe:</b>	
<ol style="list-style-type: none"> <li>Umiejętność programowania w języku obiektowym (Java).</li> <li>Znajomość podstawowych pojęć technologii sieciowych (zakres przedmiotu Technologie Sieciowe).</li> </ol>	
<b>Treści modułu kształcenia:</b>	
<p style="text-align: center;">Treści wykładu:</p> <ol style="list-style-type: none"> <li><b>Współczesna architektura aplikacji sieciowej oparta na mikroserwisach w Chmurach obliczeniowych (CNApp)</b></li> <li><b>PGP – odpis cyfrowy i AES</b></li> <li><b>Bezpieczeństwo o CNApp oparte na PGP: protokół rejestracji i logowania</b></li> <li><b>Data plane - control plane</b></li> <li><b>MPLS i uogólniony forwarding: flow tables</b></li> <li><b>SDN oraz MFV</b></li> <li><b>Protokół OpenFlow</b></li> <li><b>Data centers networking</b></li> <li><b>Service Mesh ogólnie:</b> architektura i podstawowa funkcjonalność.</li> <li><b>Przegląd istniejących open source platform dla service mesh:</b> Linkerd, Istio, Cilium, Consul, Kuma</li> <li><b>Service Mesh:</b> skalowanie CNApp czyli równoważenie obciążenia i zarządzanie wydajnością oraz zasobami</li> <li><b>Service Mesh:</b> odporność na awarie, czyli rekonfiguracja i odtwarzanie mikroserwisów, które uległy awarii</li> <li><b>Protokół SSMMP</b> (<a href="https://github.com/sambrosz/SSMMP-a-simple-protocol-for-Service-Mesh-management/tree/main">https://github.com/sambrosz/SSMMP-a-simple-protocol-for-Service-Mesh-management/tree/main</a>)</li> <li><b>ELONET - Starlink</b></li> <li><b>Sieci optyczne</b></li> <li><b>Aplikacje multimedialne</b></li> <li><b>Zarządzanie sieciami, część 1.</b> Wprowadzenie do zarządzania sieciami, motywacje, główne komponenty. Infrastruktura do zarządzania sieciami (internetowymi). MIB: management information base, SMI: Structure of Management Information - język do definiowania struktur danych.</li> <li><b>Zarządzanie sieciami, część 2.</b> SNMP: protokół do zarządzania sieciami. Bezpieczeństwo i administrowanie sieciami. Zdalny nadzór sieci – RMON.</li> <li><b>Omówienie najnowszych technologii sieciowych, SDN i 5G.</b> Przegląd nowych rozwiązań i technologii w sieciach komputerowych. Zakreślenie trendów i perspektyw rozwoju nowych technologii w sieciach komputerowych i aplikacjach sieciowych.</li> </ol> <p style="text-align: center;">Treści zajęć laboratoryjnych</p> <ol style="list-style-type: none"> <li><b>Realizacja prostej aplikacji CNApp</b></li> <li><b>Przesyłanie obiektów w Javie.</b> Serializacja obiektów - interfejs Serializable. Wykrywanie typów w czasie wykonania - RTTI (run-time type identification).</li> <li><b>Programowanie na PGP - cz. 1.</b> Generowanie kluczy RSA. Realizacja prostej aplikacji klient serwer z wykorzystaniem bibliotek Java do uwierzytelniania.</li> <li><b>Programowanie na PGP - cz. 2.</b> Klucz symetryczny AES, szfrowanie danych, a następnie połączenie z uwierzytelnianiem.</li> <li><b>Implementacja protokołu rejestracji i logowania w CNApp opartego na PGP</b></li> <li><b>Prosta implementacja skalowania CNApp</b></li> </ol>	



7. Prosta implementacja odporności CNApp na awarie
8. Założenia do implementacji protokołu SSMMP
9. SSMMP: Service API
10. SSMMP: Agent API
11. SSMMP: Menadzer API
12. Implementacja całości SSMMP mna prostej Aplikacji testowej CNApp
13. Oddanie i zaliczenie projektu z programowania sieciowego: implementacja prostej CNApp w oparciu o service mesh.

#### Literatura podstawowa:

1. James Kurose, Keith Ross - Computer Networking\_ A Top-Down Approach, 7th Edition (2017) and 8<sup>th</sup> Edition (2022)
2. Kurose J. F., Ross K. W., Sieci komputerowe. Od ogółu do szczegółu z Internetem w tle, Wydawnictwo Helion, Gliwice 2006
3. Tom Laszewski, Kamal Arora, Erik Farr, Piyum Zonooz. Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud. 1st ed. Packt Publishing, 2018. Web. 14 Oct. 2022.
4. Jim Doherty. SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization. Copyright © 2016 Pearson Education, Inc. ISBN-13: 978-0-13-430640-7, ISBN-10: 0-13-430640-6
5. George Miranda. The Service Mesh. Publisher(s): O'Reilly Media, Inc. August 2018. ISBN: 9781492031314
6. Guo Deke. Data Center Networking: Network Topologies and Traffic Management in Large-Scale Data Centers, 1st ed. 2022 Edition. Wydawca Springer Nature. ISBN 9789811693700.

#### Literatura dodatkowa:

1. K. Krysiak. Sieci Komputerowe - Kompendium. Wydawnictwo Helion 2005
2. T. Sheldon. Wielka Encyklopedia Sieci Komputerowych. Wydawnictwo Robomatic s.c. 1999. Akademia Sieci Cisco. CCNA Exploration, Semestr 1. PWN, Warszawa 2011
3. Leinwand, B. Pinsky. Konfiguracja Routerów Cisco. Podstawy. Mikom, Warszawa 2002.
4. M. Sportack. Routing IP - podstawowy podręcznik. Mikom, Warszawa 2000.
5. M. Sportack. Sieci komputerowe. Wydawnictwo Helion 2004,
6. R. Wright. Elementarz routingu IP. Mikom, Warszawa 1999.

#### Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, Laboratoria z wykorzystaniem sprzętu sieciowego. Zamieszczanie na stronach internetowych zadań i materiałów do ćwiczeń.

#### Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekt U\_02 jest sprawdzany przy obronie programistycznego zadania.

Efekty U\_01 – U\_04, są sprawdzane w czasie ocenianych zadań na laboratoriach, niektóre zadania wykonywane są w grupach i w ten sposób sprawdzane jest efekt K\_02.

Efekty U\_02, W\_02 – W\_05 i K\_01, sprawdzane są egzaminie.

#### Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny częściowe 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. 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 – 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**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	34 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	15 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

**Studia niestacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	47 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>		<b>Komunikacja i przetwarzanie w chmurze</b>
<b>Nazwa w języku angielskim:</b>		Cloud communications and computing
<b>Język wykładowy:</b>	Polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>		Informatyka
<b>Jednostka realizująca:</b>		Wydział Nauk Ścisłych i Przyrodniczych
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>		fakultatywny
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>		pierwszego stopnia
<b>Rok studiów:</b>	III	
<b>Semestr:</b>	5	
<b>Liczba punktów ECTS:</b>	4	
<b>Imię i nazwisko koordynatora przedmiotu:</b>		Dr Dariusz Mikułowski
<b>Imię i nazwisko prowadzących zajęcia:</b>		Mgr Wojciech Nabiałek
<b>Założenia i cele przedmiotu:</b>		<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 używać chmury w celu składowania i zarządzania różnego rodzaju danymi.</p>
<b>Symbol efektu</b>	<b>Efekt uczenia się: WIEDZA</b>	<b>Symbol efektu kierunkowego</b>
<b>W_01</b>	Zna i rozumie zasady dotyczące komunikacji i przetwarzania w chmurze, zna działanie modeli dostępu do usług w chmurze ich rodzaje, (IAAS, PAAS, SAAS, IAAAS).	<b>K_W06</b>
<b>W_02</b>	Zna i rozumie sposoby przechowywania różnego rodzaju danych w chmurze.	<b>K_W06</b>

<b>W_03</b>	Zna i rozumie funkcjonowanie narzędzi i metodyk związanych z projektowaniem i programowaniem aplikacji w chmurze i migrowania istniejących aplikacji do chmury.	<b>K_W06</b>
<b>W_04</b>	Zna i rozumie sposoby komunikacji i skalowalności w chmurze.	<b>K_W06</b>
<b>W_05</b>	Zna i rozumie kwestie bezpieczeństwa związane z programowaniem aplikacji w chmurze.	<b>K_W06</b>
<b>W_06</b>	Zna i rozumie sposoby tworzenia, debugowania, monitorowania i zarządzania aplikacjami w chmurze.	<b>K_W07, K_W12</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
<b>U_01</b>	Potrafi programistycznie wykorzystać różne sposoby korzystania z chmury do przechowywania danych.	<b>K_U02</b>
<b>U_02</b>	Potrafi dostosować aplikacje do możliwości komunikacyjnych i skalowania w chmurze.	<b>K_U02, K_U20</b>
<b>U_03</b>	Potrafi zarządzać, debugować i monitorować aplikacjami w chmurze.	<b>K_U11, K_U12</b>
<b>U_04</b>	Potrafi migrować lokalnie tworzone aplikacje do chmury.	<b>K_U21</b>
<b>U_05</b>	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.	<b>K_U10</b>
<b>U_06</b>	Potrafi zaprojektować, zaimplementować oraz przetestować proste aplikacje w chmurze.	<b>K_U01, K_U02</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
<b>K_01</b>	Jest gotów do krytycznej oceny posiadanej wiedzy związanej z korzystaniem z infrastruktury dostępnych w chmurze.	<b>K_K01</b>
<b>K_02</b>	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów projektowych i programistycznych oraz do konstruktywnej krytyki powstałych rozwiązań.	<b>K_K01</b>
<b>Forma i typy zajęć:</b>	Wykład tradycyjny wspierany technikami multimedialnymi, ćwiczenia laboratoryjne na stanowiskach komputerowych, publikowanie materiałów dydaktycznych na stronach internetowych.	
<b>Wymagania wstępne i dodatkowe:</b>		
1. Dobra znajomość programowania w językach obiektowych		

