

*Załącznik do uchwały Senatu Uniwersytetu WSB Merito we Wrocławiu nr 51/2026 z dnia 24 marca 2026 roku w sprawie ustalenia programów studiów dla kierunku Informatyka realizowanym na Wydziale Finansów i Zarządzania Uniwersytetu WSB Merito we Wrocławiu dla cyklu kształcenia rozpoczynającego się w roku akademickim 2026/2027 - nabór letni*



Uniwersytet WSB Merito we Wrocławiu  
Wydział Finansów i Zarządzania

Program studiów  
dla kierunku

**Informatyka**  
**studia I stopnia**

Studia: stacjonarne/niestacjonarne  
Profil: praktyczny  
Rok akademicki: 2026/2027

## I. OGÓLNA CHARAKTERYSTYKA KIERUNKU STUDIÓW

<b>nazwa kierunku studiów</b>	<b>Informatyka</b>	
<b>Poziom kształcenia</b> (studia pierwszego stopnia / studia drugiego stopnia / jednolite studia magisterskie)	studia pierwszego stopnia	
<b>Profil kształcenia</b>	praktyczny	
<b>Forma studiów</b> stacjonarne/niestacjonarne	stacjonarne/niestacjonarne	
<b>Czas trwania studiów (w semestrach)</b>	7	
<b>Łączna liczba punktów ECTS dla danej formy studiów.</b>	Studia stacjonarne 210	Studia niestacjonarne 210
<b>Łączna liczba godzin określona w programie studiów</b>	Studia stacjonarne 3028	Studia niestacjonarne 2394
<b>Tytuł zawodowy nadawany absolwentom</b>	inżynier	
<b>Wymiar praktyk zawodowych</b>	960 godzin	
<b>Język prowadzenia studiów</b>	polski lub angielski	
<b>Rok rozpoczęcia cyklu kształcenia</b>	2026	

## II. EFEKTY UCZENIA SIĘ

Symbol	Kierunkowe efekty uczenia się	Odniesienie do charakterystyk drugiego stopnia zgodnie z PRK	Odniesienie do charakterystyk drugiego stopnia prowadzących do osiągnięcia kompetencji inżynierskich
<b>WIEDZA</b> <b>absolwent zna i rozumie</b>			
K_W01	pojęcia analizy matematycznej, algebry, logiki konieczne do zrozumienia i rozwiązywania zadań inżynierskich.	P6S_WG	P6S_WG
K_W02	w zaawansowanym stopniu zasady funkcjonowania aparatury informatycznej, działania elektronicznych urządzeń pomiarowych.	P6S_WG	P6S_WG
K_W03	w zaawansowanym stopniu zasady działania urządzeń wchodzących w skład systemów i sieci telekomunikacyjnych w tym sieci bezprzewodowych, światłowodowych oraz konfigurowania tych urządzeń w sieciach lokalnych, a także w zakresie urządzeń	P6S_WG	P6S_WG

	wchodzących w skład przemysłowych sieci komunikacyjnych.		
K_W04	zasady zarządzania organizacjami.	P6S_WK	
K_W05	możliwości praktycznego zastosowania programów i systemów związanych z technologiami informacyjnymi.	P6S_WG	
K_W06	zasady tworzenia i funkcjonowania systemów baz danych.	P6S_WG	
K_W07	zasady i metody wizualizacji danych, tworzenia opracowań formalnych, technik realizacji i wygłaszania prezentacji multimedialnej oraz prowadzenia telekonferencji internetowych.	P6S_WG	
K_W08	w zaawansowanym stopniu metody programowania na poziomie języków wysokiego poziomu, zna konstrukcje programistyczne, algorytmy oraz metody oceny i testowania poprawności programów.	P6S_WG	P6S_WG
K_W09	w zaawansowanym stopniu architekturę oprogramowania; zna rozwiązania klient-serwer, architekturę SOA; rozumie działanie zaawansowanych platform programistycznych.	P6S_WG	P6S_WG
K_W10	metody analizy i prezentacji danych biznesowych z wykorzystaniem informatycznych pakietów analizy danych i ich praktyczne zastosowanie.	P6S_WG	
K_W11	w zaawansowanym stopniu metody i narzędzia modelowania i projektowania informatycznych systemów zarządzania procesami biznesowymi.	P6S_WG	P6S_WG
K_W12	w zaawansowanym stopniu metody inżynierii oprogramowania, w tym projektowania (wzorce projektowe, architektura oprogramowania, analiza i projektowanie obiektowe), wykorzystania API, narzędzi i środowisk wytwarzania oprogramowania (narzędzia do analizy wymagań i modelowania, narzędzia do testowania, narzędzia do podglądu kodu, narzędzia do zarządzania konfiguracjami i wersjami oprogramowania), cyklu życia projektu informatycznego, specyfikacji oprogramowania, walidacji i weryfikacji, utrzymywania oprogramowania (refaktoryzacji).	P6S_WG	P6S_WG
K_W13	w zaawansowanym stopniu zasady programowania obiektowego, pojęcie klasy i obiektu, zalety programowania obiektowego i metod oraz przypadków użycia, zasady pracy z obiektami, złożonymi strukturami danych (zbiory, listy, stosy, kolejki, drzewa i pojęcie wskaźnika, referencji i obiektu).	P6S_WG	P6S_WG

K_W14	sposób działania systemów teleinformatycznych oraz sieci komputerowych.	P6S_WG	
K_W15	zasady bezpieczeństwa i ochrony danych w systemach IT; zna zagrożenia i zasady zabezpieczeń oraz oprogramowanie wspomagające monitoring i ochronę sieci intranet; rozumie normy i procedury postępowania z ryzykiem informacyjnym polityką i audytem bezpieczeństwa oraz skutki utraty informacji.	P6S_WG P6S_WK	P6S_WG, P6S_WK
K_W16	w zaawansowanym stopniu ogólne zasady w zakresie programowania, algorytmów i złożoności, języków formalnych i automatów, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, grafiki i komunikacji człowiek-komputer, baz danych, inżynierii oprogramowania, zawartą w treściach przedmiotów programu studiów.	P6S_WG	P6S_WG
K_W17	zagadnienia z zakresu nauk humanistycznych odnoszących się do technologii IT; zna ekonomiczne, prawne i etyczne uwarunkowania działalności informatyka w tym zasady i pojęcia z zakresu ochrony własności przemysłowej i prawa autorskiego; rozumie społeczny kontekst systemów IT.	P6S_WG	
K_W18	w zaawansowanym stopniu produkty rynkowe z zakresu informatyki w tym open source; zna produkty hardware, urządzenia wejścia-wyjścia; rozumie czym jest funkcjonalność i użyteczność wybranej technologii IT.	P6S_WG	
K_W19	procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych Informatyki; rozumie zagrożenia wynikające ze złożoności oraz niedoskonałości systemów IT i konieczność ich ciągłej modernizacji.	P6S_WG P6S_WK	P6S_WG P6S_WK
<b>UMIEJĘTNOŚCI absolwent potrafi</b>			
K_U01	posługiwać się aparatem analizy matematycznej, statystyki matematycznej, algorytmami analizy numerycznej oraz rozwiązywać złożone i nietypowe problemy z ich wykorzystaniem.	P6S_UW	
K_U02	wykorzystać poznane modele i metody, a także symulacje komputerowe do rozwiązywania zadań inżynierskich. Potrafi przedstawić otrzymane wyniki w formie liczbowej i graficznej, dokonać ich interpretacji i wyciągnąć właściwe wnioski.	P6S_UW	P6S_UW
K_U03	samodzielnie projektować algorytmy rozwiązując zadania (obliczeniowe, wyszukujące, porządkujące), kompiluje i	P6S_UW	P6S_UW

	wykonuje programy na różnych platformach systemowych, potrafi podać specyfikacje algorytmów i zapisać algorytmy w postaci pseudokodu, umie wyznaczyć rząd złożoności algorytmów, implementuje algorytmy i dobiera odpowiednie struktury danych, analizuje wpływ struktur danych na złożoność programów.		
K_U04	ocenić, przydatności rutynowych metod i narzędzi informatycznych potrafi wybrać i zastosować właściwą metodę i narzędzia do typowych zadań informatycznych, potrafi wykonać analizę sposobu funkcjonowania systemu informatycznego i ocenić istniejące rozwiązania informatyczne, przynajmniej w odniesieniu do ich cech funkcjonalnych, tworzy, ocenia i realizuje plan testowania oprogramowania.	P6S_UW	
K_U05	projektować i wdrażać systemy IT wspomagające zarządzanie oraz technologie multimedialne; potrafi planować i prowadzić projekty IT stosując zasady bezpieczeństwa informacyjnego; potrafi oszacować koszt oraz zasadność projektu IT posługując się normami, standardami oraz analizami ekonomicznymi (CBA i wielokryterialną).	P6S_UO P6S_UW	P6S_UW
K_U06	posługiwać się sprzętem multimedialnym, tworzyć prezentacje multimedialne, wizualizować wyniki ekonomiczne, potrafi swobodnie funkcjonować w sferach komunikacji multimedialnej oraz wirtualnej.	P6S_UW	
K_U07	konfigurować serwery WWW oraz portale korporacyjne w oparciu o platformy CMS i technologie mobilne; potrafi konfigurować systemy CMR, CMS; potrafi wykonać stronę WWW za pomocą standardowych generatorów; stosuje oprogramowanie open source oraz standardowe pakiety biurowe i graficzne.	P6S_UW	P6S_UW
K_U08	czytać ze zrozumieniem programy zapisane w języku programowania jak również tworzyć i testować programy komputerowe; potrafi uruchomić i podłączyć program wspomagający określoną funkcjonalność do procesów organizacji; stosuje analizatory kodu oraz makropolecenia dostępne w pakietach narzędziowych.	P6S_UW	
K_U09	konstruować algorytmy komputerowe, wykonać analizę złożoności algorytmów, a także zaprojektować schemat relacyjnej bazy danych; potrafi wykorzystywać generatory wysokiego poziomu oraz platformy klient-serwer; wykorzystuje standardowe narzędzia w projektowaniu i kodowaniu.	P6S_UW	P6S_UW

K_U10	instalować, konfigurować i diagnozować sieci hybrydowe i LAN/WAN, potrafi instalować i konfigurować serwery aplikacji, proxy, SQL, Microsoft Windows na poziomie podstawowym, potrafi zarządzać usługami w sieci oraz klientami	P6S_UW	P6S_UW
K_U11	wdrażać narzędzia wirtualizacji biznesu oraz przenoszenia danych do chmury; potrafi stosować zasady funkcjonowania gospodarki elektronicznej, a w szczególności technologie teleinformatyczne BI, potrafi zintegrować dane; wykorzystuje narzędzia e-commerce, e-bankingu, e-logistyki oraz platformy wirtualnych usług i cloud computing.	P6S_UW	
K_U12	krytycznie analizować dokumenty projektowe w zakresie IT (studium wykonalności, instrukcja, projekt sieci); potrafi przygotować w języku polskim i angielskim dokumentację dotyczącą realizacji projektu informatycznego oraz raport prezentujący zrealizowane zadanie; wykorzystuje pakiety biurowe i generatory dokumentów.	P6S_UW	
K_U13	projektować, tworzyć i testować aplikacje z wykorzystaniem platform low-code i no-code; integruje te aplikacje z istniejącymi systemami informatycznymi; stosuje narzędzia wizualne do modelowania procesów biznesowych oraz zapewnia bezpieczeństwo i jakość tworzonych rozwiązań.	P6S_UW	
K_U14	posługiwać się językiem obcym na poziomie B2 Europejskiego Systemu Opisu Kształcenia Językowego.	P6S_UK	
K_U15	pracować i współdziałać w różnych grupach społecznych i w różnych rolach.	P6S_UO	
K_U16	samodzielnie zdobywać, uzupełniać i doskonalić wiedzę oraz umiejętności zawodowe przez całe życie, potrafi podejmować decyzje o dalszym uczeniu się.	P6S_UU	
K_U17	prezentować zdobytą wiedzę oraz umiejętności w mowie i piśmie z wykorzystaniem m.in. narzędzi informacyjnych, potrafi oceniać różne stanowiska oraz dyskutować o nich, potrafi brać udział w debacie.	P6S_UK	
<b>KOMPETENCJE SPOŁECZNE</b> <b>absolwent jest gotów do</b>			
K_K01	brania odpowiedzialności za powierzone mu zadania związane z wykonywaną pracą oraz pełnioną rolą zawodową, przestrzegania zasad etyki zawodowej i wymagania tego od innych oraz podejmowania działań mających	P6S_KR	

	na celu dbałość o dorobek i tradycje zawodów związanych ze studiowanym kierunkiem.		
K_K02	myślenia i działania w sposób przedsiębiorczy, postępować odpowiedzialnie mając świadomość skutków pozatechnicznych swojej działalności. Jest gotów do inicjowania i współorganizowania działań na rzecz środowiska i interesu publicznego.	P6S_KO	
K_K03	dokonania krytycznej samooceny stanu swojej wiedzy i umiejętności z zakresu informatyki oraz doceniania znaczenia wiedzy i zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązaniem problemów poznawczych i praktycznych.	P6S_KK	