2. Podstawowa praktyczna znajomość środowisk programistycznych Visual Studio oraz IntelliJ Idea,

### Treści modułu kształcenia:

1. Wprowadzenie do komunikacji i przetwarzania w chmurze: Modele usług chmurowych, PAS, IAAS, SAAS, główni dostawcy usług w chmurze, fizyczne rozmieszczenie centrów (regiony i pary regionów),
2. Wprowadzenie do Microsoft Azure: dostępne usługi na platformie Azure, maszyny wirtualne i ich konfiguracja, tworzenie sieci i podsieci dla maszyn wirtualnych Azure, bezpieczeństwo aplikacji w chmurze Azure.
3. Azure Storage – Tabele: przechowywanie różnego rodzaju danych - strukturalnych, pół-strukturalnych i niestukturalnych na platformie Azure, dostępne usługi przechowywania danych, blooby kontra object storage, usługi kopii zapasowych i kolejek komunikatów.
4. Aplikacja w chmurze: Migracja aplikacji do chmury, koncepcja i architektura mikroserwisów, zarządzanie usługami.
5. Infrastructure as a code: opisywanie usług w formacie JSON, zarządzanie usługami za pomocą skryptów: pojęcie konteneryzacji (docker composer), zarządzanie usługami za pomocą Kubernetes.
6. Komunikacja i skalowanie w chmurze: konfigurowanie dostępu sieciowego dla maszyn wirtualnych Azure, tworzenie sieci i podsieci wirtualnych, kolejgowanie komunikatów, optymalizacja komunikacji na platformie Azure.
7. Obliczenia w chmurze Azure: programowanie bezserwerowe (serverless), usługi uczenia maszynowego na platformie Azure, Azure functions.
8. Zarządzanie, debugowanie i monitorowanie aplikacji w Microsoft Azure, monitorowanie wykorzystania zasobów, bezpieczeństwo przetwarzania danych.
9. Wprowadzenie do platformy amazon Web services: kategorie usług w AWS, migracja aplikacji do AWS, różne perspektywy wdrażania aplikacji w chmurze AWS, organizacje AWS, obliczenia w AWS, Infrastructure as a code, modele płatności w AWS.
10. Magazyny danych na platformie aws: przechowywanie i zarządzanie danymi.
11. Uczenie maszynowe w chmurze: usługi uczenia maszynowego na platformie Azure, uczenie maszynowe na platformie AWS.
12. Migracja aplikacji do chmury: zaplanowanie migracji, zmiana architektury aplikacji, przeprowadzenie migracji na platformie Azure, przeprowadzenie migracji na platformie AWS. komponowanie usług z różnych platform.

### Literatura podstawowa:

1. Windows Azure. Wprowadzenie do programowania w chmurze Zbigniew Fryźlewicz, Daniel Nikończuk. Wydawnictwo Helion.
2. Kubernetes - rozwiązania chmurowe w świecie DevOps. Tworzenie, wdrażanie i skalowanie nowoczesnych aplikacji chmurowych John Arundel, Justin Domingus. Wydawnictwo Helion.
3. Amazon Web Services. Podstawy korzystania z chmury AWS Mark Wilkins. Wydawnictwo Helion.

### Literatura dodatkowa

1. Chmura Azure. Praktyczne wprowadzenie dla administratora. Implementacja, monitorowanie i zarządzanie ważnymi usługami i komponentami IaaS/PaaS Mustafa Toroman
2. Kubernetes i Docker w środowisku produkcyjnym przedsiębiorstwa. Konteneryzacja i skalowanie aplikacji oraz jej integracja z systemami korporacyjnymi Scott Surovich, Marc Boorshtein

**Planowane formy/działania/metody dydaktyczne:**

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

**Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:**

Efekty **W\_01 – W\_06** będą sprawdzane podczas ćwiczeń i na egzaminie pisemnym (egzamin może być przeprowadzony jako test online). Student będzie odpowiadał na pytania dotyczące zagadnień związanych z korzystaniem z usług w chmurze. Przykładowe pytania:

- 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.

Efekt **U\_01 - U\_06** będą systematycznie sprawdzane podczas ćwiczeń. Zadania na następne ćwiczenia udostępniane będą kilka dni wcześniej. Student, na podstawie wykładu będzie mógł 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 pisemnym, który może mieć formę testu elektronicznego zawierającego pytania otwarte i zamknięte. 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 oraz projekt końcowy, za który można uzyskać maksymalnie 40 punktów).

Ćwiczenia laboratoryjne będą zaliczone w wypadku uzyskania powyżej połowy punktów z każdego z ćwiczeń oraz powyżej połowy punktów z projektu końcowego.

Podczas egzaminu można uzyskać maksymalnie 40 punktów. Egzamin będzie zaliczony w przypadku uzyskania powyżej połowy punktów. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 190 pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 95 punktów: niedostateczna (F),
- 96 – 115 punktów: dostateczna (E),
- 116 – 130 punktów: dostateczna plus (D),
- 131 – 150 punktów: dobra (C),
- 151 – 175 punktów: dobra plus (B),
- 176 – 190 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	5 godziny
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	40 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godzin</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Udział w konsultacjach godz. z przedmiotu	5 godzina
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	55 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godzin</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

Sylabus przedmiotu / modułu kształcenia			
<b>Nazwa przedmiotu/modułu kształcenia:</b>		<b>Rozproszone Bazy Danych</b>	
<b>Nazwa w języku angielskim:</b>		Distributed Database	
<b>Język wykładowy:</b>	polski		
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>		Informatyka	
<b>Jednostka realizująca:</b>	Wydział Nauk Ścisłych i Przyrodniczych		
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>		fakultatywny	
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>		pierwszego stopnia	
<b>Rok studiów:</b>	trzeci		
<b>Semestr:</b>	piąty		
<b>Liczba punktów ECTS:</b>	4		
<b>Imię i nazwisko koordynatora przedmiotu:</b>		prof. dr hab. inż. Andrzej Barczak	
<b>Imię i nazwisko prowadzących zajęcia:</b>		prof. dr hab. inż. Andrzej Barczak, Mgr Wojciech Nabiałek	
<b>Założenia i cele przedmiotu:</b>		Celem kursu jest zapoznanie studentów z teoretycznymi zagadnieniami związanymi z rozproszonymi bazami danych, ich trendami rozwojowymi oraz modelami stosowanymi w przetwarzaniu rozproszonym	
Symbol efektu	Efekty uczenia się		Symbol efektu kierunkowego
	WIEDZA		
W_01	zna i rozumie podstawowe zagadnienia z zakresu teoretycznych podstaw rozproszonych baz danych, wykorzystywanych modeli oraz ich projektowania		K_W08
W_02	orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych z zakresu rozproszonych baz danych, zna i rozumie podstawowe zagadnienia z zakresu standardów i norm z zakresu baz danych		K_W06
W_03	Zna i rozumie zagadnienia związane z podstawowymi modelami i problemami przetwarzania rozproszonego i równoległego, zna i rozumie modele systemów rozproszonych i techniki		K_W07



<b>Sylabus przedmiotu / modułu kształcenia</b>		
	przetwarzania rozproszonego, w tym w szczególności w zakresie rozproszonych baz danych	
<b>Symbol efektu</b>	<b>UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
U_01	potrafi pozyskiwać informacje na temat rozproszonych baz danych z literatury i innych źródeł, w tym zwłaszcza internetowych; potrafi analizować, interpretować, porządkować, oceniać przydatność i użyteczność oraz agregować i integrować uzyskane informacje, a także wyciągać wnioski z ich treści i formułować opinie	K_U01
U_02	przy identyfikowaniu i formułowaniu specyfikacji zadań inżynierskich w dziedzinie baz danych oraz przy ich rozwiązywaniu potrafi dokonać wstępnej oceny proponowanych rozwiązań i podejmowanych działań inżynierskich	K_U09
U_03	potrafi zaprojektować, zaimplementować oraz przetestować system informatyczny o charakterze rozproszonej bazy danych	K_U09, K_U10, K_U12
U_04	potrafi ocenić przydatność dostępnych metod i narzędzi służących do rozwiązywania zadań inżynierskich w dziedzinie baz danych i rozwiązywać praktyczne zadania inżynierskie wymagające korzystania ze standardów i norm inżynierskich oraz stosowania technologii informatycznych, wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	K_U10, K_U24
<b>Symbol efektu</b>	<b>KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
K_01	Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań projektowych z zakresu rozproszonych baz danych	K_K01
K_02	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu zadań projektowych z zakresu rozproszonych baz danych oraz krytycznie potrafi ocenić swoje działania	K_K01
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia (24 godz.), studia niestacjonarne: wykłady (15 godz.), ćwiczenia (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		

## Sylabus przedmiotu / modułu kształcenia

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

1. Bazy danych
2. Podstawy przetwarzania równoległego
3. Platformy programowania
4. Systemy operacyjne.

lub znajomość literatury obowiązującej w tych przedmiotach.

### Treści modułu kształcenia:

1. Wprowadzenie do systemów rozproszonych. Podstawowe pojęcia związane z rozproszeniem. Rozproszenie zasobów. Główne aspekty rozproszenia baz danych. Przechowywanie rozproszenia. Współdziałanie. Heterogeniczność. *Definiowanie łącznika bazy danych.\*;*
2. Rozproszona baza danych - podstawowe pojęcia, cele i zalety rozproszenia. Komunikacja: aplikacja - baza danych, dostęp do zbioru nazw usług. Lokalny zbiór nazw usług, katalogowa. Baza danych LDAP. Serwer nazw, adresowanie serwera. Zewnętrzny serwis katalogowy. Konfigurowanie lokalnego zbioru nazw usług. Konfigurowanie środowiska klienta. *Łącznik współdzielony. Perspektywy i synonimy.*
3. Typy i reguły rozproszonych baz danych. Homogeniczność i heterogeniczność programowania. Stopień lokalnej autonomii. Federacyjny system baz danych i zarządzanie nim. Relacyjno-objektowe bazy danych. Techniki sterowania współbieżnego. Reguły Date. Fragmentacja pozioma, pionowa i mieszana. Kryteria poprawności fragmentacji. *Transakcje rozproszone cz. 1.*
4. Architektura rozproszonych baz danych. Architektura rozproszonej bazy danych. Specjalizowane oprogramowanie sieciowe. Łącznik bazy danych, perspektywa, synonim, migawka. Nazewnictwo baz danych w sieci, domena i nazwa globalna. Architektura klient - serwer, mechanizm komunikacji między klientem a serwerem. Określenie jednostki komunikacji klient-serwer, funkcje po stronie klienta i po stronie serwera. Architektura klient - broker - serwer, broker - pośrednik w dostępie do odległych zasobów. Architektura odniesienia ANSI. *Transakcje rozproszone cz. 2.*
5. Federacyjne systemy baz danych. Zarządzanie hierarchiami elementów baz danych. Protokół drzewiasty. Sterowanie współbieżnością za pomocą znaczników czasowych. Sterowanie współbieżnością za pomocą walidacji. Tryby integrowania danych. *Protokół zatwierdzania dwufazowego.*
6. Podejścia do projektowania rozproszonych baz danych. Podejście top-down, bottom-up, ad-hoc. Podział schematu logicznego, metody. Podejście oparte o globalny schemat. Problem alokacji. Problematyka replikacji. Określanie jednostki replikacji, ilości replikowanych danych, momentu i sposobu odświeżania. *Awarie i odtwarzanie transakcji rozproszonych.*
7. Komunikacja: aplikacja - baza danych. Szeregi i plany szeregowane. Szeregowalność kolizyjna. Zapewnienie atomowości rozproszonej. Zatwierdzanie dwufazowe. Odtwarzanie transakcji rozproszonych. *Migawka – perspektywa zmaterializowana.*
8. Przetwarzanie i optymalizacja zapytań rozproszonych. Transakcja rozproszona. Architektura zarządzania transakcjami rozproszonymi. Protokół 2PC, scentralizowany, zdecentralizowany i liniowy. Rodzaje optymalizacji poleceń, wybór optymalizatora i celu optymalizacji. Generowanie statystyk i algorytmy łączenia tabel.  
  
Wykonywanie zapytań rozproszonych. Filtrowanie, grupowanie i sortowanie danych z tabeli zdalnej. Łączenie tabel, wykorzystanie wskazówek w łączeniu tabel. *Typy migawek. Dziennik migawki.*
9. Replikacje. Odświeżanie replik. Migawka - perspektywa zmaterializowana, typ migawki. Dziennik migawki, definiowanie dziennika. Implementacja dziennika, fizyczne parametry składowania dziennika. Modyfikowanie i usuwanie dziennika. Grupa odświeżania. *Optymalizacja zapytań rozproszonych.*

## Sylabus przedmiotu / modułu kształcenia

10. Przetwarzanie transakcji. Szeregowalność i odtwarzalność. Szeregowalność perspektywiczna. Rozwiązywanie problemu zakleszczeń. Zatwierdzanie rozproszone. Transakcje o długim czasie trwania. *Grupy odświeżania*.
11. Partycjonowanie tabel i indeksów. Algorytmy partycjonowania danych, partycjonowanie tabel. Partycjonowanie bazujące na wartości, haszowe i hybrydowe. Fizyczne parametry składowania tabel partycjonowanych. Zarządzanie tabelami partycjonowanymi. Partycjonowanie indeksów, typy indeksów. Zarządzanie indeksami partycjonowanymi. *Partycjonowanie danych*.

\* Kursywą zaznaczono zagadnienia praktyczne realizowane w formie zajęć laboratoryjnych

### Literatura podstawowa:

1. Bębel B., Wrembel R.; Oracle. Projektowanie rozproszonych baz danych; Wydawnictwo Helion, 2003
2. Wrembel R., Bębel B., Oracle : projektowanie rozproszonych baz danych : wiedza niezbędna do projektowania oraz zarządzania rozproszonymi bazami danych Wydawnictwo Helion, 2005
3. Andrzej Barczak, Dariusz Zacharczuk, Damian Pluta., Metody optymalizacji rozproszonych baz danych w systemie Oracle w Studia Informatica. - 2015, Vol. 36, no. 1 (119), s. 113-129.
4. Andrzej Barczak, Dariusz Zacharczuk, Damian Pluta., Methods of optimization of distributed databases in oracle – part 1 w Studia Informatica: Systems and Information Technology - 2016, nr 1-2 (20), s. 5-15
5. Andrzej Barczak, Dariusz Zacharczuk, Damian Pluta., Methods of optimization of distributed databases in oracle – part 1 w Studia Informatica: Systems and Information Technology - 2017, nr 1(21),

### Literatura dodatkowa:

1. Garcia-Molina H., Ullman J. D., Widom J.; Systemy baz danych Pełny wykład; Wydawnictwa Naukowo Techniczne, 2006
2. Mueller J. P.; Poznaj Soap; Wydawnictwo Mikom, 2002
3. Di Stefano M.; Distributed Data Management For Grid Computing; John Wiley & Sons, Inc., 2005
4. Durbin J., Creekbaum W., Bobrowski S., Vasterd P.; Oracle8i Distributed Database Systems; Oracle Corporation. 1999
5. Tyagi S., McCammon K., Vorburger M., Bobzin H.; Java Data Objects; Wydawnictwo Helion, 2004
6. Connolly T., Begg C.; Systemy baz danych - Praktyczne metody projektowania, implementacji i zarządzania. Tom 1, 2, Wydawnictwo RM,
7. Elmasri R., Navathe S. B., Wprowadzenie do systemów baz danych;

### Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratoria – praktyczna praca na komputerze. Zamieszczanie na stronach internetowych zagadnień teoretycznych i zadań ćwiczeniowych.

### Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekty W\_01 - W\_03 będą weryfikowane na kolokwium. Przykładowe pytania

- Wyjaśnij podstawowe pojęcia: przezroczystość rozproszenia, współdziałanie, heterogeniczność itd.
- Omów główne aspekty rozproszenia bazy danych □ Wymień i scharakteryzuj typy rozproszonych baz danych
- Omów architekturę systemu rozproszonej bazy danych,
- Omów wady i zalety rozproszonych baz danych

## Sylabus przedmiotu / modułu kształcenia

Efekty U\_01 - U\_04 weryfikowane będą w trakcie zajęć oraz sprawdzane na kolokwium. Przykładowe zadania:

- Przeprowadź optymalizację zapytań w systemie rozproszonej bazy danych.
- Realizacja transakcji rozproszonych, problemy związane z przetwarzaniem rozproszonym,
- Przygotuj plan wykonania transakcji rozproszonej,
- Przeprowadź odświeżanie przyrostowe migawki

Efekty K\_01 i K\_02 będą weryfikowane, w oparciu o posiadaną wiedzę i umiejętności, w czasie zajęć laboratoryjnych, podczas zaliczania zadania indywidualnego, a także będą sprawdzane na kolokwium.

Przykładowe zadania:

- Zaprojektuj strukturę rozmów z potencjalnymi użytkownikami rozproszonej bazy danych w celu sformułowania podstawowych wymagań funkcjonalnych na system, Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań.

Tematy zadań indywidualnych podawane są studentom najpóźniej na 3 zajęciach laboratoryjnych. Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich przygotować samodzielnie lub korzystając z konsultacji

### 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 – 40 pkt.,
- Obrona zadania indywidualnego – 20 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 - niedostateczna (2,0) (F)
- 51 -60 - dostateczna (3,0) (E)
- 61 -70 - dostateczna plus (3,5) (D)
- 71 -80 - dobra (4,0) (C)
- 81 -90 - dobra plus (4,5) (B)
- 91 -100 - bardzo dobra (5,0).(A).

Poprawy: Dwa terminy kolokwium pisemnego. Drugi termin zaliczenia laboratorium – w toku sesji egzaminacyjnej.

### Bilans punktów ECTS\*:

#### Studia stacjonarne

#### Aktywność

#### Obciążenie studenta

<b>Sylabus przedmiotu / modułu kształcenia</b>	
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 kolokwium i obecność na kolokwium	15 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>
<b>Studia niestacjonarne</b>	
<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	47 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	20 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>			
<b>Nazwa przedmiotu/modułu kształcenia:</b>	<b>Projektowanie obiektowe</b>		
<b>Nazwa w języku angielskim:</b>	Object oriented design		
<b>Język wykładowy:</b>	Polski		
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>	Informatyka		
<b>Jednostka realizująca:</b>	Wydział Nauk Ścisłych i Przyrodniczych		
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>	Fakultatywny		
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>	Pierwszego		
<b>Rok studiów:</b>	3		
<b>Semestr:</b>	5		
<b>Liczba punktów ECTS:</b>	3		
<b>Imię i nazwisko koordynatora przedmiotu:</b>	dr Jarosław Skaruz		
<b>Imię i nazwisko prowadzących zajęcia:</b>	dr Jarosław Skaruz		
<b>Założenia i cele przedmiotu:</b>	Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z projektowaniem obiektowym tj. m.in. zasadami projektowania obiektowego, wzorcami projektowymi oraz architekturami systemów informatycznych		
<b>Symbol efektu</b>	<b>Efekty uczenia się</b>		<b>Symbol efektu kierunkowego</b>
	<b>WIEDZA</b>		
<b>W_01</b>	Zna i rozumie zagadnienia z zakresu projektowania obiektowego oraz architektur systemów informatycznych.	<b>K_W06</b>	
<b>W_02</b>	Zna i rozumie zagadnienia z zakresu wzorców projektowych i refaktoryzacji kodu.	<b>K_W06</b>	
	<b>UMIĘJĘTNOŚCI</b>		
<b>U_01</b>	Potrafi projektować program z wykorzystaniem zasad projektowania obiektowego SOLID	<b>K_U05</b>	
<b>U_02</b>	Potrafi refaktoryzować kod źródłowy zgodnie z zasadami projektowania obiektowego	<b>K_U08, K_U19</b>	
<b>U_03</b>	Potrafi projektować oprogramowanie z wykorzystaniem wzorców projektowych	<b>K_U11, K_U22</b>	
	<b>KOMPETENCJE SPOŁECZNE</b>		