**III. ZAJĘCIA LUB GRUPY ZAJĘĆ NIEZALEŻNIE OD FORMY PROWADZENIA WRAZ Z PRZYPISANIEM DO NICH EFEKTÓW UCZENIA SIĘ I TREŚCI PROGRAMOWYCH ZAPEWNIAJĄCYCH UZYSKANIE EFEKTÓW**

## A) PRZYPISANIE EFEKTÓW UCZENIA SIĘ DO ZAJĘĆ LUB GRUPY ZAJĘĆ NIEZALEŻNIE OD FORMY ICH PROWADZENIA

Nazwa przedmiotu	K_W01	K_W02	K_W03	K_W04	K_W05	K_W06	K_W07	K_W08	K_W09	K_W10	K_W11	K_W12	K_W13	K_W14	K_W15	K_W16	K_W17	K_W18	K_W19	K_U01	K_U02	K_U03	K_U04	K_U05	K_U06	K_U07	K_U08	K_U09	K_U10	K_U11	K_U12	K_U13	K_U14	K_U15	K_U16	K_U17	K_K01	K_K02	K_K03										
Deep Impact / Tiefer Einfluss																	x																																
Ekonomia				x																																													
Etyka																	x																																
Język obcy																																																	
Język obcy																																																	
Język obcy																																																	
Język obcy																																																	
Metody efektywnego uczenia się																																																	
Microsoft Office 365 i Google Apps SCRIPT	x				x		x														x																												
Podstawy komunikacji społecznej																																																	
Podstawy zrównoważonego rozwoju																		x																															
Prawo nowych technologii																																																	
Proseminarium																																																	

















**B) ZAJĘCIA LUB GRUPY ZAJĘĆ ORAZ TREŚCI PROGRAMOWE ZAPEWNIAJĄCE  
UZYSKANIE EFEKTÓW UCZENIA SIĘ**

Nazwa przedmiotu	Treści programowe
BHP	Wprowadzenie do problematyki bezpieczeństwa i higieny pracy. Prawne aspekty bezpieczeństwa i higieny pracy.
	Pomieszczenia i warunki środowiskowe. Charakterystyka zagrożeń.
	Pracownie na uczelni. Wypadki na uczelni.
	Ochrona przeciwpożarowa. Pierwsza pomoc w nagłych wypadkach.
Deep Impact / Tiefer Einfluss	Characteristics of economic development as a space for introducing deep impact. / Merkmale der wirtschaftlichen Entwicklung als Rahmen für die Einführung tiefgreifender Auswirkungen.
	Innovation as impacts for economic growth and enterprise development. / Innovation als treibende Kraft für Wirtschaftswachstum und Unternehmensentwicklung.
	Implementation of impacts, using change management methodology. / Umsetzung von tiefgreifenden Auswirkungen unter Anwendung von Methoden des Veränderungsmanagements.
	Deep impacts in practice. / Tiefgreifende Auswirkungen in der Praxis.
Ekonomia	Przedmiot i zakres ekonomii.
	Analiza popytu i podaży-wyznaczanie punktu równowagi rynkowej. Zmiana punktu równowagi rynkowej - zadania.
	PKB i inne wskaźniki makroekonomiczne - analiza danych statystycznych, ujęcie nominalne i realne.
	Wpływ zmian stóp procentowych na gospodarkę - inflacja i polityka pieniężna -zadania i przykłady.
	Analiza polityki rynku pracy, obliczanie stopy bezrobocia.
	Wpływ cen i embarga na gospodarkę. Zmiany kursu walutowego a gospodarka.
	Przykłady błędów poznawczych w decyzjach finansowych.
	Analiza wpływu operacji otwartego rynku, rezerw obowiązkowych i stop procentowych na gospodarkę.
	Podstawowe prawa rynku.
	Mierniki makroekonomiczne.
	Determinanty dochodu narodowego.
	Model wzrostu dochodu wg. J. M. Keynesa.
	Bezrobocie i inflacja.
	Polityka monetarna i fiskalna.
Model IS-LM.	
Podział ekonomii na mikro i makro-przykłady i zadania.	
Etyka	Definicje etyki, przedmiot etyki, aspekty etyki.

	Etyka w biznesie i etyka gospodarcza.
	Etyka zarządzania. Etyka zawodowa. Społeczna odpowiedzialność biznesu (CSR) a etyka.
Język obcy	Rozumienie i analiza tekstów zamieszczonych w podręczniku Gramatyka i słownictwo ogólnobiznesowe i kierunkowe Komunikacja ustna w życiu codziennym i zawodowym. Komunikacja pisemna biznesowa
Metody efektywnego uczenia się	Proces uczenia się - fakty i mity o efektywnym uczeniu się. Jak działa pamięć, mózg niezwykle narząd, czy istnieją style uczenia się?
	Techniki efektywnego uczenia się (techniki pamięciowe, mapy myśli, kreatywne notatki, gry) oraz narzędzia wspierające te procesy.
	Jak sztuczna inteligencja może nam pomóc w procesie uczenia się? (kurs AI)
	Wyszukiwanie i weryfikacja źródeł informacji.
	Tworzenie opisów bibliograficznych oraz bibliografii.
	Narzędzia i aplikacje przydatne w procesie uczenia się.
	Prokrastynacja.
	Organizacja procesu uczenia się (wyniki eksperymentu myślowego).
Microsoft Office 365 i Google Apps Script	Wymagania dot. przygotowywania prac o charakterze naukowym oraz praktyczne wykorzystanie zaawansowanych funkcji edytora tekstu Word.
	Tworzenie i formatowanie arkuszy obliczeniowych przy wykorzystaniu Excel.
	Praca z narzędziami Office 365 i Google Apps script (interakcje z aplikacjami w ramach Google Workspace).
	Analiza wykorzystywanych narzędzi w zakresie prezentacji danych.
Podstawy komunikacji społecznej	Zasady efektywnej komunikacji.
	Informowanie a przekonywanie. Dwa typy komunikowania.
	Skuteczne komunikowanie niewerbalne.
	Zasady efektywnej dyskusji w grupie.
	Techniki erystyczne w publicznych dyskusjach.
	Jak skonstruować wystąpienie publiczne?
Podstawy zrównoważonego rozwoju	Wprowadzenie do zrównoważonego rozwoju
	Wymiar ekologiczny/ środowiskowy zrównoważonego rozwoju
	Wymiar społeczny zrównoważonego rozwoju
	Wymiar ekonomiczny/ biznesowy zrównoważonego rozwoju i ESG
Prawo nowych technologii	Podstawy prawa - pojęcia, system, gałęzie / wykładnia prawa.
	Warunki rejestracji i zastrzeżenia nazwy domeny jako znaku towarowego.
	Prawne aspekty sztucznej inteligencji / internetu rzeczy.
	Systemy AI wysokiego ryzyka i zabronione.
	Umowy w obrocie gospodarczym - w IT.

	Analiza przykładowej umowy SLA.
	Odpowiedzialność karna i cywilna.
	Analiza przykładowych orzeczeń.
	Wykładnie prawne.
	Standardy ochrony informacji, w tym danych osobowych oraz prywatności.
	ISO 27001 / RODO.
	Elementy tajemnicy przedsiębiorstwa oraz nieuczciwej konkurencji.
	Analiza przykładowej umowy NDA.
	Źródła prawa i ochrony nowych technologii.
	Krajowy System Cyberbezpieczeństwa / NIS 2.
	Ochrona wynalazków / wzorów przemysłowych.
Proseminarium	Wprowadzenie do problematyki przygotowania i pisania pracy dyplomowej; Wymogi merytoryczne stawiane pracy dyplomowej; Wymogi formalne pracy. Standardy obrony pracy.
Repetitorium z matematyki	Liczby rzeczywiste (działania na ułamkach, potęgach, pierwiastkach, logarytmach)
	Procenty, wyrażenia algebraiczne (wzory skróconego mnożenia)
	Równania i nierówności
	Funkcje liniowe i kwadratowe
Różnice kulturowe	Czym jest kultura? Wprowadzenie.
	Stereotypy i uprzedzenia.
	Główne orientacje kulturowe - G. Hofstede - R. Gesteland - E. Meyer.
	Komunikacja międzykulturowa. Komunikacja bezpośrednia i pośrednia. Bariery w komunikacji werbalnej. Komunikacja niewerbalna.
	Religia, wartości, postawy, zwyczaje – wpływ na biznes.
	Proces negocjacji międzykulturowych.
	Szok kulturowy.
Wprowadzenie do studiowania na kierunku	Ogólne informacje o WFIZ.
	Formalno-organizacyjno-prawne aspekty studiowania.
	Istota studiowania na wybranym kierunku studiów.
	Organizacja procesu kształcenia.
	Kultura akademicka i etyka studiowania.
	Działalność studencka, organizacje studenckie.
	Korzystanie z infrastruktury i zasobów w procesie uczenia się i pracy zespołowej.
Zakres wsparcia w procesie uczenia się.	
Administrowanie systemem Linux	Komendy i zastosowania. Omówienie wybranych komend, wyjaśnienie istoty systemu. Zapoznanie się z systemem Linux.
	Omówienie metod nadawania uprawnień oraz bardziej skomplikowanych komend. Testowanie zaawansowanych komend i określania uprawnień użytkowników.