K_01	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów projektowania obiektowego.	K_K01
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
1 Umiejętność programowania		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li>1. <b>Zasadny Projektowanie Obiektowego SOLID</b></li> <li>2. <b>Wprowadzenie do architektur oprogramowania.</b> Modułowość, parametry architektury, pomiar parametrów architektury i zarządzanie nimi, zakres parametrów</li> <li>3. <b>Zasady komponentów.</b> Komponenty, spójność komponentów, łączenie komponentów.</li> <li>4. <b>Style architektoniczne.</b> Architektury warstwowe, potokowe, mikrojądro, usługi, sterowanie zdarzeniami, architektury przestrzenne.</li> <li>5. <b>Reguły projektowania obiektowego.</b> Reguła jednej odpowiedzialności, reguła otwarte – zamknięte, zasada rozdzielania interfejsów, zasada odwrócenia zależności.</li> <li>6. <b>Wprowadzenie do architektury Domain Driven Design.</b> Zastosowanie modelu dziedziny, elementy składowe projektu sterowanego modelem</li> <li>7. <b>Zaawansowane zagadnienia architektury Domain Driven Design.</b> Refaktoryzacja architektury, projekt strategiczny, moduły, agregaty, fabryki, repozytoria.</li> <li>8. <b>Wprowadzenie do wzorców projektowych.</b> Wprowadzenie, wzorce strategia, obserwator, dekorator, fabryka</li> <li>9. <b>Zaawansowane wzorce projektowe.</b> Wzorce adapter, fasada, singleton, iterator, kompozyt</li> <li>10. <b>Rozszerzone wzorce projektowe.</b> Porządkowanie logiki dziedziny, stan sesji, obiekty rozproszone, wzorce logiki dziedziny, wzorce architektury źródła danych</li> <li>11. <b>Wprowadzenie do refaktoryzacji kodu.</b> Zasady refaktoryzacji, dane globalne, pętle, zduplikowany kod, testy, katalog przekształceń refaktoryzacyjnych</li> <li>12. <b>Zaawansowane techniki refaktoryzacji kodu.</b> Enkapsulacja, przenoszenie funkcjonalności, porządkowanie danych, refaktoryzacja API.</li> </ol>		
<b>Literatura podstawowa:</b>		
<ol style="list-style-type: none"> <li>1. E. Evans, Domain-Driven Design. Zapanuj nad złożonym systemem informatycznym, Helion, 2015</li> <li>2. M. Fowler, Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe., Helion, 2005</li> <li>3. R. C. Martin, Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów, Helion, 2018</li> <li>4. E. Freeman, E. Robson, Wzorce projektowe. Rusz głową! Tworzenie rozszerzalnego i łatwego w utrzymaniu oprogramowania obiektowego. Wydanie II, Helion, 2021</li> </ol>		
<b>Literatura dodatkowa:</b>		
<ol style="list-style-type: none"> <li>1. V. Vernon, DDD dla architektów oprogramowania (ebook), Helion, 2016</li> <li>2. M. Richards, N. Ford, Podstawy architektury oprogramowania dla inżynierów, Helion, 2020</li> <li>3. M. Fowler, Refaktoryzacja. Ulepszanie struktury istniejącego kodu. Wydanie II, Helion, 2019</li> </ol>		

**Planowane formy/działania/metody dydaktyczne:**

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia z wykorzystaniem narzędzi CASE i środowisk programowania. 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\_01-U\_04 będą sprawdzane na każdym zajęciach laboratoryjnych.

Efekt K1 jest sprawdzany podczas zajęć laboratoryjnych.

**Forma i warunki zaliczenia:**

Moduł kończy się zaliczeniem z oceną. 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 – 50 pkt.,
- Obrona zadania indywidualnego – 50 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 25 pkt., obrona indywidualnego zadania – co najmniej 25 pkt. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt.

Zaliczenie jest kolokwium pisemnym. Można na nim uzyskać do 100 pkt. Kolokwium będzie zaliczony w przypadku uzyskania co najmniej 50 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 200pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 100 pkt: niedostateczna (F),
- 101 – 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).

**Bilans punktów ECTS:****Studia stacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
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 egzaminu i obecność na egzaminie	10 godz
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz</b>



<b>Punkty ECTS za przedmiot</b>	<b>3</b>
<b>Studia niestacjonarne</b>	
<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godz
Udział w ćwiczeniach laboratoryjnych	15 godz
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	27 godz
Udział w konsultacjach godz. z przedmiotu	3 godz
Przygotowanie się do egzaminu i obecność na egzaminie	15 godz
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz</b>
<b>Punkty ECTS za przedmiot</b>	<b>3</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>	<b>Analiza danych</b>	
<b>Nazwa w języku angielskim:</b>	Data mining	
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>	Informatyka	
<b>Jednostka realizująca:</b>	Instytut Informatyki	
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>	fakultatywny	
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>	pierwszego stopnia	
<b>Rok studiów:</b>	trzeci	
<b>Semestr:</b>	piąty	
<b>Liczba punktów ECTS:</b>	3	
<b>Imię i nazwisko koordynatora przedmiotu:</b>	dr Anna Wawrzyńczak-Szaban	
<b>Imię i nazwisko prowadzących zajęcia:</b>	dr Anna Wawrzyńczak-Szaban	
<b>Założenia i cele przedmiotu:</b>	Celem przedmiotu jest zapoznanie studentów z pojęciami i metodami analizy oraz ich praktycznego zastosowania w wybranym środowisku obliczeniowym.	
<b>Symbol efektu</b>	<b>Efekty kształcenia</b>	<b>Symbol efektu kierunkowego</b>
	<b>WIEDZA</b>	
<b>W_01</b>	Zna i rozumie podstawowe zagadnienia związane z analizą danych.	<b>K_W01, K_W10,</b>
<b>W_02</b>	Zna i rozumie założenia poszczególnych modeli analizy danych oraz potrafi wybrać odpowiedni, zaimplementować i zastosować do konkretnego zagadnienia.	<b>K_W08, K_W10</b>
	<b>UMIEJĘTNOŚCI</b>	
<b>U_01</b>	Potrafi analizować i przygotować odpowiedni zbiór danych do wybranej metody analizy danych i ją zaimplementować.	<b>K_U01, K_U22</b>
<b>U_02</b>	Potrafi tworzyć wybrane modele analizy danych w wybranym środowisku obliczeniowym i ocenić ich efekty.	<b>K_U07, K_U12, K_U22</b>
<b>U_03</b>	Potrafi wykorzystać poznane metody do rozwiązywania problemów klasyfikacji i predykcji. Potrafi dokonać ich estymacji oraz zwizualizować i zinterpretować wyniki.	<b>K_U01, K_U20</b>
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
Umiejętność programowania, znajomość statystyki i analizy matematycznej.		
<b>Treści modułu kształcenia:</b>		

<ol style="list-style-type: none"> <li><b>1. Wstęp do analizy danych.</b> Cele stosowania. Technologie pokrewne. Etapy procesu analizy danych. Metody reprezentacji informacji.</li> <li><b>2. Wstępne przetwarzanie danych i analiza statystyczna.</b> Czyszczenie, transformacja, redukcja danych. Analiza głównych składowych. Analiza czynnikowa. Analiza regresji. Korelacja kanoniczna.</li> <li><b>3. Reprezentacja wiedzy w analizie danych.</b> Miary ciekawości. Reprezentowanie danych wejściowych i interpretacja danych wyjściowych. Techniki wizualizacji.</li> <li><b>4. Algorytmy eksploracji danych: Reguły asocjacyjne.</b> Wzorce składowe. Wydajne generowanie zestawów przedmiotów i reguł. Analiza korelacji. Generowanie reguł – algorytm A priori. Analiza koszykowa.</li> <li><b>5. Algorytmy eksploracji danych: Klasyfikacja.</b> Drzewa decyzyjne. Algorytm SVM. Kryteria oceny klasyfikatorów.</li> <li><b>6. Algorytmy eksploracji danych: Predykcja.</b> Klasyfikacja statystyczna (bayesowska). Sieci bayesowskie. Modele liniowe.</li> <li><b>7. Analiza skupień.</b> Aglomeracja. Miary odległości. Grupowanie obiektów i cech. Grupowanie metodą k-średnich. Grupowanie metodą EM.</li> </ol>
<b>Literatura podstawowa:</b>
<ol style="list-style-type: none"> <li>1. Morzy T., Eksploracja danych, Metody i algorytmy, PWN, Warszawa (2013)</li> <li>2. Andrew Bruce, Peter Bruce, Peter Gedeck, Statystyka praktyczna w data science, Helion, 2021</li> <li>3. Larose D. T., Metody i modele eksploracji danych, PWN, Warszawa (2008.)</li> </ol>
<b>Literatura dodatkowa:</b>
<ol style="list-style-type: none"> <li>1. Cichosz P., Systemy uczące się, WNT, Warszawa (2000)</li> <li>2. MathWorks, Statistics and Machine Learning Toolbox User Guide, (2016)</li> <li>3. Witten I. H., Ian H., Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Morgan Kaufmann, New York (2005)</li> </ol>
<b>Planowane formy/działania/metody dydaktyczne:</b>
Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko umożliwiające prowadzenie analizy danych np. MatLab, R, Python. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.
<b>Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:</b>
<p>Efekty W_01 – W_02 będą weryfikowane poprzez kolokwium pisemne na ostatnim wykładzie.</p> <p>Efekty U_01 -U_03 sprawdzane będą na zajęciach laboratoryjnych. Zadania na następne laboratorium muszą być dostępne tydzień przed zajęciami.</p>
<b>Forma i warunki zaliczenia:</b>
<p>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, za które można uzyskać maksymalnie 60pkt. Zaliczenie zajęć laboratoryjnych następuje w przypadku uzyskania, co najmniej 30pkt.</p> <p>Za pisemne kolokwium można na nim uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu, co najmniej 21 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):</p> <ul style="list-style-type: none"> <li>• 0 – 50 pkt: niedostateczna (F),</li> </ul>

- 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 możliwe jest w sesji egzaminacyjnej, odpowiednio przed drugim i trzecim terminem egzaminu pisemnego.