	Omówienie sposobów deklaracji użytkowników w systemie Linux. Wykonanie praktycznego zadania dodawania użytkowników.
	Omówienie procesów w systemie Linux. Wprowadzenie komend oraz realizacja zadań związanych z procesami. Testowanie komend związanych z procesami na rzeczywistych problemach.
	Wprowadzenie do koncepcji wyrażeń regularnych. Przedstawienie rozwiązań i ich aplikacja.
	Rozwinięcie wyrażeń regularnych o dodatkowe elementy. Wprowadzenie do skryptów korzystających z nabytej wcześniej wiedzy.
	Tworzenie zaawansowanych skryptów.
Algorytmy i struktury danych	Podstawowe oraz zaawansowane struktury danych. Tablica. Lista i jej warianty: kolejka, stos, kolejka podwójna. Zbiór. Graf. Drzewo. Kopiec. Kolejka priorytetowa. Struktury. Unie. Abstrakcyjne typy danych.
	Algorytmy kompresji. Algorytmy RLE (Run Length Encoding) oraz Huffmana (o stałej i zmiennej długości kodu). Inne metody kompresji plików.
	Wybrane algorytmy tablicowe i listowe. Sortowanie i wyszukiwanie. Operacje oparte o zbiory liczbowe, tekstowe a także zaawansowane struktury.
	Algorytmy oparte o operacje na ciągach znaków. Kryptografia: algorytm Cezara, algorytm RSA. Kompresja danych: algorytm RLE, algorytm Huffmana.
	Algorytmy grafowe: przeszukiwanie włąb, przeszukiwanie wszerek, drzewa rozpinające, ścieżki, kolorowanie grafów, przepływy w sieciach, algorytm DMKM, problem komiwojażera. Problemy permutacyjne w grafach.
	Generacja obiektów kombinatorycznych. Znane typy obiektów: kombinacje, permutacje, nieporządki, podziały, drzewa itp. Algorytmy optymalizacyjne. Zastosowania: wyszukiwanie wyczerpujące, testowanie algorytmów kombinatorycznych.
	Wprowadzenie do struktur danych i ich zastosowania w algorytmach MinMax, wyszukiwania wartości w zbiorze uporządkowanym oraz nieuporządkowanym.
	Trzy klasy algorytmów sortowania w zastosowaniu do różnych typów danych. Złożoność obliczeniowa i empiryczne testy działania algorytmów z tej samej grupy.
	Algorytmy operujące na łańcuchach znaków. Palindromy proste oraz złożone, anagramy. Wyszukiwanie najdłuższego palindromu w łańcuchu znaków.
Algorytmy operujące na łańcuchach znaków. Porównanie różnego typu wyszukiwania wzorca w tekście, algorytmy BruteForce, Boyera-Moore'a, Knutha-Morrisa-Pratta oraz Rabina Karpa.	
Architektura systemów komputerowych	Istota działania systemu komputerowego, elementy składowe.
	Reprezentacje liczb całkowitych: uzupełnieniowa, spolaryzowana oraz SD. Dodawanie i odejmowanie w systemach uzupełnieniowych, nadmiar, elementy arytmetyki resztowej.
	Tworzenie i uruchamianie programów w językach assemblerowych. Deasemblacja programów.
	Architektura von Neumana, Architektura harwardzka, Maszyna Turinga.
	Procesory CISC i RISC. Charakterystyka, różnice, wady i zalety.
	Architektura x86, x86-64, ARM. Architektura pamięci RAM.
	Systemy wieloprocessorowe, NUMA. Architektury do specjalizowanych obliczeń (w tym GPU i DSP).
	Historia architektur, przegląd rozwiązań polskich i światowych na przestrzeni lat. Komputery ODRA, Setun (logika trójwartościowa).
Blockchain i jego zastosowanie	Wprowadzenie do blockchain. Typy łańcuchów bloków.
	Bezpieczeństwo technologii Blockchain. Skalowalność i inne problemy technologii blockchain.

	Decentralizacja z użyciem łańcucha bloków. Metody decentralizacji.
	Implementacje technologii łańcucha bloków. Kryptowaluty i inne zastosowanie.
	Wprowadzenie do Bitcoina. Sieć Bitcoina i płatności.4
	Klienci i interfejsy API Bitcoina.
	Organizacje zdecentralizowane.
	Algorytmy kryptograficzne.
	Ethereum i inne kryptowaluty
	Blockchain w nauce i w medycynie
Elektronika	Wybrane zjawiska dotyczące prądu elektrycznego, wielkości i jednostki elektryczne.
	Pomiar wielkości fizycznych – napięcie, natężenie, opór.
	Wykonywanie pomiarów wielkości fizycznych – napięcie, natężenie, opór.
	Praca z wybranymi zjawiskami dotyczącymi prądu elektrycznego, wielkościami i jednostkami elektrycznymi.
	Elementy obwodów. Wybrane prawa w elektronice.
	Analiza elementów obwodów, wybranych praw w elektronice.
	Warunki przepływu prądu, zdolność materiałów do przewodzenia prądu.
	Obwody. Obliczanie rozptyłu prądów w obwodach rozgałęzionych.
	Analiza obwodów. Obliczanie rozptyłu prądów w obwodach rozgałęzionych.
	Urządzenia i układy. Obliczanie rezystancji zastępczej układów.
	Obsługa urządzeń i układów. Obliczanie rezystancji zastępczej układów.
Fizyka	Przedstawienie wielkości wektorowych w kartezjańskim układzie współrzędnych, rachunek wektorowy.
	Zjawiska akustyczne – przykłady, Propagacja dźwięku, rezonans, natężenie i wysokość tonu - doświadczenia
	Przedstawianie wektorów w 2D i 3D, Operacje na wektorach (dodawanie, mnożenie skalarne i wektorowe), Zastosowanie rachunku wektorowego w fizyce.
	Mechanika punktu materialnego w układzie inercyjnym, Nieinercyjne układy odniesienia, siły i reakcje, Praca i energia mechaniczna. Zasada zachowania energii mechanicznej, Zasada zachowania pędu i jej zastosowanie.
	Ruch w układzie inercyjnym i nieinercyjnym (przykłady), Siły, reakcje, zasady dynamiki Newtona (zadania), Praca, energia mechaniczna i ich przemiany (zadania), Zasady zachowania energii i pędu – przykłady i zastosowania.
	Dynamika bryły sztywnej. Zasada zachowania momentu pędu, Wybrane elementy szczególnej i ogólnej teorii względności Einsteina, Ruch drgający oscylatora mechanicznego.
	Zadania z zakresu szczególnej teorii względności, Oscylator harmoniczny – rozwiązania i interpretacja, Równanie fali, energia i interferencja fal, Fale stojące – warunki powstawania.
	Fale mechaniczne: równanie i energia fali, interferencja fal, fale stojące , Elementy teorii kinetyczno-molekularnej gazu doskonałego, rozkłady Maxwella i Boltzmanna, przemiany stanu gazu doskonałego.

	Teoria kinetyczno-molekularna – podstawy, rozkłady Maxwella i Boltzmanna, Przemiany gazowe i równania stanu, Zasady termodynamiki, energia wewnętrzna, ekwipartycja energii.
	Zasady termodynamiki, energia wewnętrzna, zasada ekwipartycji energii, Elementy akustyki; co i jak słyszymy?
Funkcjonalność zintegrowanych systemów informatycznych SAP S/4HANA	Wprowadzenie do zintegrowanych systemów informatycznych klasy ERP.
	ABC obsługi systemu klasy ERP na przykładzie systemu SAP S/4HANA.
	Obsługa funkcjonalna systemów ERP na przykładzie wybranego procesu w systemie SAP S/4HANA.
	Praca na tabelach i praktyczne zastosowanie tabel w systemach klasy ERP.
	Tworzenie raportów w systemach klasy ERP na przykładzie systemu SAP S/4HANA.
	Customizacja systemów ERP na przykładzie systemu SAP S/4HANA.
	Zagadnienia administracyjne i programistyczne dotyczące zintegrowanych systemów informatycznych klasy ERP.
	Migracja danych.
Inżynieria bezpieczeństwa IT	Wprowadzenie do bezpieczeństwa IT.
	Incydenty bezpieczeństwa związane z uwierzytelnieniem (hasła i 2FA). Przegląd ataków na systemy IT i użytkowników.
	Bezpieczeństwo systemów operacyjnych.
	Bezpieczeństwo korzystania ze stron i serwisów społecznościowych.
	Certyfikaty SSL w systemach. Szyfrowanie poczty i komunikacji.
	Ochrona danych. Bezpieczeństwo korzystania z usług bankowych.
	Bezpieczeństwo urządzeń mobilnych.
	Bezpieczeństwo firmowych sieci LAN. Polityki ochrony brzegowej w systemach UTM.
Inżynieria Internetu rzeczy	Pojęcie Internetu rzeczy. Zagadnienia wprowadzające do Internetu rzeczy.
	Implementowanie rozszerzonych funkcjonalności w oparciu o urządzenia znajdujące się w laboratorium, praktyczne testowanie czujników.
	Zapoznanie się z możliwościami mikrokomputerów. Zaimplementowanie programu umożliwiającego działanie jako serwer dla Internetu rzeczy.
	Zapoznanie się z urządzeniami automatyki budynkowej. Zaimplementowanie zachowania przykładowego systemu automatyki budynkowej
	Zapoznanie się z wybranymi elementami oferowanymi przez firmy komercyjne związane z Internetem rzeczy. Implementowanie programów oraz testowanie urządzenia.
	Protokoły komunikacyjne Internetu rzeczy.
	Zastosowania Internetu rzeczy. Mikrokontroler Arduino Uno oraz Raspberry Pi.
	Automatyka budynkowa - urządzenia i zastosowania.
	Inteligentne miasta - przykładowe zastosowania.
	Bezpieczeństwo Internetu rzeczy
Zapoznanie się z pojęciem Internetu rzeczy, podstawowymi zastosowaniami. Przypomnienie podstawowych elementów z programowania.	

	Implementowanie i testowanie podstawowych funkcjonalności w oparciu o urządzenia znajdujące się w laboratorium.
	Wykorzystanie rozwiązań komunikacji zdalnej do zbierania informacji przesyłanych z mikrokomputera.
Inżynieria oprogramowania	Rola wytwarzania oprogramowania we współczesnym świecie, wizja projektu informatycznego. Modele cyklu życia oprogramowania.
	Projektowanie fizycznej infrastruktury informatycznej.
	Wprowadzenie do języka UML.
	Modelowanie systemów z wykorzystaniem języka UML.
	Narzędzia CASE wspomagające tworzenie oprogramowania.
	Tworzenie oprogramowania z wykorzystaniem narzędzi CASE.
	Metody testowania oprogramowania.
	Testowanie oprogramowania w praktyce.
	Praktyczne zadania związane z tworzeniem wizji projektu informatycznego.
	Analiza wymagań: definiowanie założeń projektowych, inżynieria wymagań. Wymagania funkcjonalne, нефункционалне i biznesowe. Wymagania z punktu widzenia użytkownika. Definiowanie wymagań biznesowych. Pozyskiwanie wymagań.
	Tworzenie i dokumentowanie wymagań funkcjonalnych i нефункционалnych.
	Inżynieria dostępności i niezawodności. Inżynieria bezpieczeństwa i odporności oprogramowania.
	Ćwiczenia z projektowania systemów z uwzględnieniem dostępności i niezawodności.
	Metody projektowania funkcji systemu i modelowanie przepływu informacji.
	Ćwiczenia z projektowania systemów z uwzględnieniem modelowania i przepływu informacji.
	Metody projektowania fizycznej infrastruktury informatycznej.
Inżynieria oprogramowania w środowisku MS .NET	Architektura systemów oprogramowania. Języki programowania, ich podział, paradygmaty programowania
	Obsługa błędów i wyjątków (try...)
	Obsługa błędów i wyjątków (try...) w praktyce programistycznej.
	Proste i złożone struktury danych. Tworzenie klas, obiektów i metod.
	Tworzenie klas, obiektów i metod.
	Tworzenie przykładowych aplikacji konsolowych.
	Tworzenie przykładowych aplikacji WPF.
	Zaawansowane platformy programistyczne. Programowanie wizualne w środowisku MS Visual Studio. Aplikacje graficzne (okienkowe): komponenty (przyciski, okienka edycyjne itp.).
	Implementacja wybranych algorytmów.
	Elementy języka programowania. Instrukcje, identyfikatory, zmienne, wyrażenia. Podstawowe grupy operatorów
	Wybrane algorytmy i konstrukcje programistyczne w języku C#

	Wybrane typy danych. Konwersje typów danych.
	Wykorzystanie wybranych typów danych w praktyce programistycznej.
	Stosowanie operatorów logicznych. Instrukcje decyzyjne (if ...), wyboru (switch ...)
	Zastosowanie operatorów logicznych i instrukcji decyzyjnych (if ...), wyboru (switch ...) w praktyce programistycznej.
	Złożone operatory przypisania. Instrukcje iteracyjne (for ...; while ...; do ...). Instrukcja zakończenia (break) oraz kontynuacji (continue); Priorytety operatorów
	Zastosowanie operatorów przypisania i instrukcji iteracyjne (for ...; while ...; do ...) w praktyce programistycznej.
Inżynieria serwerowych systemów operacyjnych Windows	Dystrybucje systemu Windows Server, opis instalacji systemu serwera i stacji roboczej.
	Instalacja usługi Active Directory oraz jej konfiguracja.
	Zarządzanie usługami domenowymi w usłudze Active Directory.
	Konfiguracja sieciowa serwera i klienta oraz wykorzystanie narzędzi diagnostycznych.
	Instalacja i konfiguracja serwera nazw DNS oraz serwera DHCP.
	Konfiguracja i zarządzanie pamięcią dyskową oraz serwerem plików.
	Konfiguracja środowiska pracy przez polityki grupowe.
	Instalacja i zarządzanie serwerem aplikacji oraz serwerem dostępu zdalnego.
	Pojęcia związane z domeną Active Directory, jej opis i konfiguracja usług domenowych.
	Obiekty Active Directory oraz zarządzanie środowiskiem Active Directory.
	Zagadnienia sieciowe związane z konfiguracją serwerów.
	Serwer nazw DNS oraz usługa automatycznej konfiguracji hosta DHCP.
	Pamięci dyskowe oraz serwery plików.
	Zasady grupy i zarządzanie nimi.
	Serwer aplikacji oraz serwer dostępu zdalnego.
Instalacja i konfiguracja systemu serwera i stacji roboczej.	
Inżynieria zachowań interdyscyplinarnych w organizacji	Analiza stylów komunikacji w zespole. Rozpoznawanie barier komunikacyjnych. Pierwsze wrażenie i jego znaczenie w budowaniu relacji zawodowych.
	Diagnoza dysfunkcji w zachowaniach organizacyjnych. Tworzenie profilu kultury etycznej.
	Sytuacje kryzysowe w środowisku wielokulturowym. Klasyfikacja kultur.
	Analiza ukrytych struktur władzy i kompetencji w zespole.
Komunikacja i efektywność pracy inżynierów	Efektywna komunikacja i aktywne słuchanie jako podstawa pracy inżynierów.
	Bariery komunikacyjne - jak je przezwyciężyć w komunikacji w obszarze IT oraz komunikacji w zespołach deweloperów?
	Feedback - jak komunikować informacje zwrotne? Przekazywanie informacji zwrotnych -pozytywnych oraz negatywnych (z uwzględnieniem pracy inżynierów).

	Komunikacja empatyczna w IT.
	Konflikt - umiejętność rozpoznawania i doskonalenie wybranych kompetencji komunikacyjnych w obliczu konfliktu.
	Negocjacje - strategie negocjacyjne, jak komunikować swoje oczekiwania, na czym polegają negocjacje, które można stosować w branży IT.
	Efektywne zarządzania czasem oraz skuteczna organizacja pracy własnej osoby, jak i w odniesieniu do zespołów IT. Optymalizacja działań w czasie w perspektywie zarządzania zwinnego.
Matematyka dyskretna	Rachunek zdań, zbiorów, kwantyfikatorów
	Analiza tautologii, rozwiązywanie zadań z wykorzystaniem metody zero-jedynkowej, wykorzystanie praw de Morgana i praw rozdzielności w rachunku zbiorów, negowanie zdań zapisanych z użyciem kwantyfikatorów
	Iloczyn kartezjański zbiorów, relacja porządku, relacja równoważności
	Rozwiązywanie zadań dotyczących iloczynu kartezjańskiego zbiorów, rysowanie diagramu Hassego, wyznaczanie klas abstrakcji
	Kongruencje, zasada indukcji matematycznej
	Rozwiązywanie układów kongruencji (chińskie twierdzenie o resztach), dowodzenie twierdzeń z wykorzystaniem zasady indukcji, dowodzenie twierdzeń metodą nie wprost
	Kryptosystemy asymetryczne (RSA, krzywe eliptyczne)
	Rozwiązywanie zadań z wykorzystaniem uogólnionego algorytmu Euklidesa, szybkiego potęgowania modulo, algorytmu RSA. Arytmetyka modularna punktów na krzywej eliptycznej.
Matematyka I	Pojęcie macierzy. Działania na macierzach. Operacje elementarne na macierzach. Zapis macierzowy układu równań.
	Rozwiązywanie zadań z zakresu analizy funkcji jednej zmiennej.
	Pojęcie ciągu liczbowego. Ciąg liczbowy jako funkcja jednej zmiennej. Monotoniczność i granica ciągu. Liczba Eulera.
	Badanie monotoniczności ciągów i obliczanie granic - rozwiązywanie zadań.
	Granice właściwe i niewłaściwe funkcji.
	Obliczanie granic funkcji - ćwiczenia praktyczne.
	Pojęcie granicy jednostronnej. Ciągłość i asymptoty funkcji.
	Aspekt praktyczny granic jednostronnych funkcji. Badanie ciągłości i wyznaczanie asymptot funkcji.
	Działania na macierzach - ćwiczenia praktyczne. Wykorzystywanie własności działań na macierzach w celu uproszczenia obliczeń.
	Pojęcie i metody obliczania wyznacznika macierzy kwadratowej. Wykorzystywanie wyznacznika do rozwiązywania układów równań liniowych.
	Obliczanie wyznaczników macierzy kwadratowych różnych stopni - ćwiczenia. Wykorzystywanie własności wyznaczników w celu uproszczenia obliczeń. Rozwiązywanie układów równań liniowych za pomocą wzorów Cramera.
	Definicja i metody wyznaczania macierzy odwrotnej. Metoda odwracania macierzy w rozwiązywaniu równań macierzowych i układów równań liniowych.
	Rozwiązywanie zadań rozwijających umiejętność wyznaczania macierzy odwrotnej różnymi metodami. Praktyczne zastosowanie macierzy odwrotnej w rozwiązywaniu równań macierzowych,
	Układy równań i nierówności liniowych w opisie problemów decyzyjnych. Zastosowanie operacji elementarnych w rozwiązywaniu układów równań.

	<p>Rozwiązywanie układów równań i nierówności liniowych metodą operacji elementarnych - studia przypadków i zastosowania praktyczne w teorii podejmowania decyzji.</p> <p>Funkcja jednej zmiennej - definicja, własności, zastosowania w badaniach ekonomicznych. Przegląd funkcji elementarnych.</p>
Matematyka II	<p>Pochodna funkcji jednej zmiennej. Pochodna jako granica ilorazu różnicowego, wzory i reguły różniczkowania.</p> <p>Ćwiczenia utrwalające umiejętność obliczania całek nieoznaczonych.</p> <p>Całka oznaczona Riemanna. Różne interpretacje całki oznaczonej. Całka jako pole powierzchni.</p> <p>Praktyczne zastosowania całki oznaczonej - rozwiązywanie zadań.</p> <p>Pojęcie funkcji wielu zmiennych. Dziedzina, pochodne cząstkowe pierwszego i drugiego rzędu.</p> <p>Ćwiczenia utrwalające umiejętność wyznaczania dziedziny i obliczania pochodnych funkcji wielu zmiennych.</p> <p>Ekstrema lokalne i warunkowe funkcji wielu zmiennych.</p> <p>Rachunek ekstremów funkcji wielu zmiennych - rozwiązywanie zadań, w tym dotyczących rozwiązywania problemów decyzyjnych.</p> <p>Praktyczne ćwiczenia z zakresu obliczania pochodnej funkcji jednej zmiennej.</p> <p>Ekonomiczna i geometryczna interpretacja pochodnej.</p> <p>Rozwiązywanie zadań z wykorzystaniem różnych interpretacji pochodnej funkcji. Interpretacja wyników.</p> <p>Monotoniczność, ekstrema lokalne i globalne funkcji jednej zmiennej. .</p> <p>Zastosowania monotoniczności i rachunku ekstremów funkcji w rozwiązywaniu problemów decyzyjnych.</p> <p>Badanie przebiegu zmienności funkcji.</p> <p>Badanie przebiegu zmienności funkcji w ujęciu praktycznym - rozwiązywanie zadań.</p> <p>Definicja funkcji pierwotnej i całki nieoznaczonej. Podstawowe wzory i reguły rachunku całkowego. Metody wyznaczania funkcji pierwotnych.</p>
MVP – prototypowanie aplikacji z wykorzystaniem AI	<p>Autonomiczna jazda, poziomy autonomiczności, budowa współczesnego samochodu, czujniki i urządzenia wchodzące w skład pojazdów autonomicznych (LIDAR, RADAR, GNSS, IMU, rejestracja obrazu), przetwarzanie danych z otoczenia, problemy z przetwarzaniem sygnałów i filtr Kalmana, sieci neuronowe: perceptron jedno- i wielowarstwowy, konwolucyjne sieci neuronowe, semantyczna segmentacja, sterowanie predykcyjne, testowanie autonomicznych pojazdów.</p> <p>Interfejsy komunikacyjne w motoryzacji, przegląd interfejsów (zastosowanie, historia standardu, opis warstwy fizycznej, opis warstwy łącza danych, model komunikacji, koncepcje budowy węzła, topologie sieci): LIN, CAN (ponadto opis standardu CAN Flexible Data, warstwa aplikacji), FlexRay, MOST, Ethernet</p> <p>Wprowadzenie do AutoSAR, motywacja do powstania standardu, architektura systemu zgodnego ze standardem AutoSAR, realizacja systemu zgodnego ze standardem AutoSAR, założenia normy ISO26262, przedstawienie procesu V-Model, standardu MISRA</p>
Narzędzia do automatyzacji budowy oprogramowania	<p>Praktyczne wprowadzenie do systemu kontroli wersji jako sposobu zarządzania kodem i jego zmianami na przykładzie GITa i GITHUBa.</p> <p>Wprowadzenie do dobrych praktyk podczas pracy/tworzenia własnego repozytorium z kodem aplikacji (README, Makefile, analiza statyczna kodu, testy, korzystanie z zewnętrznych bibliotek).</p>

	Praktyczne wprowadzenie do ciągłej integracji na przykładzie narzędzi GIT/GITHUB oraz CircleCI.
	Praktyczne wprowadzenie do ciągłego dostarczania na przykładzie narzędzi GIT/GITHUB, CircleCI oraz Docker.
	Zaawansowane wprowadzenie do ciągłego dostarczania (ang. continuous delivery) na przykładzie narzędzi GIT/GITHUB, CircleCI oraz Docker/Dockerhub.
Platformy Apple	Tryby serwisowe i recoveryOS.
	Diagnozowanie problemów z aplikacjami.
	Sieci – konfiguracja.
	Sieci – udostępnianie.
	Peryferia.
	Apple Business Manager.
	MDM i zarządzanie.
	Bezpieczeństwo.
	Diagnozowanie z wykorzystaniem trybów serwisowych i recoveryOS.
	Konfigurowanie mechanizmów dostępności oraz podstawowych elementów środowiska.
	Tworzenie skryptów powłoki oraz AppleScript.
	Dostępność i podstawy środowiska graficznego.
	Tworzenie i praca z kontami użytkowników i środowiskami multi-user.
	Konfigurowanie FileVault i wykorzystanie różnych systemów plików.
	Praca z różnymi typami danych i archiwów.
	Konfigurowanie mechanizmów Spotlight, QuickLook i App Extensions.
	Przygotowywanie kopii zapasowych.
	Konfigurowanie aplikacji oraz ustawień i danych użytkownika
	Ocena i usuwanie problemów z aplikacjami.
	Konfigurowanie elementów związanych z nawiązywaniem i utrzymywaniem połączeń sieciowych.
	Konfigurowanie mechanizmów udostępniania zasobów.
	Diagnozowanie problemów z peryferiami i ich konfiguracja.
	Linia poleceń i skryptowanie.
	Konfigurowanie Apple Business Manager.
	Konfigurowanie MDM i przygotowywanie profili konfiguracyjnych.
	Konfigurowanie mechanizmów odpowiedzialnych za bezpieczeństwo platformy.
	Konta użytkowników i multi-user.
Systemy plików i FileVault.	