### **Bilans punktów ECTS:**

#### **Studia stacjonarne**

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

#### **Studia niestacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	27 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	15 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>		<b>Cyfrowe przetwarzanie obrazu i dźwięku</b>
<b>Nazwa w języku angielskim:</b>		Digital Image and Sound Processing
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>		informatyka
<b>Jednostka realizująca:</b>		Instytut Informatyki
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>		fakultatywny
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>		pierwszego stopnia
<b>Rok studiów:</b>	trzeci	
<b>Semestr:</b>	piąty	
<b>Liczba punktów ECTS:</b>	4	
<b>Imię i nazwisko koordynatora przedmiotu:</b>		dr Mirosław Szaban
<b>Imię i nazwisko prowadzących zajęcia:</b>		dr Mirosław Szaban
<b>Założenia i cele przedmiotu:</b>		Celem przedmiotu jest nabycie przez studentów wiedzy z zakresu przekształcania cyfrowych form danych obrazowych oraz cyfrowego przetwarzania dźwięku. Studenci zdobędą umiejętności wykorzystania algorytmów poprawy jakości obrazów, usuwania uszkodzeń form obrazowych, filtrowania danych obrazowych, wykrywania cech cyfrowego obrazu a także analizy i obróbki dźwięku.
<b>Symbol efektu</b>	<b>Efekt uczenia się: WIEDZA</b>	<b>Symbol efektu kierunkowego</b>
<b>W_01</b>	Zna podstawowe pojęcia i metody związane z pozyskiwaniem obrazów cyfrowych, ich opisem i reprezentacją.	<b>K_W01, K_W02, K_W03, K_W09</b>
<b>W_02</b>	Zna metody i podstawowe algorytmy związane z przetwarzaniem obrazów w postaci cyfrowej. Poprawą ich jakości w zakresie podstawowym i uniwersalnym, manipulacją obrazem poprzez operacje jedno i wieloargumentowe.	<b>K_W01, K_W02, K_W03, K_W09</b>
<b>W_03</b>	Zna metody i podstawowe algorytmy związane z operacjami, filtrowaniem liniowym i nieliniowym obrazów, w tym usuwaniem szumu, segmentacją i operacjami morfologicznymi.	<b>K_W01, K_W02, K_W03, K_W09</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>

U_01	Posługuje się wybranymi popularnymi aplikacjami do przetwarzania obrazu i dźwięku oraz narzędziami w nich dostępnymi. Potrafi korzystać z wybranych środowisk programistycznych i ich bibliotek pod kątem ich wykorzystania w obróbce obrazu i dźwięku.	K_U01, K_U02, K_U06, K_U07, K_U11, K_U22
U_02	Potrafi wybrać właściwy algorytm naprawy uszkodzonego obrazu i zastosować go w celu usunięcia usterek. Potrafi poprawić jakość pliku obrazu (jasność, kontrast, nasycenie barw). Potrafi usunąć wybrany rodzaj szumu, stosuje wybrane filtry poprawy jakości obrazu.	K_U01, K_U02, K_U06, K_U07, K_U11, K_U22
U_03	Potrafi pracować z plikami dźwiękowymi korzystając w wybranych aplikacjach i środowiskach programistycznych.	K_U01, K_U02, K_U06, K_U07, K_U11, K_U22
Symbol efektu	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
K_01	Jest gotów do podejmowania decyzji, krytycznej oceny działań własnych rozwiązywaniu zadań i problemów cyfrowego przetwarzania danych obrazowych i dźwięku.	K_K01, K_K02
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
1. Umiejętność podstaw programowania.		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li><b>Wstęp do przetwarzania obrazów.</b> Metody grafiki komputerowej. Formy danych obrazowych. Przekształcenia form danych obrazowych. Podstawowe przekształcenia geometryczne. Barwa, modelowanie, animacja. Zapoznanie z niezbędnym oprogramowaniem.</li> <li><b>Dyskretyzacja obrazów z gradacją kontrastu.</b> Algorytmy dyskretyzacji obrazów z gradacją kontrastu: próbkowanie, kwantowanie, rozdzielczość obrazu.</li> <li><b>Przetwarzanie obrazów z gradacją kontrastu.</b> Histogram i korekcja histogramu. Algorytmy macierzy sąsiedztwa. Algorytmy filtrowania obrazów. Filtry kierunkowe, dwuczęściowe, aproksymacji funkcyjnej.</li> <li><b>Operacje punktowe jednoargumentowe:</b> LUT, negacja, progowanie, zmiana jasności i kontrastu, korekcja Gamma, poprawa jakości z użyciem histogramu: rozciągnięcie i wyrównanie histogramu, zmiana intensywności kanałów, balans kolorów, przesunięcie kolorów.</li> <li><b>Operacje punktowe wieloargumentowe:</b> dodawanie obrazów proporcjonalne i ważone, kanał alfa, dodawanie obrazów z saturacją, odejmowanie obrazów, obrazy różnicowe, usuwanie tła, mnożenie i potęgowanie obrazów, maskowanie obrazów, dzielenie obrazów – wykrywanie ruchu.</li> <li><b>Elementy steganografii</b> – szyfrowania obrazami, ukrywanie i odczytywanie wiadomości w obrazie.</li> <li><b>Operacje kontekstowe: filtry liniowe,</b> konwolucja i splot, wygładzanie obrazu, usuwanie szumu białego i typu „Salt and Pepper”, filtr Gaussa, rozmycie kierunkowe, wyostrzanie obrazów, gradient i Laplasjan, wykrywanie linii.</li> <li><b>Operacje kontekstowe: filtry nieliniowe,</b> filtr medianowy, usuwanie szumu, wygładzanie konserwatywne, filtry wartości środkowej, uśredniające i adaptacyjne, filtr odplamiający Crimmins, inne filtry nieliniowe...</li> </ol>		

9. **Segmentacja.** Rodzaje i algorytmy segmentacji. Segmentacja dwupoziomowa i wielopoziomowa. Progowanie. Wykrywanie krawędzi, operator krzyżowy Robertsa, operator Sobela, operator kompasowy, maski Prewitta, operator Kirscha, inne metody detekcji krawędzi, filtry.
10. **Nieliniowe przetwarzanie obrazów** (morfologia matematyczna): dylatacja, erozja, operacje złożone.
11. **Tworzenie, edycja i przetwarzanie plików dźwiękowych.** Odczytywanie, modyfikacja i zapis plików dźwiękowych w wybranych formatach w środowisku Julia lub MatLab. Wykonywanie podstawowych efektów dźwiękowych. Standardy zapisu dźwięku. Podstawowe operacje na plikach dźwiękowych. Konwersje plików dźwiękowych pomiędzy wybranymi formatami.
12. **Wybrane aplikacje do obróbki sygnałów audio i narzędzia do obróbki sygnałów audio-wideo.** Przegląd wybranych aplikacji umożliwiających operacje na plikach audio-wideo. Odczytywanie, modyfikacja i zapis plików audio-wideo w wybranych formatach. **Ukrywanie obrazu w pliku dźwiękowym.**

#### Literatura podstawowa:

- a. Michał Choraś, Ryszard S. Choraś Editors, Image processing and communications challenges. 9, Advances in Intelligent Systems and Computing, 2194-5357 ; 681, Springer 2018
- b. Witold Malina, Maciej Smiatacz, Cyfrowe przetwarzanie obrazów, EXIT, 2012
- c. Anna Korzyńska, Małgorzata Przytułska, Przetwarzanie obrazów – ćwiczenia, Wydawnictwo PJWSTK, 2006

#### Literatura dodatkowa:

1. Theo Pavlidis, Grafika i przetwarzanie obrazów, WNT, 1987.
2. Z. Wróbel, R. Koprowski, Praktyka przetwarzania obrazów w programie Matlab, wyd. EXIT 2004.
3. Tomasz P. Zieliński, Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań, WKŁ, 2014

#### Planowane formy/działania/metody dydaktyczne:

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

#### Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W\_01 – W\_03 sprawdzane będą na kolokwium pisemnym w na ostatnim wykładzie, jako zagadnienia teoretyczne. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania:

- Opisz, na czym polega kwantowanie.
- W jaki sposób wyznacza się histogram obrazu? Zilustruj przykładem.
- Czym jest filtr medianowy i jakie ma zastosowanie?
- W jaki sposób przebiega detekcja krawędzi?

Efekty U\_01 – U\_03 oraz K\_01 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:

Poprawianie jakości obrazu poprzez manipulację histogramem. Korzystając z programu Julia lub MatLab, dokonać poprawy jakości obrazów z plików pomocy (krokodyl.png, morze.jpg oraz twarze.jpg), poprzez:

- rozciąganie histogramu (`imadjust('filename' )`),
- wyrównanie histogramu (`histeq('filename' )`),
- adaptacyjne wyrównanie histogramu (`adapthisteq('filename' )`).

Przykłady:

`imadjust('obraz.jpg')`, `histeq('obraz.jpg')`, `adapthisteq('obraz.jpg')`, `imadjust(f)`, `histeq(f)`, `adapthisteq(f)`, `imadjust(f(:, :, i))`, `i=1, 2, 3`, `histeq(f(:, :, i))`, `i=1, 2, 3`, `adapthisteq(f(:, :, i))`, `i=1, 2, 3`

Polecenie:

Utwórz galerię (tabelę) 2x2 (`subplot(2, 2, n);`), w której komórkach umieść obrazy po manipulacji histogramem:

- komórka 1: `krokodyl.png`, komórka 2: `krokodyl.png` + rozciągnięcie histogramu,
- komórka 3: `krokodyl.png` + wyrównanie histogramu, komórka 4: `krokodyl.png` + adaptacyjne wyrównanie histogramu,

Która metoda daje najlepszą jakość?

### Forma i warunki zaliczenia:

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

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

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

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

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

### Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.



Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie.	22 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	35 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do kolokwium i obecność na kolokwium	30 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>4 ECTS</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>		<b>Wprowadzenie do systemów kognitywnych</b>
<b>Nazwa w języku angielskim:</b>		Introduction to cognitive systems
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>		informatyka
<b>Jednostka realizująca:</b>		<b>Wydział Nauk Ścisłych i Przyrodniczych</b>
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>		fakultatywny
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>		Pierwszego stopnia
<b>Rok studiów:</b>	Trzeci	
<b>Semestr:</b>	Piąty	
<b>Liczba punktów ECTS:</b>	4	
<b>Imię i nazwisko koordynatora przedmiotu:</b>		Dr hab. Jerzy Tchórzewski, prof. uczelni
<b>Imię i nazwisko prowadzących zajęcia:</b>		Dr hab. Jerzy Tchórzewski, prof. uczelnis Mgr Dariusz Ruciński
<b>Założenia i cele przedmiotu:</b>		Celem kursu jest przekazanie studentom wiedzy na temat różnych aspektów systemów wyposażonych w technologie kognitywne, w tym ich projektowania oraz zdobycie praktycznych umiejętności ich wykorzystania w implementowanych systemach
<b>Symbol efektu</b>	<b>Efekt uczenia się: WIEDZA</b>	<b>Symbol efektu kierunkowego</b>
<b>W_01</b>	Zna założenia, podstawy teoretyczne oraz sposoby projektowania i implementacji technologii kognitywnych	<b>K_W10, K_W06</b>
<b>W_02</b>	Posiada wiedzę na temat różnych metod modelowania systemów kognitywnych, zna ich mocne i słabe strony	<b>K_W10</b>
<b>W_03</b>	Zna zastosowania technologii kognitywnych w aplikacjach użytkowych i systemach biznesowych, a także związane z nimi zagrożenia	<b>K_W03</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
<b>U_01</b>	Potrafi stosować metody i algorytmy przetwarzania danych wykorzystywane w systemach kognitywnych.	<b>K_U22</b>

<b>U_02</b>	Potrafi posługiwać się metodami i algorytmami dostępnymi w bibliotekach języka Python (np. pandas, scikit-learn, XGBoost, PyTorch) przy implementacji systemów kognitywnych	<b>K_U11</b>
<b>U_03</b>	Potrafi wykorzystać kognitywne komponenty i narzędzia chmurowe dostarczane przez Microsoft Azure	<b>K_U02</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
<b>K_01</b>	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	<b>K_K01</b>
<b>Forma i typy zajęć:</b>	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów: <ul style="list-style-type: none"> <li>• Analiza matematyczna</li> <li>• Matematyka dyskretna</li> <li>• Algorytmy i złożoność</li> <li>• Wybrane zastosowania języka Python</li> </ul> lub znajomość literatury obowiązującej w tych modułach. Student musi mieć opanowane podstawy algorytmiki, kombinatoryki i statystyki oraz posiadać umiejętność programowania w języku Python		
<b>Treści modułu kształcenia:</b>		
<ol style="list-style-type: none"> <li>1. Wstęp do kogniistyki. Mózg, umysł, zmysły, procesy poznawcze. Percepcja i jej dynamika</li> <li>2. Architektury kognitywne. Systemy kognitywne w biznesie</li> <li>3. Biblioteki języka Python i technologie chmurowe w kontekście systemów kognitywnych</li> <li>4. Teoria gier z perspektywy systemów kognitywnych. Strategie indywidualne i grupowe. Systemy wieloagentowe.</li> <li>5. Modele poznawcze bazujące na logice i wnioskowaniu. Ontologie. Podejścia Belief-Desire-Intention.</li> <li>6. Metody Bayesowskie w modelowaniu percepcji i akcji. Planowanie i podejmowanie decyzji</li> <li>7. Modele pamięci. Podobieństwa, kategoryzacja, generalizacja. Sieci neuronowe</li> <li>8. Metody uczenia maszynowego w systemach kognitywnych</li> <li>9. Przetwarzanie tekstu i języka naturalnego</li> <li>10. Pozyskiwanie i agregacja danych z sieci społecznościowych</li> </ol>		
<b>Literatura podstawowa:</b>		
<ol style="list-style-type: none"> <li>1. Laurence Moroney, Sztuczna inteligencja i uczenie maszynowe dla programistów : praktyczny przewodnik po sztucznej inteligencji, Helion, 2021</li> <li>2. Paul Deitel, Harvey Deitel, Python dla programistów : z analizami przypadków wprowadzającymi w tematykę sztucznej inteligencji, Helion, 2020.</li> <li>3. Ryan T. White, Archana Tikayat Ray, Matematyka dyskretna dla praktyków : algorytmy i uczenie maszynowe w Pythonie, Helion 2022</li> <li>4. Włodzisław Duch, Architektury kognitywne, czyli jak zbudować sztuczny umysł. UMK Toruń, 2010, <a href="https://repozytorium.umk.pl/handle/item/213">https://repozytorium.umk.pl/handle/item/213</a></li> </ol>		
<b>Literatura dodatkowa:</b>		
<ol style="list-style-type: none"> <li>1. Delip R., Natural Language Processing with PyTorch, O'Reilly, 2019</li> <li>2. Bonanno, G. Game Theory. An open access textbook with 165 solved exercises, <a href="https://arxiv.org/pdf/1512.06808.pdf">https://arxiv.org/pdf/1512.06808.pdf</a></li> </ol>		

3. Lee, M. D., & Wagenmakers, E.-J. Bayesian Cognitive Modeling: A Practical Course. Cambridge University Press, 2014	
<b>Planowane formy/działania/metody dydaktyczne:</b>	
Wykład tradycyjny wspomagany technikami multimedialnymi. Zajęcia laboratoryjne – zajęcia praktyczne z wykorzystaniem języka Python i wybranego środowiska programistycznego (np. Visual Studio, PyCharm).	
<b>Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:</b>	
<p>Efekty W_01 – W_03 będą sprawdzane na egzaminie pisemnym. Na egzaminie pisemnym zadania będą dotyczyły wybranych zagadnień teoretycznych związanych z systemami kognitywnymi. Przykładowe pytania:</p> <ul style="list-style-type: none"> <li>• Scharakteryzuj architektury kognitywne</li> <li>• Wymień metody uczenia maszynowego stosowane w systemach kognitywnych i opisz szczegółowo jedną z nich</li> <li>• Co to jest ontologia? Podaj przykład</li> </ul> <p>Efekty U_01, U_02 i U_03 sprawdzane będą sukcesywnie i oceniane po każdym laboratorium, również poprzez sprawdzanie prac domowych i kolokwium. Przykładowe zadania:</p> <ul style="list-style-type: none"> <li>• Korzystając z Azure Machine Learning opracuj i wytrenuj model predykcyjny dla podanego zestawu danych</li> <li>• Wykorzystując bibliotekę XGBoost i język Python opracuj, wytrenuj i przetestuj klasyfikator dla podanego zbioru danych</li> </ul> <p>Efekt K_01 będzie weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie wykładów i zajęć laboratoryjnych.</p>	
<b>Forma i warunki zaliczenia:</b>	
<p>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. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania powyżej 25 punktów.</p> <p>Poprawa laboratorium: najpóźniej w ciągu 3 tygodni od danych zajęć. Poprawa nie jest możliwa po zakończeniu semestru.</p> <p>Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania powyżej 25 pkt.</p> <p>Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):</p> <ul style="list-style-type: none"> <li>• 0 – 50 pkt: niedostateczna (F),</li> <li>• 51 – 60 pkt: dostateczna (E),</li> <li>• 61 – 70 pkt: dostateczna plus (D),</li> <li>• 71 – 80 pkt: dobra (C),</li> <li>• 81 – 90 pkt: dobra plus (B),</li> <li>• 91 – 100 pkt: bardzo dobra (A).</li> </ul>	
<b>Bilans punktów ECTS:</b>	
<b>Studia stacjonarne</b>	
<b>Aktywność</b>	<b>Obciążenie studenta</b>

Udział w wykładach	21 godzin
Udział w ćwiczeniach laboratoryjnych	24 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych, prace domowe, zadania	30 godzin
Udział w konsultacjach godz. z przedmiotu	6 godzin
Studia indywidualne z tematów realizowanych na wykładach	9 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godzin</b>
<b>Punkty ECTS za przedmiot</b>	<b>4</b>
<b>Studia niestacjonarne</b>	
<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych, prace domowe, zadania	30 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Studia indywidualne z tematów realizowanych na wykładach	28 godzin
Przygotowanie się do egzaminu i obecność na egzaminie	10 godzin
<b>Sumaryczne obciążenie pracą studenta</b>	<b>100 godzin</b>
<b>Punkty ECTS za przedmiot</b>	<b>4</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>		<b>Środowiska Programowania Aplikacji Wirtualnych i Multimedialnych</b>
<b>Nazwa w języku angielskim:</b>		Virtual and Multimedia Application Programming Environment
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>		informatyka
<b>Jednostka realizująca:</b>	Wydział Nauk Ścisłych i Przyrodniczych	
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>		fakultatywny
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>		pierwszego stopnia
<b>Rok studiów:</b>	trzeci	
<b>Semestr:</b>	szósty	
<b>Liczba punktów ECTS:</b>	3	
<b>Imię i nazwisko koordynatora przedmiotu:</b>		dr Grzegorz Terlikowski
<b>Imię i nazwisko prowadzących zajęcia:</b>		dr Grzegorz Terlikowski
<b>Założenia i cele przedmiotu:</b>		Celem przedmiotu jest zapoznanie się zagadnieniami związanymi z aplikacjami wirtualnymi i multimedialnymi, w szczególności z technologiami i językami programowania.
<b>Symbol efektu</b>	<b>Efekt uczenia się: WIEDZA</b>	<b>Symbol efektu kierunkowego</b>
<b>W_01</b>	Zna i rozumie na czym polega wirtualizacja oraz zna założenia dotyczące programowania aplikacji wirtualnych i multimedialnych.	<b>K_W02, K_W10</b>
<b>W_02</b>	Zna i rozumie różne atrybuty multimedii i wie jak je wykorzystać w tworzonych przez siebie aplikacjach. Rozumie potrzebę wykorzystania elementów sieciowych w aplikacjach multimedialnych.	<b>K_W02, K_W05, K_W06, K_W10</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
<b>U_01</b>	Potrafi w praktyce wykorzystać narzędzia umożliwiające tworzenie aplikacji wirtualnych i multimedialnych. Potrafi wesprzeć się adekwatnymi materiałami dydaktycznymi oraz posługiwać się pojęciami związanymi z multimediami i wirtualizacją.	<b>K_U01, K_U10, K_U11</b>