	Archiwa, typy danych i domena użytkownika.
	Spotlight, QuickLook.
	Kopie zapasowe.
	Zarządzanie aplikacjami oraz danymi użytkownika.
Praktyka zawodowa I	<p>1. Podstawy prawne i przedmiot działalności przedsiębiorstwa - status prawny, struktura własnościowa, przedmiot i zakres działalności przedsiębiorstwa, misja przedsiębiorstwa, strategia przedsiębiorstwa</p> <p>2. Organizacja podmiotu gospodarczego - struktura organizacyjna, uprawnienia decyzyjne i zakres odpowiedzialności</p> <p>3. Dokumentacja organizacyjna działu IT - regulaminy, instrukcje, zasady obiegu dokumentów</p> <p>4. Infrastruktura przedsiębiorstwa - ze szczególnym naciskiem na infrastrukturę informatyczną, ale także infrastruktura produkcyjna, infrastruktura sprzedażowo-usługowa, infrastruktura marketingowa, infrastruktura finansowa</p> <p>5. Organizacja działu IT w przedsiębiorstwie - funkcje/zadania działu, struktura organizacyjna i odpowiedzialność pracowników, zarządzanie wiedzą, usługami, użytkownikami, infrastrukturą; wykorzystywane oprogramowanie, zarządzanie dokumentami; współpraca z zewnętrznymi systemami IT; udział w projektach informatycznych.</p>
Programowanie aplikacji low-code/no-code	<p>Wprowadzenie do narzędzi Low Code/No Code</p> <p>Zasady projektowania aplikacji w technologii Low Code/No Code</p> <p>Wprowadzenie do Microsoft Power Apps, omówienie typów aplikacji</p> <p>Tworzenie bazy danych dla aplikacji z wykorzystaniem MS Excel</p> <p>Generowanie aplikacji na podstawie źródła danych</p> <p>Konfiguracja wyglądu aplikacji, tworzenie/modyfikacja interfejsu użytkownika</p> <p>Tworzenie i konfiguracja wyszukiwarek danych</p> <p>Testowanie i udostępnianie przygotowanej aplikacji</p>
Programowanie obiektowe w języku Python	<p>Istota obiektowości, Odpowiedzialność. Narzędzia programowania obiektowego. Język obiektowy, a strukturalny.</p> <p>Metody działające na obiektach klasy. Wykorzystanie struktur wewnątrz klasy.</p> <p>Abstrakcja w tworzeniu klas. Metody abstrakcyjne.</p> <p>Hermetyzacja i dziedziczenie w Pythonie.</p> <p>Budowa klasy. tworzenie obiektów, konstruktor, przeciążenie metod i konstruktorów. Komunikaty. Dziedziczenie, Metoda super. Znaczenie słów; static, final. Odwołanie do pól static oraz instance. Nadpisanie metod, a przeciążenie metod. Modyfikatory dostępu i zasięg zmiennych. Zmienne lokalne, a zmienne globalne. Promocja typów.</p> <p>Enum, metoda equals, hermetyzacja danych. Gettery i settery. Metoda String toString.</p> <p>Abstrakcja. Klasy abstrakcyjne, Metody abstrakcyjne.</p> <p>Interfejsy, Implementacja Interfejsów, Polimorfizm.</p> <p>Asocjacja, agregacja, kompozycja. Ich graficzna reprezentacja.</p> <p>Istota wzorców projektowych. Podział wzorców. Implementacja wzorców projektowych – w szczególności: Singleton oraz Adapter</p>

	Podział programu na funkcje, pakiety i biblioteki.
	Wprowadzenie do klas. Opis atrybutów i tworzenie obiektów.
Programowanie w języku Python	Wprowadzenie do architektury systemów oprogramowania. Wprowadzenie do języka Python – omówienie języka, pierwszy program.
	Pakiety i importy. Instrukcja for each (enanced for). Wyjątki – obsługa wyjątków, zgłaszanie wyjątków. Opcjonalnie dodatkowo: wprowadzenie do podstawowych struktur danych.
	Literały i zmienne. Pojęcie typu. Typy proste. Typy zmiennych.
	Deklaracje. Inicjalizacja. Operatory i wyrażenia. Rzutowanie typów (zawężające i rozszerzające).
	Inkrementacja oraz dekrementacja (prefiksowa, postfiksowa). Konwersje arytmetyczne. Dzielenie całkowite i reszta
	Przegląd instrukcji sterujących. Operatory i wyrażenia porównania. Operatory i wyrażenia logiczne.
	Pojęcie pętli iteracyjnej. Warunkowe pętle iteracyjne. Pętle iteracyjne o danej liczbie powtórzeń.
	Praktyczna znajomość klas i bibliotek. Praca z plikiem tekstowym.
	Deklaracja oraz tworzenie tablic (jednowymiarowych i dwuwymiarowych).
	Implementacja algorytmów z użyciem tablic.
	Literały i zmienne. Pojęcie typu. Typy proste. Typy zmiennych. Deklaracje. Inicjalizacja. Operatory i wyrażenia. Rzutowanie typów (zawężające i rozszerzające). Inkrementacja oraz dekrementacja (prefiksowa, postfiksowa). Konwersje arytmetyczne. Dzielenie całkowite i reszta.
	Przegląd instrukcji sterujących. Operatory i wyrażenia porównania. Operatory i wyrażenia logiczne. Podejmowanie decyzji: instrukcje if oraz if-else. Switch. Operator warunkowy?:
	Pojęcie pętli iteracyjnej. Warunkowe pętle iteracyjne. Pętle iteracyjne o danej liczbie powtórzeń. Przerywanie i kontynuowanie pętli. Iteracja, a rekurencja (Silnia, Ciąg Fibbonacciego)
	Działania na łańcuchach znaków. Definiowanie klas. Do czego służą klasy? Definiowanie pól. Definiowanie metod. Modyfikatory dostępu. Hermetyzacja danych. Gettery oraz Settery. Definiowanie konstruktorów. Składowe statyczne. Przeciążanie metod i konstruktorów.
	Wprowadzenie do świata obiektów. Składowe statyczne (class), a składowe instance. Zastosowanie słowa final. Zmienne lokalne, a globalne. Czas życia zmiennych. Wprowadzenie do dziedziczenia.
	Praktyczna znajomość klas i bibliotek. Wyprowadzanie napisów i liczb, komunikaty. Praca z plikiem tekstowym.
	Deklaracja oraz tworzenie tablic (jednowymiarowych i dwuwymiarowych). Podstawowe operacje na tablicach.
Implementacja algorytmów z użyciem tablic np. sortowanie bąbelkowe, znajdowanie najmniejszego elementu w zadanej kolumnie/wierszu, sortowanie przez wstawianie. Wprowadzenie do algorytmów i ich złożoności.	
Projektowanie i tworzenie stron WWW	Wstęp do projektowania stron WWW. Omówienie środowiska pracy (Figma) i prezentacja narzędzi. Wybrane elementy layoutu - przycisków.
	Zasady tworzenia i realizacja formularzy wykorzystujących różne stany komponentów - pola tekstowe, pole drop down menu.
	Hierarchia komunikatów i podstawy typografii oraz kompozycji na przykładzie urządzenia mobilnego.
	Tworzenie layoutu strony mobilnej z wykorzystaniem zdjęć i tekstów.

	<p>Tworzenie struktury dokumentu HTML5. Ćwiczenia z użycia różnych znaczników HTML5 do budowy strony internetowej.</p> <p>Tworzenie stylów i zastosowanie właściwości CSS do formatowania tekstu i układu strony.</p> <p>Implementacja zaawansowanych technik CSS, takich jak flexbox i grid. Tworzenie responsywnych układów stron przy użyciu media queries.</p> <p>Manipulacja elementami DOM za pomocą JavaScript i jQuery. Dodawanie interaktywności do stron internetowych przy użyciu zdarzeń i animacji.</p>
Systemy baz danych	<p>Systemy bazodanowe i ich zastosowanie, cechy, architektura</p> <p>Analiza zadań projektowych, określanie kolejności zadań w zakresie tworzenia bazy</p> <p>Wzorce zapytań w zakresie podstawowych klauzul, filtrowanie, projekcja, sortowanie</p> <p>Structured Query Language jako język zapytań. Podzapytania, złączenia wewnętrzne i zewnętrzne, samo-złączenia oraz anty-złączenia; funkcje agregujące oraz grupowanie; operacje na zbiorach</p> <p>Wzorce zapytań w zakresie klauzul. Podzapytania, złączenia wewnętrzne i zewnętrzne, samo-złączenia oraz anty-złączenia; funkcje agregujące oraz grupowanie; operacje na zbiorach - ćwiczenia</p> <p>Analiza zapytań DQL do wskazanej bazy danych</p> <p>Przykłady wykorzystania systemów bazodanowych. Organizacja danych w praktyce, obiekty, atrybuty</p> <p>Wprowadzenie do problematyki projektowania baz danych, model E-R – wybrane notacje - UML, Barker, Chen, encje, atrybuty, odwzorowanie na model relacyjny. Rozszerzenia obiektowe modelu E-R specjalizacja, generalizacja. Model relacyjny, algebra relacji w relacyjnym modelu danych. Metody konwersji pomiędzy modelem E-R a modelem relacyjnym</p> <p>Praktyczne modelowanie schematu baz danych</p> <p>Nomenklatura podstawowych obiektów bazodanowe w różnych terminologiach. Podstawowe pojęcia relacyjnego modelu danych: relacja, atrybut, krotka, typy danych, klucz podstawowy, klucz obcy, inne ograniczenia integralnościowe; standardy SQL; przegląd systemów zarządzania bazami danych i klientów bazodanowych. Typy danych, kluczy, związków w SQL</p> <p>Modelowanie relacyjnego schematu (definiowanie relacji, atrybutów, kluczy, związków, typy danych)</p> <p>Normalizacja baz danych. Integralność baz danych. Język SQL, konstrukcje DDL, DML, DCL. Procedury składowane, wyzwalacze. Zarządzanie bazami danych, autoryzacja i bezpieczeństwo.</p> <p>Praktyka normalizacji w bazach danych (minimalizacja redundancji danych), wprowadzanie danych</p> <p>Structured Query Language jako język zapytań. Podstawowe klauzule, filtrowanie, projekcja, sortowanie.</p>
Systemy operacyjne	<p>Definicja i funkcje systemu operacyjnego, struktura oprogramowania systemowego i jego związek ze sprzętem, klasyfikacja systemów operacyjnych. Jądro systemu operacyjnego.</p> <p>Instalacja sterowników oraz dodatkowych narzędzi diagnostycznych. Administracja i zarządzanie siecią w Windows 11.</p> <p>Zarządzanie plikami i systemem plików w Windows 11. Praca w konsoli CMD.</p> <p>Instalacja wybranej dystrybucji Linuxa (np. Ubuntu) i konfiguracja ustawień.</p> <p>Tworzenie i zarządzanie kontami użytkowników oraz grupami w Linuxie.</p>

	<p>Ćwiczenia z montowania i zarządzania systemami plików. Praktyczne zadania z użycia narzędzi do zarządzania dyskami i partycjami.</p> <p>Konfiguracja interfejsów sieciowych i zarządzanie połączeniami. Implementacja i testowanie usług sieciowych, takich jak SSH i FTP</p> <p>Implementacja zasad bezpieczeństwa w systemach Windows i Linux.</p> <p>Procesy, przerwania. Metody obsługi przerwań jedno- i wielopoziomowe Wątki.</p> <p>Systemy plików</p> <p>Użytkownicy w systemach operacyjnych</p> <p>Obsługa wejścia-wyjścia</p> <p>Zarządzanie pamięcią. Mechanizmy synchronizacji i komunikacji między procesami, semaforey. Dijkstry, sekcja krytyczna.</p> <p>Systemy wieloprocesorowe. Wirtualizacja, konteneryzacja systemów operacyjnych.</p> <p>Bezpieczeństwo systemów operacyjnych</p> <p>Instalacja i konfiguracja systemu Windows 11.</p>
Techniczny projekt nowatorski (praca przejściowa)	<p>Zasady przygotowania i pisanie projektu inżynierskiego, przedstawienie zasad edytorskich.</p> <p>Wymogi formalne i edytorskie stawiane pracom inżynierskim.</p> <p>Omówienie pracy w trybie recenzje, wyjaśnienie i implementacja wskazówek promotora.</p> <p>Metodologia pisania oraz realizacji projektów. Opracowanie koncepcji badań, zebranie materiału badawczego.</p> <p>Plagiatowanie i prawa autorskie w pisaniu pracy inżynierskiej. Odniesienia do źródeł, parafrazowanie i pozostałe techniki zapobiegania plagiatowi.</p> <p>Wyjaśnienie wymogów dopuszczenia projektu do obrony oraz omówienie przebiegu obrony.</p> <p>Omówienie sposobu interpretacji i analizy danych, prezentacji wyników.</p> <p>Wybór tematu, konstrukcja problemu inżynierskich i hipotez badawczych do rozwiązania.</p> <p>Dobór źródeł, normy, dane techniczne.</p> <p>Propozycje tematów, spisów treści projektów i planowanej zawartości.</p> <p>Opracowanie części praktycznej pracy.</p> <p>Prezentacje aktualnych postępów w realizacji prac</p> <p>Przedstawienie i podsumowanie projektu inżynierskiego</p>
Technologie sieciowe	<p>Wprowadzenie do sieci komputerowych. Definicje, historia sieci komputerowych, urządzenia sieciowe, sieci LAN, WAN, wprowadzenie do modelu OSI i TCP/IP, zapoznanie z IOS Cisco.</p> <p>Warstwa sieci - budowanie sieci z przełącznikiem i routerem, analiza działania protokołu IP, podstawowa konfiguracja routera</p> <p>Projektowanie adresacji IP w praktyce - matematyka sieci, bity, bajty, konwersja adresów IPv4 do postaci binarnej, identyfikacja adresów, sprawdzanie ścieżki za pomocą ping i traceroute, adresacja IPv6.</p> <p>Planowanie sieci - zaplanowanie adresacji w zależności od liczby hostów w poszczególnych podsieciach, uruchomienie sieci, skonfigurowanie urządzeń, routing statyczny.</p>

	Warstwa łącza danych i warstwa fizyczna - odczytywanie adresów MAC z urządzeń sieciowych, badanie tablicy ARP.
	Konfiguracja haseł dostępu, dostępu zdalnego, odzyskiwanie hasła, przechowywanie plików konfiguracyjnych na serwerze TFTP.
	Warstwa aplikacji - omówienie funkcjonalności, przegląd wybranych usług i protokołów z warstwy aplikacji. Warstwa transportowa - protokoły warstwy transportowej, segmentacja, scalanie, multipleksowanie.
	Warstwa sieci, protokoły warstwy sieci, porównanie IPv4 i IPv6, budowa adresu IPv4 i IPv6, podział sieci na podsieci.
	Tworzenie podsieci IPv4 - obliczanie adresów sieci, hosta, rozgłoszeniowego, podział sieci na podsieci techniką VLSM.
	Routing statyczny i dynamiczny - planowanie adresacji, podział sieci na podsieci techniką VLSM, uruchomienie routingu statycznego i dynamicznego, wady i zalety obu rozwiązań.
	Warstwa łącza danych i warstwa fizyczna - dostęp warstw wyższych do medium, protokoły warstwy łącza danych, zasada działania warstwy fizycznej, media transmisyjne.
	Wprowadzenie do bezpieczeństwa urządzeń sieciowych.
	Zapoznanie ze środowiskiem Packet Tracer w praktyce, zapoznanie z programem Wireshark, nawigacja w IOS. Konfigurowanie sieciowego systemu operacyjnego - konfigurowania adresu do zarządzania przełącznikiem.
	Warstwa aplikacji - konfigurowanie i analiza wybranych usług i protokołów warstwy aplikacji. Warstwa transportowa - numery portów TCP i UDP, obserwacja i analiza działania protokołów TCP i UDP, nawiązywanie i kończenie sesji TCP.
Testowanie oprogramowania	Wprowadzenie do testowania oprogramowania.
	Poziomy i typy testów.
	Techniki testowania.
	Zarządzanie jakością oprogramowania.
	Tworzenie przypadków testowych.
	Tworzenie przypadków testowych w podejściu BDD.
	Testowanie eksploracyjne.
	Raportowanie defektów.
Uczenie maszynowe	Wybrane pojęcia z dziedziny uczenia maszynowego. Obszary zastosowań. Rola danych w uczeniu maszynowym- eksploracja danych.
	Implementacja regresji liniowej i logistycznej w Python/sklearn w praktyce.
	Budowa sieci MLP w Keras/TensorFlow w praktyce.
	Eksperymenty z technikami regularyzującymi np. dropout, batch normalization w praktyce.
	Walidacja modeli, interpretacja metryk (np. precision, recall, ROC) w praktyce
	Implementacja i wizualizacja klasteryzacji w praktyce.
	Implementacja i porównanie modeli drzewiastych i ensemble w praktyce.
	Laboratorium: Implementacja i porównanie modeli gradient boosting w praktyce.
	Regresja liniowa i logistyczna.

	Płytkie wielowarstwowe sieci jednokierunkowe - architektura i zasada uczenia.
	Techniki poprawiające uczenie sieci neuronowych.
	Dobre praktyki w sieciach neuronowych (overfitting, walidacja krzyżowa, metryki, fairness)
	Algorytmy grupowania. Klasteryzacja.
	Drzewa decyzyjne i uczenie Bayesowskie. Algorytmy Ensemble
	Algorytmy gradient boosting np. XGBoost, LightGBM, CatBoost.
	Konfiguracja środowiska do prowadzenia badań (scikit-learn, keras, matplotlib).
Współczesne systemy telekomunikacyjne	Historia telekomunikacji
	Wprowadzenie do współczesnych systemów telekomunikacyjnych
	Bluetooth i WiFi, a systemy telekomunikacyjne
	System 4G – porównanie części radiowej (RAN) systemów 3G i 4G, zagadnienia radiowe
	System 4G – część corowa sieci, architektura sieci 4G, usługi niezbędne do działania sieci
	System 5G – część radiowa (RAN), podobieństwa i różnice między systemami 4G i 5G
	Sieć 5G – zagadnienia i przypadki użycia częstotliwości FR2 (wysokie częstotliwości, fale milimetrowe)
	Nowe trendy w telekomunikacji. System 5G z perspektywy użytkownika końcowego.
Wykorzystanie sztucznej inteligencji w IT	Wybrane pojęcia z dziedziny sztucznej inteligencji. Obszary zastosowań.
	Implementacja wybranego algorytmu inteligencji stadnej do rozwiązania problemu optymalizacyjnego.
	Przegląd praktycznych zastosowań sztucznej inteligencji.
	Mapy zastosowań AI w różnych branżach związanych z IT.
	Aspekty prawne i etyczne wykorzystania sztucznej inteligencji.
	Analiza dylematów etycznych związanych z wykorzystaniem sztucznej inteligencji w IT.
	Obszary zastosowań sztucznej inteligencji w IT.
	Problem wyszukiwania najkrótszej ścieżki w grafie i algorytmy sztucznej inteligencji rozwiązujące ten problem.
	Implementacja wybranego algorytmu do rozwiązania problemu komiwojażera.
	Logika rozmyta, projekt sterownika rozmytego.
	Projekt sterownika rozmytego dla przykładowego systemu sterowania.
	Algorytmy genetyczne w rozwiązywaniu problemów optymalizacyjnych.
	Implementacja podstawowego algorytmu genetycznego do rozwiązania problemu plecakowego.
	Algorytmy inteligencji stadnej.
	Tworzenie i zastosowanie dekreatorów dla funkcji.

Zaawansowane techniki programowania w języku Python	Tworzenie i zastosowanie dekoatorów dla metod.
	Omówienie iteratorów.
	Omówienie generatorów, wyrażeń generujących.
	Wyrażenia regularne.
	Praca z wieloma wątkami.
	Praca z wieloma procesami.
	Omówienie Global Interpreter Lock (GIL).
Zarządzanie projektami informatycznymi	Wprowadzenie do zarządzania projektami IT.
	Planowanie projektu informatycznego (zakres, cele, harmonogram, identyfikacja interesariuszy).
	Zarządzanie zadaniami i czasem w projekcie IT. Zapoznanie z narzędziami wspierającymi zarządzaniem projektami (np. GitHub, Jira).
	Zarządzanie komunikacją, ryzykiem i zmianą.
	Zwinne metodyki zarządzania projektami IT.
	Tworzenie dokumentacji projektowej.
	Tworzenie raportu końcowego i prezentacja projektu.
<b>Specjalność: AI &amp; Data Science</b>	
Modelowanie, integracja i eksploracja danych – Data Mining	Proces odkrywania wiedzy z danych.
	Zaawansowane techniki przygotowania danych do analizy.
	Analiza asocjacji i odkrywanie reguł.
	Zaawansowane metody klasyfikacji w data mining.
	Analiza skupień w złożonych zbiorach danych.
	Wykrywanie anomalii i outlierów.
	Modelowanie sekwencji i analiza szeregów czasowych.
	Integracja heterogenicznych źródeł danych w procesie data mining.
	Ćwiczenia z identyfikacji etapów odkrywania wiedzy w projektach data mining.
	Praktyczne czyszczenie i transformacja danych.
	Implementacja algorytmu Apriori i FP-Growth. Analiza wyników i interpretacja reguł asocjacyjnych.
	Implementacja i porównanie algorytmów klasyfikacyjnych.
	Zastosowanie algorytmów klastrowania oraz ocena jakości klastrów i wizualizacja wyników.
	Implementacja metod wykrywania anomalii (np. Isolation Forest, LOF).
	Prognozowanie i analiza szeregów czasowych.
Praktyczne ćwiczenia w integracji danych i analizie wyników.	
Pozyskiwanie i gromadzenie danych	Praktyczne ćwiczenia w identyfikacji źródeł danych. Uzyskiwanie dostępu do różnych typów danych
	Tworzenie skryptów do web scrapingu. Automatyzacja pobierania danych z sieci.
	Implementacja wybranego systemu gromadzenia danych.
	Techniki selekcji i poprawy jakości danych.
	Praktyczne ćwiczenia w integracji danych.

	<p>Implementacja aplikacji przetwarzających dane w czasie rzeczywistym. Analiza strumieni danych.</p> <p>Analiza przypadków dotyczących prawnych i etycznych aspektów pozyskiwania danych.</p> <p>Implementacja systemów zabezpieczeń i prywatności w procesie gromadzenia danych.</p> <p>Bezpieczeństwo i prywatność w procesie gromadzenia danych.</p> <p>Strategie i metody pozyskiwania danych.</p> <p>Web scraping i automatyzacja zbierania danych online.</p> <p>Projektowanie systemów gromadzenia danych.</p> <p>Zarządzanie jakością danych w procesie ich pozyskiwania.</p> <p>Integracja danych z różnych źródeł.</p> <p>Przetwarzanie strumieni danych w czasie rzeczywistym.</p> <p>Prawne i etyczne aspekty pozyskiwania danych.</p>
Praktyczne uczenie maszynowe	<p>Instalacja i konfiguracja środowiska Python z bibliotekami do uczenia maszynowego.</p> <p>Implementacja i ocena regresji liniowej i logistycznej przy użyciu scikit-learn.</p> <p>Tworzenie i trenowanie oraz ocena modeli drzew decyzyjnych i lasów losowych.</p> <p>Implementacja oraz ocena maszyn wektorów nośnych (SVM) dla klasyfikacji.</p> <p>Implementacja oraz ocena algorytmów klastrowania, takich jak k-means i DBSCAN.</p> <p>Implementacja oraz ocena technik ensemble learning, takich jak bagging i boosting oraz metod zespołowych.</p> <p>Implementacja i porównanie algorytmów XGBoost, LightGBM, i CatBoost.</p> <p>Implementacja technik wyjaśnialnej sztucznej inteligencji w zadaniach klasyfikacji.</p>
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Praktyczne zastosowania Data Science w analizie danych biznesowych i przemysłowych:</p> <ul style="list-style-type: none"> <li>• Analizuje zbiory danych biznesowych i/lub technicznych z wykorzystaniem narzędzi analitycznych oraz środowisk programistycznych</li> <li>• Tworzy i ocenia modele statystyczne i/lub predykcyjne (np. regresja, drzewa decyzyjne, clustering) odpowiednie do rozwiązywanych problemów operacyjnych lub biznesowych.</li> <li>• Przygotowuje i przetwarza dane pochodzące z różnych źródeł (bazy danych, API, pliki CSV/Excel), w tym na potrzeby dalszej analizy</li> <li>• Stosuje techniki analizy, eksploracji i wizualizacji danych celem identyfikacji zależności, trendów i anomalii</li> <li>• Wyszukuje, pozyskuje, ocenia, analizuje i zestawia informacje oraz dokumentację branżową stosownie do realizowanego zadania</li> <li>• Opracowuje raporty i dashboard wspomagające proces podejmowania decyzji</li> <li>• Opracowuje dokumentację projektową na temat bieżącego zadania.</li> </ul> <p>MODUŁ 2: Uczenie maszynowe i przetwarzanie danych na potrzeby projektów analitycznych i AI</p> <ul style="list-style-type: none"> <li>• Przygotowuje dane do dalszego modelowania i eksploracji poprzez czyszczenie, transformację, standaryzację i inżynierię cech</li> <li>• Tworzy, trenuje i testuje modele uczenia maszynowego (np. klasyfikacja, regresja, detekcja anomalii) z wykorzystaniem bibliotek takich jak scikit-learn, TensorFlow lub PyTorch.</li> <li>• Stosuje metody walidacji modeli oraz interpretuje metryki jakości dla konkretnych zastosowań.</li> <li>• Implementuje rozwiązania AI w postaci prototypów lub prostych integracji (np. API, notebook, pipeline)</li> <li>• Analizuje i porównuje różne podejścia modelowe w celu wyboru rozwiązania</li> </ul>