<b>U_02</b>	Potrafi implementować aplikacje multimedialne w technologiach desktopowych takich jak WPF i technologiach webowych takich jak. ElectronJS.	<b>K_U10, K_U11, K_U12</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
<b>K_01</b>	Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań projektowych z zakresu implementacji aplikacji multimedialnych	<b>K_K01, K_K03</b>
<b>K_02</b>	Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu zadań projektowych z zakresu aplikacji wirtualnych i multimedialnych.	<b>K_K01</b>
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
Umiejętność wykorzystania języków: HTML(5), Java Script, Java, C#.		
<b>Treści modułu kształcenia:</b>		
<p>1. <b>Wprowadzenie do aplikacji multimedialnych.</b> Atrybuty multimediiów, usługi multimedialne. Przegląd rozwiązań umożliwiających tworzenie bogatych, multimedialnych aplikacji internetowych.</p> <p>2. <b>Wprowadzenie do WPF.</b> Historia, architektura i cechy platformy. Cechy charakterystyczne WPF. Kontenery i pozycjonowanie. Język XAML i jego cechy. Paradygmat MVVM.</p> <p>3. <b>Zaawansowane cechy WPF.</b> Mechanizm pędzli. Animacje i ich scenariusze. Grafika 3D (układ współrzędnych 3D, rodzaje kamer, transformacje 3D).</p> <p>4. <b>Środowisko MAUI.</b> Historia, architektura. Przegląd możliwości. Obsługa podstawowych zdarzeń. Porównanie z WPF.</p> <p>5. <b>Środowisko Java FX.</b> Historia, architektura i cechy platformy. Język FXML. Efekty, transformacje, animacje i inne multimedia.</p> <p>6. <b>Środowisko ElectronJS.</b> Tworzenie aplikacji hybrydowych na pulpit. Architektura i właściwości frameworka ElectronJS. Przegląd możliwości: menu, zapis/odczyt pliku.</p> <p>7. <b>Środowisko QT.</b> Przegląd możliwości. Język QML Tworzenie interfejsów z użyciem Qt Widgets i QtQuick. Komunikacja między obiektami Qt z wykorzystaniem mechanizmu sygnałów i slotów.</p> <p>8. <b>Wprowadzenie do wirtualizacji.</b> Co to jest wirtualizacja? Podstawowe definicje. Przegląd podstawowych typów wirtualizacji - zalety, wady i zastosowanie każdego z nich.</p> <p>9. <b>Wprowadzenie do progresywnych aplikacji webowych (PWA).</b> Czym są aplikacje PWA? Zasady tworzenia aplikacji PWA. Plik manifestu, Service Workery, Cache API i praca w trybie offline.</p> <p>10. <b>Wprowadzenie do JHipster.</b> JHipster jako przykład platformy umożliwiającej szybkie tworzenie nowoczesnych multimedialnych aplikacji webowych w architekturze monolitycznej lub mikroservisów.</p>		
<b>Literatura podstawowa:</b>		
<ol style="list-style-type: none"> <li>1. J. Matulewski, MVVM i XAML w Visual Studio 2015(ebook), Helion 2016.</li> <li>2. U. Piechota, J. Piechota. JavaFX. Tworzenie graficznych interfejsów użytkownika. Helion 2021.</li> <li>3. D. Vuika. Build over 9 cross-platform desktop applications from scratch. ISBN: 978-1838552206, 2019.</li> </ol>		

4. R. McKendrick, S. Gallagher. Docker. Programowanie aplikacji dla zaawansowanych. Wydanie II. Wydawnictwo: Helion, 2018.

#### Literatura dodatkowa:

1. A. Nathan. WPF 4.5. Księga eksperta (ebook). Helion 2015.
2. I. Miell, A. H. Sayers. Docker w praktyce (ebook). Wydawnictwo Naukowe PWN. 2020.

#### Planowane formy/działania/metody dydaktyczne:

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

#### Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W\_01 – W\_02 będą sprawdzane podczas pisemnego zaliczenia wykładu na ocenę. Część pytań może mieć charakter otwarty, a część zamknięty (test).

Przykładowe pytania o charakterze otwartym:

- Wymień i scharakteryzuj znane Ci typy wirtualizacji.
- Do czego służą interpolatory animacji? Wymień i opisz ich typy w WPF?
- Scharakteryzuj możliwości i ograniczenia procesu głównego i procesu renderującego w ElectronJS.

Przykładowe pytania o charakterze zamkniętym:

1. Który kontener WPF umożliwia pozycjonowanie absolutne?
  - a. StackPanel,
  - b. Canvas,
  - c. StackPanel,
  - d. Resource.
2. Który język znacznikowy jest stosowany na platformie JavaFX.
  - a. MXML,
  - b. XAML,
  - c. FXML.
  - d. JDL.

Efekt U\_01 - U\_02 oraz K1 - K2 będą systematycznie sprawdzane na zajęciach laboratoryjnych. 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.

Przykładowe zadanie:

- Wykorzystując klasę *WebView2* zaimplementuj prostą aplikację hybrydową w WPF – formularz rejestracyjny. Użyj animacji CSS3 w celu podkreślenia multimedialnego charakteru 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 cząstkowe 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 uzyskać do 40 pkt. Zaliczenie kolokwium jest możliwe po uzyskaniu co najmniej 20 pkt

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

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

Poprawy:

3. Jednorazowa poprawa 2 wybranych przez studenta laboratoriów w trakcie konsultacji, które trwają do zakończenia zajęć laboratoryjnych w danym semestrze.
4. Jednorazowa poprawa projektu indywidualnego w sesji egzaminacyjnej.
5. Jednorazowa poprawa kolokwium 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	30 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	20 godz.
Sumaryczne obciążenie pracą studenta	<b>100 godz.</b>
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	40 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 godz.
Sumaryczne obciążenie pracą studenta	<b>100 godz.</b>
Punkty ECTS za przedmiot	4 ECTS

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>	<b>Symulacja komputerowa</b>	
<b>Nazwa w języku angielskim:</b>	Computer simulation	
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>	Informatyka	
<b>Jednostka realizująca:</b>	Instytut Informatyki	
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>	fakultatywny	
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>	pierwszego stopnia	
<b>Rok studiów:</b>	trzeci	
<b>Semestr:</b>	piąty	
<b>Liczba punktów ECTS:</b>	3	
<b>Imię i nazwisko koordynatora przedmiotu:</b>	dr Anna Wawrzyńczak-Szaban, dr Ewa Szczepanik	
<b>Imię i nazwisko prowadzących zajęcia:</b>	dr Anna Wawrzyńczak-Szaban, dr Ewa Szczepanik	
<b>Założenia i cele przedmiotu:</b>	Celem modułu jest poznanie podstawowych technik modelowania i symulacji komputerowych. Jest to jedno z najważniejszych zastosowań komputerów. Modelujemy w celu zrozumienia istoty modelowanego procesu, dopasowania parametrów modelu, badania wpływu różnych czynników na model.	
<b>Symbol efektu</b>	<b>Efekty kształcenia</b>	<b>Symbol efektu kierunkowego</b>
	<b>WIEDZA</b>	
<b>W_01</b>	Zna etapy modelowania systemów i procesu tworzenia modelu symulacyjnego	<b>K_W06</b>
<b>W_02</b>	Orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych z zakresu metod modelowania i symulacji komputerowej,	<b>K_W07, K_W09</b>
<b>UMIĘJĘTNOŚCI</b>		
<b>U_01</b>	Potrafi zanalizować wybrany proces lub zjawisko oraz utworzyć jego model matematyczny.	<b>K_U01, K_U07</b>
<b>U_02</b>	Potrafi ułożyć algorytm i przeprowadzić symulację dla zadanego problemu i zaimplementować go w języku MATLAB lub innym wybranym języku programowania.	<b>K_U07, K_U12, K_U22</b>
<b>U_03</b>	Potrafi utworzyć pełny model symulacyjny wybranego zjawiska/procesu oraz przeprowadzić symulację komputerową, eksperymentować z doбором parametrów.	<b>K_U09, K_U20</b>
<b>Forma i typy zajęć:</b>	studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.)	

**Wymagania wstępne i dodatkowe:**

Umiejętność programowania, znajomość analizy matematycznej i podstaw fizyki.

**Treści modułu kształcenia:**

1. Idea modelowania. Pojęcie modelu. Klasyfikacja modeli. Symulacja systemów naturalnych.
2. Etapy modelowania systemów. Model cybernetyczny i jego modyfikacje.
3. Model fizyczny. Przykłady modeli fizycznych wybranych systemów.
4. Model matematyczny. Ogólna postać modelu matematycznego.
5. Klasyfikacja modeli matematycznych. Przykłady modeli matematycznych wybranych systemów.
6. Model symulacyjny. Proces budowy modelu symulacyjnego. Reguły upraszczania modeli. Techniczne aspekty badań symulacyjnych.
7. Metoda Monte Carlo. Próbkowanie ważne. Procesy Markowa. Równowaga szczegółowa. Algorytm Metropolisa.
8. Symulacje Monte Carlo w różnych zespołach statystycznych.
9. Wprowadzenie do modelowania systemów masowej obsługi (teoria kolejek). Procesy stochastyczne. Procesy Markowa. Zastosowanie teorii kolejek w badaniach wydajności systemów komputerowych.
10. Symulacja przykładowych modeli matematycznych dynamicznych systemów ciągłych za pomocą oprogramowania Matlab i Simulink (systemy mechaniczne, elektryczne, ekonomiczne).
11. Narzędzia programistyczne. Grafika w środowiskach programistycznych. Wizualizacja wyników symulacji komputerowej przy użyciu podstawowych narzędzi programistycznych.
12. Symulacje przykładowych procesów mechanicznych i elektrycznych.
13. Zaawansowane pakiety oprogramowania symulacyjnego (np. MatLab, MODELLUS, AnyLogic, Easy Java Simulations). Wymiana danych pomiędzy różnymi środowiskami użytkowymi.

**Literatura podstawowa:**

1. Krajka Andrzej, Modelowanie i symulacje, Wydaw UMCS, Lublin 2012
2. Kurowski W.: Modelowanie i symulacja systemów technicznych, Wydawnictwo WSA. Łomża 2002. [www.wsa.edu.pl](http://www.wsa.edu.pl)
3. Gutenbaum J.: Modelowanie matematyczne systemów. Akademicka Oficyna Wydawnicza EXIT. Warszawa 2003.
4. Kotowski Romuald, Tronczyk Piotr, Modelowanie i symulacje komputerowe. Wydaw. Uniw. Kazimierza Wielkiego w Bydgoszczy. Bydgoszcz 2009.

**Literatura dodatkowa:**

1. Oniszczyk W.: Metody modelowania. Wydawnictwa Politechniki Białostockiej 1995. Podlaska Biblioteka Cyfrowa. <http://pbc.biaman.pl/dlibra>
2. Tikhonenko O.: Modele obsługi masowej w systemach informacyjnych. Akademicka Oficyna Wydawnicza EXIT. Warszawa
3. Matyka Maciej, Symulacje komputerowe w fizyce. Helion, Gliwice 2002. (<http://panoramix.ift.uni.wroc.pl/~maq/pl/>).