	<p>optymalnego.</p> <ul style="list-style-type: none"> <li>• Pozyskuje, analizuje i wykorzystuje branżowe informacje i odpowiednią dokumentację oraz stosuje dobre praktyki i odpowiednie normy, w szczególności dotyczące algorytmów i narzędzi AI</li> <li>• Przygotowuje dokumentację techniczną zrealizowanego zadania, uwzględniającą przebieg prac oraz uzyskane wyniki</li> </ul> <p>- zajęcia terenowe,</p>
Proces uczenia się w SI	Praktyczne ćwiczenia w środowisku Anaconda.
	Eksperymenty z różnymi sposobami reprezentacji danych.
	Eksperymenty z różnymi strategiami uczenia.
	Eksperymenty z różnymi modelami w celu zrozumienia problemu overfittingu.
	Implementacja transfer learning z użyciem pretrenowanych modeli.
	Praktyczne zastosowanie narzędzi AutoML do automatyzacji procesu uczenia.
	Implementacja algorytmów uczenia przyrostowego.
	Techniki oceny modeli i interpretacji wyników. Wizualizacja wyników i analiza błędów.
Systemy rozmyte i ekspertowe	Wprowadzenie do systemów rozmytych i systemów ekspertowych.
	Projektowanie zbiorów rozmytych i funkcji przynależności.
	Systemy wnioskowania rozmytego (FIS, Mamdani, Sugeno).
	Systemy ekspertowe oparte na regułach.
	Reprezentacja niepewności w systemach ekspertowych.
	Integracja systemów rozmytych z technikami uczenia maszynowego.
	Zastosowania systemów rozmytych w sterowaniu i podejmowaniu decyzji.
	Ewaluacja i walidacja systemów ekspertowych.
	Implementacja prostych systemów ekspertowych
	Tworzenie i wizualizacja zbiorów rozmytych.
	Implementacja systemów wnioskowania typu Mamdani.
	Tworzenie bazy reguł dla systemów ekspertowych. Implementacja i testowanie systemów opartych na regułach.
	Techniki reprezentacji niepewności w systemach ekspertowych. Ćwiczenia z wykorzystaniem teorii możliwości i prawdopodobieństwa.
	Praktyczne ćwiczenia z integracji metod rozmytych i uczenia maszynowego.
	Implementacja systemów rozmytych w aplikacjach typu systemy ekspertowe i wspomaganie decyzji.
Praktyczna ewaluacja i walidacja systemów ekspertowych i wspomaganie decyzji.	
Uczenie głębokie	Projektowanie głębokich sieci neuronowych.
	Implementacja sieci CNN.
	Implementacja sieci RNN oraz LSTM.
	Implementacja autoenkoderów.
	Implementacja sieci typu transformer.
	Implementacja sieci GAN.
	Praktyczne wykorzystanie transfer learning w głębokich sieciach neuronowych.
	Wdrażanie technik optymalizacji i regularyzacji w celu zwiększenia wydajności modeli głębokich.
	Architektura głębokich sieci neuronowych.
	Konwolucyjne sieci neuronowe (CNN) i ich zastosowania.
	Rekurencyjne sieci neuronowe (RNN) i LSTM.

	Autoenkodery i uczenie nienadzorowane w sieciach głębokich.
	Transformery oraz LLM.
	Generative Adversarial Networks (GAN).
	Transfer learning w głębokich sieciach neuronowych.
	Optymalizacja i regularyzacja w uczeniu głębokim.
Widzenie komputerowe	Wprowadzenie do przetwarzania obrazów cyfrowych.
	Segmentacja obrazu i detekcja krawędzi.
	Rozpoznawanie obiektów i klasyfikacja obrazów.
	Głębokie uczenie w widzeniu komputerowym. Sieci CNN oraz Transfer learning (np. ResNet, EfficientNet).
	Detekcja obiektów: YOLO (You Only Look Once), SSD, Faster R-CNN oraz segmentacja semantyczna i instancyjna: U-Net, Mask R-CNN, DeepLabV3+.
	Vision Transformers (ViT) – architektura, działanie i zastosowanie.
	CLIP (Contrastive Language–Image Pretraining) – łączenie tekstu i obrazu. SAM (Segment Anything Model), Diffusion Models i generowanie obrazów.
	Zastosowania praktyczne: Rozpoznawanie twarzy i emocji. OCR – rozpoznawanie tekstu (Tesseract, CRNN). Widzenie maszynowe w robotyce i medycynie.
	Implementacja i analiza algorytmów przetwarzania obrazów.
	Implementacja technik segmentacji obrazu, takich jak segmentacja progowa i klastrowanie.
	Implementacja algorytmów rozpoznawania obiektów, takich jak SIFT i SURF.
	Implementacja sieci CNN oraz Transfer learning w zadaniach praktycznych.
	Implementacja algorytmów detekcji obiektów w zadaniach praktycznych analizy video.
	Implementacja algorytmów ViT w zadaniach praktycznych.
Praca z systemami generowania obrazów.	
Implementacja algorytmów rozpoznawania twarzy i emocji oraz OCR w zadaniu rozpoznawania tekstu.	
Wizualizacja i raportowanie danych	Wybrane elementy percepcji wizualnej i projektowania informacji. Ćwiczenia z analizy percepcji wizualnej. Projektowanie efektywnych wizualizacji informacji.
	Zaawansowane techniki wizualizacji danych wielowymiarowych. Implementacja technik wizualizacji danych wielowymiarowych.
	Tworzenie raportów dynamicznych i interaktywnych. Projektowanie dynamicznych raportów.
	Interaktywne dashboardy i storytelling z danymi. Projektowanie i implementacja interaktywnych dashboardów.
	Wizualizacja danych geograficznych i czasoprzestrzennych. Tworzenie map i wizualizacji danych geograficznych.
	Wizualizacja big data i strumieni danych w czasie rzeczywistym. Techniki wizualizacji dużych zbiorów danych.
	Narzędzia i biblioteki do zaawansowanej wizualizacji danych. Praktyczne zastosowanie narzędzi i bibliotek do wizualizacji.
	Projektowanie systemów raportowania biznesowego oraz raportów biznesowych. Implementacja raportów biznesowych i analiza wyników.
<b>Specjalność: AI w grafice komputerowej i technologiach multimedialnych</b>	
Grafika 3D	Wizualizacje 3D - cechy, projektowanie, zastosowanie.
	Oprogramowanie Blender Foundation. Zapoznanie z interfejsem, konfiguracją, funkcjami i możliwościami.
	Realizacja projektu wizualizacji 3D w programie Blender.

	Generowanie - renderowanie sceny projektu w bitmapę oraz jej zastosowanie np. w Mock-upie
Narzędzia AI w kampaniach marketingowych	Wprowadzenie do narzędzi AI w marketingu
	Generatory tekstów reklamowych i blogowych z wykorzystaniem AI
	Narzędzia do personalizacji treści marketingowych
	AI do analizy sentymentu w treściach marketingowych
	Narzędzia AI do generowania i edytowania grafik
	Generowanie animacji i wideo marketingowego z pomocą AI
	Automatyzacja kampanii marketingowych na mediach społecznościowych
	AI w e-mail marketingu
	Chatboty i systemy automatycznej obsługi klienta
	Narzędzia do analizy danych marketingowych i raportowania
	AI w optymalizacji reklam w mediach społecznościowych
	Automatyzacja tworzenia treści wideo i generowanie voice-overów
	Wykrywanie fałszywego ruchu i oszustw reklamowych
	Narzędzia AI do segmentacji i targetowania odbiorców w kampaniach
	Narzędzia do analizy konkurencji i trendów rynkowych
Integracja narzędzi AI w kompleksową strategię marketingową	
Obróbka fotografii	Obsługa funkcji cyfrowego aparatu fotograficznego oraz elementów wyposażenia studia.
	Realizacja fotograficznego projektu plenerowego "Głębia".
	Realizacja fotograficznego projektu studyjnego "Portret/Produkt".
	Przygotowanie wybranej fotografii do dalszej eksploatacji za pomocą programów z zakresu obróbki zdjęć Adobe Photoshop, Adobe Lightroom.
	Dokumentowanie wykonanego projektu z uwzględnieniem pola eksploatacji.
Obsługa programów graficznych	Przygotowanie materiałów graficznych do druku.
	Różne style graficzne. Tworzenie projektu wizytówki.
	Narzędzia w programach graficznych budujące elementy projektu graficznego.
	Tworzenie grafik (antonimów) przy pomocy linii, kształtów, kolorów, typografii, tekstury i przestrzeni.
	Zasady kompozycji na przykładzie wybranych projektów graficznych, systemu siatek, tworzenia layoutów. Cechy dobrego projektu graficznego.
	Tworzenie layoutów. Projekt ulotki składanej.
	Projektowanie logo, kategoryzacja znaków, metody tworzenia, zasady funkcjonowania w grafice wektorowej.
	Projektowanie spójnej oprawy graficznej dla wydarzenia kulturalnego lub sportowego: znak, plakat, bilet, okładka płyty DVD, baner internetowy.
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Projektowanie i implementacja rozwiązań AI w grafice komputerowej</p> <p>Projektuje i implementuje algorytmy przetwarzania obrazu, animacji i grafiki komputerowej z wykorzystaniem metod sztucznej inteligencji.</p> <p>Tworzy modele do analizy, przekształcania i generowania treści wizualnych (np. segmentacja, detekcja, generowanie sekwencji animacji).</p> <p>Integruje rozwiązania AI z istniejącymi systemami graficznymi oraz środowiskami do tworzenia treści wizualnych.</p> <p>Ocenia działanie zastosowanych metod (wydajność, jakość wyników, ograniczenia).</p> <p>Dokumentuje przebieg prac zgodnie z dobrymi praktykami oraz przygotowuje raport techniczny z realizowanego zadania.</p> <p>Wyszukuje i analizuje informacje oraz dokumentację branżową potrzebną do</p>