**Planowane formy/działania/metody dydaktyczne:**

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko umożliwiające prowadzenie symulacji komputerowych np. MatLab. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

**Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:**

Efekty W\_01 – W\_02 będą weryfikowane poprzez kolokwium pisemne na ostatnim wykładzie.

Przykładowe pytania:

- Wymień i scharakteryzuj etapy modelowania systemów
- Opisz różnice pomiędzy modelem fizycznym i matematycznym.
- W jakich zagadnieniach mają zastosowanie procesy Markowa?

Efekty U\_01 -U\_03 sprawdzane będą na zajęciach laboratoryjnych. Zadania na następne laboratorium muszą być dostępne 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, 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 21 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 – 80 punktów: niedostateczna (F),
- 81 – 96 punktów: dostateczna (E),
- 97 – 112 punktów: dostateczna plus (D),
- 113 – 128 punktów: dobra (C),
- 129 – 144 punktów: dobra plus (B),
- 145 – 160 punktów: bardzo dobra (A).

Poprawy:

Uzyskanie poprawkowego zaliczenia laboratoriów możliwe jest w sesji egzaminacyjnej, odpowiednio przed drugim i trzecim terminem egzaminu pisemnego.

**Bilans punktów ECTS:****Studia stacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń	10 godz.
Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się i udział w egzaminie	10 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>

<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>
<b>Studia niestacjonarne</b>	
<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godz.
Udział w konsultacjach godz. z przedmiotu	10 godz.
Przygotowanie się do egzaminu	15 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

<b>Sylabus przedmiotu / modułu kształcenia</b>		
<b>Nazwa przedmiotu/modułu kształcenia:</b>	<b>Tworzenie interfejsów użytkownika</b>	
<b>Nazwa w języku angielskim:</b>	Creating user interfaces	
<b>Język wykładowy:</b>	polski	
<b>Kierunek studiów, dla którego przedmiot jest oferowany:</b>	informatyka	
<b>Jednostka realizująca:</b>	Wydział Nauk Ścisłych i Przyrodniczych	
<b>Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):</b>	fakultatywny	
<b>Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):</b>	pierwszego stopnia	
<b>Rok studiów:</b>	trzeci	
<b>Semestr:</b>	piąty	
<b>Liczba punktów ECTS:</b>	3	
<b>Imię i nazwisko koordynatora przedmiotu:</b>	dr Marcin Stępniaak	
<b>Imię i nazwisko prowadzących zajęcia:</b>	dr Marcin Stępniaak	
<b>Założenia i cele przedmiotu:</b>	<p>Studenci przystępujący do tego przedmiotu powinni posiadać podstawową wiedzę z zakresu grafiki komputerowej i technologii internetowych.</p> <p>Celem przedmiotu jest zapoznanie studentów z historią i ewolucją komunikacji człowiek-komputer, a także ze specyfiką zmysłów (np. wzrok, słuch) wykorzystywanych w tej komunikacji. Studenci poznają zasady i normy projektowania IU oraz sposoby oceny ich jakości. Poza powyższymi studenci poznają także narzędzia i technologie wykorzystywane w projektowaniu IU.</p>	
<b>Symbol efektu</b>	<b>Efekt uczenia się: WIEDZA</b>	<b>Symbol efektu kierunkowego</b>
<b>W_01</b>	Zna i rozumie zagadnienia związane z podbudowaną teoretycznie wiedzą w zakresie zasad i metodyki projektowania interfejsów użytkownika ze szczególnym uwzględnieniem fazy analizy i modelowania konceptualnego.	<b>K_W03, K_W06, K_W09</b>
<b>W_02</b>	Zna i rozumie zagadnienia dotyczące oceny użyteczności multimedialnego interfejsu użytkownika.	<b>K_W09</b>
<b>W_03</b>	Zna i rozumie technologie i narzędzia wykorzystywane w projektowaniu IU.	<b>K_W06, K_W12</b>

<b>W_04</b>	Zna i rozumie aspekty mające wpływ na trendy rozwojowe interfejsów użytkownika. Zna i rozumie najistotniejsze nowe osiągnięcia w zakresie interfejsów użytkownika.	<b>K_W03, K_W12</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: UMIEJĘTNOŚCI</b>	<b>Symbol efektu kierunkowego</b>
<b>U_01</b>	Potrafi opracować szczegółową dokumentację projektu interfejsu użytkownika ze szczególnym uwzględnieniem widoków użytkownika.	<b>K_U08</b>
<b>U_02</b>	Potrafi zaproponować ulepszenia istniejących rozwiązań, ocenić przydatność i możliwość wykorzystania nowych osiągnięć w zakresie metod do projektowania i implementacji systemów informatycznych z naciskiem na interfejs użytkownika.	<b>K_U10, K_U15</b>
<b>U_03</b>	Potrafi zastosować odpowiednie techniki i narzędzia do projektowania i budowy interfejsów użytkownika.	<b>K_U02, K_U10</b>
<b>U_04</b>	Potrafi integrować wiedzę, pochodzącą z różnych źródeł, z dziedziny informatyki i innych dyscyplin, stosując podejście systemowe, z uwzględnieniem aspektów pozatechnicznych (w tym psychologii poznania).	<b>K_U01</b>
<b>Symbol efektu</b>	<b>Efekt uczenia się: KOMPETENCJE SPOŁECZNE</b>	<b>Symbol efektu kierunkowego</b>
<b>K_01</b>	Jest gotów myśleć i działać w sposób kreatywny. Zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego kształcenia.	<b>K_K01</b>
<b>K_02</b>	Jest gotów do przekazywania społeczeństwu informacji i opinii dotyczących osiągnięć informatyki.	<b>K_K02</b>
<b>Forma i typy zajęć:</b>	Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)	
<b>Wymagania wstępne i dodatkowe:</b>		
1. Wiedza z zakresu grafiki komputerowej. 2. Wiedza i umiejętności z zakresu technologii WWW.		
<b>Treści modułu kształcenia:</b>		
1. <b>Rola i ewolucja użytkownika i komunikacji człowiek-komputer.</b> Podstawowe pojęcia: Interfejs użytkownika, Multimedialny Interfejs Użytkownika, Mentalny Model Użytkownika. Historia i ewolucja interfejsów użytkownika; 2. <b>Percepcja obrazów.</b> Filozofia widzenia. Percepcja barw. Złudzenia optyczne 3. <b>Percepcja dźwięków.</b> Jak słyszymy? Rozpoznawanie mowy. 4. <b>Zasady projektowania IU.</b> Psychologia postaci. Zasady percepcji. Zasady ergonomiczne. Wykorzystanie zasad percepcji w projektowaniu interfejsów. 5. <b>Normy projektowe.</b> Normy: ISO 9126, ISO/IEC 25010, ISO 9241. 6. <b>Zasady i standardy tworzenia dostępnych interfejsów użytkownika.</b>		



7. **Techniki szczegółowego modelowania działań użytkownika.** Poziomy zadań. Poziom semantyczny. Poziom syntaktyczny.
8. **Projektowanie interfejsu użytkownika.** Projektowanie User Experience (UX design), Projektowanie zorientowane na użytkownika - User Centered Design (UCD), Projektowanie IU, Metody Agile UX
9. **Ocena jakości interfejsu użytkownika** Pojęcie użyteczności IU. Heurystyki Nielsena. Kryteria oceny jakości interfejsu graficznego i multimedialnego. Rodzaje testów. Kwestionariusze. Eye-tracking. Web-usability.
10. **Projektowanie i użyteczność interfejsów głosowych.** Interakcyjne tworzenie systemu. Ocena jakości VUI;
11. **Narzędzia i technologie wykorzystywane w projektowaniu IU.**

#### Literatura podstawowa:

1. M. Kasperski, Anna Boguska-Torbicz Projektowanie stron WWW. Użyteczność w praktyce, Helion 2008.
2. J.Nielsen, Projektowanie funkcjonalnych serwisów internetowych, Helion 2003.
3. J. Tidwell Projektowanie interfejsów. Sprawdzone wzorce projektowe, Helion 2012.

#### Literatura dodatkowa:

1. J. Spolsky, Projektowanie interfejsu użytkownika. Poradnik dla programisty, Mikom, 2001.
2. Masarek K., Sikorski M. (red), Interfejs użytkownika Kansei w praktyce, Wydawnictwo PJWSTK, Warszawa 2006.
3. The Essential Guide to User Interface Design - An Introduction to GUI Design Principles and Techniques, Wilbert O. Galitz, Wiley Publishing, Inc. 2007.
4. Designing with the Mind in Mind - Simple Guide to Understanding User Interface Design Rules, Jeff Johnson, Elsevier, 2010.

#### Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia praktyczne. Zamieszczanie na stronach internetowych materiałów do zajęć laboratoryjnych.

#### Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Efekty W\_01 - W\_04 będą weryfikowane w trakcie zajęć laboratoryjnych i realizacji zadania indywidualnego.

Efekty U\_01 - U\_04 weryfikowane będą w trakcie zajęć oraz podczas oceny realizacji zadania indywidualnego.

Efekty K\_01 - K\_02 będą weryfikowane podczas prezentacji zadania indywidualnego przez studenta.

#### Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu:

- Regularne zajęcia – 100 pkt.
- Obrona zadania indywidualnego – 60 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej 51% sumy punktów. Wymagane jest uzyskanie co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 50 pkt., obrona indywidualnego zadania – co najmniej 30 pkt. Ocena końcowa z modułu

(wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 160pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

- 0 – 80 punktów: niedostateczna (F),
- 81 – 96 punktów: dostateczna (E),
- 97 – 112 punktów: dostateczna plus (D),
- 113 – 128 punktów: dobra (C),
- 129 – 144 punktów: dobra plus (B),
- 145 – 160 punktów: bardzo dobra (A).

Poprawy:

- Poprawa wybranych laboratoriów w trakcie konsultacji. W jednym tygodniu można poprawić maksymalnie 2 ćwiczenia laboratoryjne.
- Poprawa zadania indywidualnego w sesji egzaminacyjnej przed drugim terminem egzaminu.

### **Bilans punktów ECTS:**

#### **Studia stacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>

#### **Studia niestacjonarne**

<b>Aktywność</b>	<b>Obciążenie studenta</b>
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.
<b>Sumaryczne obciążenie pracą studenta</b>	<b>75 godz.</b>
<b>Punkty ECTS za przedmiot</b>	<b>3 ECTS</b>