	<p>realizacji projektu..</p> <p>MODUŁ 2: Inteligentne systemy multimedialne i interakcja człowiek-komputer</p> <p>Tworzy i wdraża interaktywne aplikacje multimedialne wykorzystujące sztuczną inteligencję (np. rozpoznawanie gestów, mimiki, emocji):</p> <p>Projektuje i implementuje rozwiązania multimedialne wykorzystujące sztuczną inteligencję do przetwarzania obrazu, dźwięku, mowy i sygnałów użytkownika.</p> <p>Tworzy interaktywne systemy, w których AI wspiera analizę zachowań, interpretację sygnałów oraz adaptację interfejsu do użytkownika.</p> <p>Integruje algorytmy AI z aplikacjami multimedialnymi, systemami wizualizacji oraz interfejsami VR/AR.</p> <p>Testuje działanie opracowanych rozwiązań, analizuje ich funkcjonalność i formułuje wnioski dotyczące ulepszeń.</p> <p>Stosuje zasady projektowania doświadczeń użytkownika (UX) w kontekście multimedii wspieranych AI.</p> <p>Dokumentuje wykonane prace oraz sporządza raport podsumowujący proces projektowy i wyniki testów.</p> <p>Wyszukuje i ocenia źródła branżowe niezbędne do realizacji projektu.</p> <p>- zajęcia terenowe</p>
Projektowanie graficzne z elementami brandingu	Wprowadzenie do projektowania graficznego z wykorzystaniem branding.
	Przygotowanie do projektowania identyfikacji wizualnej marki - szkic logotypu.
	Projekt logotypu.
	Projekt księgi znaku marki.
	Projekt grafik promocyjnych z zastosowaniem logotypu w przestrzeni cyfrowej
Projekt grafik promocyjnych z zastosowaniem logotypu w publikacji drukowanej.	
Projektowanie opakowań i wzorów przemysłowych	Przygotowanie koncepcji graficznej i funkcjonalnej wzoru przemysłowego.
	Projekt graficzny opakowania zbiorczego z elementami dla spedytorów.
	Projekt graficzny opakowania jednostkowego, kartonowego.
	Przygotowanie projektów opakowań do druku oraz prezentacja 3D opakowań i wzoru.
	Projekt opakowania szklanego.
	Realizacja wzoru przemysłowego za pomocą modelowania 3D, oraz dokumentacji technicznej.
Wprowadzenie do projektowania opakowań i wzornictwa przemysłowego.	
Skład komputerowy DTP	Ogólne zasady przygotowania projektów do druku.
	Błędy w przygotowaniu plików cyfrowych do druku. Analiza błędów przygotowanych wcześniej plików kierowanych do drukarni.
	Zasady składu komputerowego. Skład wielostronicowych tekstów. Najczęściej popełniane błędy przy składzie tekstu. Analiza błędów typograficznych i nauka ich unikania.
	Typografia. Ćwiczenia typograficzne związane ze składem krótkiego tekstu.
	Druk wielkoformatowy – zasady przygotowania plików.
	Skład komputerowy (gazety, czasopisma, książki).
	Nauka przygotowania plików do druku (znaki drukarskie, spady drukarskie, wybarwienia).
	Przygotowanie plików do druku na materiałach innych niż papier.
	Przygotowanie do druku plików z uszlachetnieniem wydruku - dodatkowy kolor, lakier UV, uwypuklenia.
	Przygotowanie do druku plików z wykrojnikami.
Tworzenie stron internetowych	Landing Page - zasadność stosowania uproszczonych stron internetowych
	Mapy empatii
	Ścieżka użytkownika

	Usability - zasady użyteczności stron internetowych
	Acessability - wytyczne dostępności stron internetowych
	Architektura informacji - struktury budowy stron www
	Projekt własny strony typu - Landing Page
	Projekt własny - strona dla wybranej firmy usługowej
	Projektowanie bannerów stron internetowych
	Wykorzystanie sztucznej inteligencji do tworzenia podstawowych projektów storn internetowych
	Praktyczne ćwiczenia w stosowaniu sztucznej inteligencji do tworzenia layoutu strony
	Tworzenie wstępnego layoutu strony mobilnej z wykorzystaniem zdjęć i tekstów.
	Programy ułatwiające pracę ze sztuczną inteligencją w zakresie projektowania graficznego
	Animowanie i dynamizowanie layoutu strony internetowej
	Wprowadzenie do makietowania stron internetowych "wireframe"
	Prototypowanie i symulacja interakcji przy tworzeniu stron internetowych
	Zaawansowane techniki kompozycji i pozycjonowanie obiektów na stronach internetowych
	Projektowanie bazowych elementów strony internetowej
	Responsywność stron internetowych - wprowadzenie do tematu
	User Interface - zasady tworzenia funkcjonalnych layoutów
	User Expierience - Zasady tworzenia właściwych interakcji
	User Center Design - Zasady tworzenia projektów skoncentrowanych na potrzeby użytkownika
	Projekt podsumowujący - praca nad indywidualnie zaprojektowaną stroną internetową
	Call to action - przyciski wzywające do działania - zasady ich stosowania
	Design system i material design - systemy tworzenia baz
<b>Specjalność: Architekt rozwiązań IT w chmurze obliczeniowej (tylko forma niestacjonarna)</b>	
Aplikacje Internetu rzeczy	Wprowadzenie do systemów Intenetu Rzeczy w kontekście innych platform i języków programowania.
	Biblioteki i mechanizmy stosowane w Raspberry PI
	Implementacja funkcjonalności komunikacji pomiędzy Raspberry PI, a innymi urządzeniami.
	Wykonywanie zadań zawierających standardowe aplikacje Internetu Rzeczy.
	Wykonywanie zadań zawierających rozszerzone aplikacje Internetu Rzeczy.
	Implementowanie funkcjonalności przy użyciu czujników dostępnych w Laboratorium
	Wykonywanie zadań związanych z zastosowaniem aplikacji Internetu Rzeczy w automatyce budynkowej.
	Wykonywanie zadań związanych z zastosowaniem aplikacji Internetu Rzeczy w automatyce przemysłowej.
	Protokoły komunikacyjne używane w Raspberry PI
	Automatyka budynkowa. Systemy inteligentnych budynków.
	Automatyka przemysłowa.
Budowa architektury bezpieczeństwa IT	Wprowadzenie do budowy architektury bezpieczeństwa IT
	Podział zagrożeń na sprzętowe i software'owe
	Separacja uprawnień i rodzaje kont

	Konteneryzacja i sandboxing
	Zagrożenia sieciowe i metody ich minimalizacji
	Współczesne mechanizmy prewencji i ograniczenia ataków runtime
	Bezpieczeństwo zasobów chmurowych
	Logowanie, monitoring i analiza zdarzeń
	Przygotowanie środowiska pod kątem ograniczenia zagrożeń software'owych i sprzętowych
	Tworzenie i konfigurowanie kont oraz definiowanie uprawnień
	Zarządzanie i wykorzystanie mechanizmów konteneryzacji
	Konfigurowanie i wykorzystywanie usług sieciowych
	Konfigurowanie mechanizmów ochrony środowiska runtime
	Konfigurowanie usług chmurowych
	Konfigurowanie i uruchamianie mechanizmów logujących i monitorujących
Budowa hybrydowej infrastruktury IT	Wprowadzenie do hybrydowej architektury IT i modeli chmury.
	Usługi obliczeniowe w chmurze - IaaS, PaaS, SaaS w różnych platformach.
	Architektury chmury prywatnej - VMware vSphere, Hyper-V, OpenStack.
	Infrastructure as Code (IaC) - Terraform, CloudFormation, ARM Templates.
	Konteneryzacja i orkiestracja - Docker, Kubernetes, Docker Swarm.
	Strategie migracji do chmury - lift-and-shift, re-platforming, refactoring.
	Monitoring, bezpieczeństwo i zarządzanie tożsamością w środowisku hybrydowym.
	Multi-cloud, edge computing i optymalizacja kosztów w środowisku hybrydowym.
	Porównanie platform chmurowych (AWS, Azure, Google Cloud) - utworzenie kont i podstawowa konfiguracja.
	Budowa środowiska w wybranej platformie chmurowej z wykorzystaniem podstawowych usług obliczeniowych.
	Implementacja środowiska hybrydowego łączącego infrastrukturę lokalną z chmurą publiczną.
	Automatyzacja wdrożeń infrastruktury z wykorzystaniem wybranego narzędzia IaC.
	Budowa środowiska kontenerowego w architekturze hybrydowej - wdrożenie aplikacji na klastrze Kubernetes.
	Planowanie i wykonanie migracji przykładowej aplikacji z infrastruktury lokalnej do chmury.
Implementacja systemów monitoringu (Prometheus, Grafana) i rozwiązań bezpieczeństwa w infrastrukturze hybrydowej.	
Budowa rozwiązania multi-cloud z analizą kosztów i implementacją zasad governance.	
Infrastructure as Code	Tworzenie prostych konfiguracji w narzędziach imperatywnych i deklaratywnych (np. Bash vs Terraform).
	Tworzenie środowisk przy użyciu stosu infrastruktury (np. Terraform + AWS/Azure).
	Automatyzacja wdrożeń infrastruktury z wykorzystaniem Jenkins/Terraform.
	Wdrażanie i zarządzanie maszynami wirtualnymi oraz kontenerami w praktyce.
	Tworzenie, budowanie, uruchamianie i usuwanie obrazów i kontenerów Dockera. Uruchamianie wielu kontenerów Dockera za pomocą systemu Docker Compose.

	Tworzenie klastrów i wdrażanie aplikacji w Kubernetes (minikube, kubectl, Helm).
	Wdrażanie aplikacji w usługach bezserwerowych. Bezpieczne zmienianie infrastruktury.
	Praktyczne testowanie kodu oraz stosów infrastruktury w środowiskach chmurowych.
	Infrastruktura jako kod. Podejście imperatywne vs deklaratywne. Klasyfikacja narzędzi automatyzacji infrastruktury.
	Platformy i zasoby infrastruktury. Stos infrastruktury.
	Niezmienna integracja/ciągłe wdrażanie infrastruktury przy użyciu narzędzi Jenkins, Terraform, Ansible, Chef i Pulumi.
	Wdrażanie i zarządzanie maszynami wirtualnymi oraz kontenerami.
	Docker i Docker Compose.
	Orkiestracja kontenerów w Kubernetes.
	Technologie bezserwerowe. Środowiska wykonawcze aplikacji.
	Testowanie kodu oraz stosów infrastruktury.
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p><b>MODUŁ 1: Projektowanie architektury systemów IT w chmurze obliczeniowej</b></p> <ul style="list-style-type: none"> <li>• Analizuje wymagania funkcjonalne i нефункционалне systemów IT oraz tworzy koncepcje architektury chmurowej dopasowane do potrzeb organizacji.</li> <li>• Projektuje struktury komponentów, usługi obliczeniowe, warstwy integracji, przepływy danych i mechanizmy komunikacji w rozproszonych środowiskach chmurowych.</li> <li>• Dobiera modele usług chmurowych (IaaS, PaaS, SaaS) i modele wdrożenia (publiczna, prywatna, hybrydowa) do specyfikacji projektu.</li> <li>• Projektuje mechanizmy wysokiej dostępności, skalowalności oraz odporności systemów opartych na rozwiązaniach chmurowych.</li> <li>• Projektuje zasady bezpieczeństwa systemów w chmurze: polityki dostępu, ochronę danych, segmentację zasobów, mechanizmy monitorowania i reagowania.</li> <li>• Przygotowuje dokumentację projektową, w tym diagramy architektury, modele przepływu danych, specyfikacje komponentów i uzasadnienia projektowe.</li> <li>• Pozyskuje, analizuje i interpretuje dokumentację oraz normy branżowe, stosując je w procesie projektowania.</li> </ul> <p><b>MODUŁ 2: Wdrażanie, integracja i nadzorowanie rozwiązań chmurowych</b></p> <ul style="list-style-type: none"> <li>• Dobiera i konfiguruje środowiska chmurowe, w tym komponenty obliczeniowe, bazodanowe, sieciowe i integracyjne, zgodnie z projektem architektury.</li> <li>• Integruje systemy chmurowe z istniejącymi aplikacjami i usługami organizacji, planuje migrację danych i usług do chmury.</li> <li>• Stosuje metody automatyzacji wdrożeń i konfiguracji z użyciem infrastruktury jako kodu (IaC).</li> <li>• Konfiguruje procesy zarządzania środowiskami chmurowymi, obejmujące monitorowanie, rejestrowanie zdarzeń, kontrolę dostępu, polityki bezpieczeństwa i zarządzanie kosztami.</li> <li>• Wykonuje analizę ryzyka i wdraża mechanizmy zapewniające ciągłość działania systemów chmurowych.</li> <li>• Przeprowadza przeglądy architektoniczne i ocenia zgodność wdrożenia z projektem, wymaganiami i standardami organizacji.</li> <li>• Przygotowuje pełną dokumentację wdrożeniową, operacyjną i bezpieczeństwa.</li> <li>• Współpracuje w zespole projektowym, konsultując i uzgadniając decyzje architektoniczne z interesariuszami technicznymi i biznesowymi.</li> </ul> <p>- zajęcia terenowe,</p> <ul style="list-style-type: none"> <li>• Stosuje metody automatyzacji wdrożeń i konfiguracji z użyciem infrastruktury jako kodu (IaC).</li> <li>• Konfiguruje procesy zarządzania środowiskami chmurowymi, obejmujące monitorowanie, rejestrowanie zdarzeń, kontrolę dostępu, polityki</li> </ul>

	<p>bezpieczeństwa i zarządzanie kosztami.</p> <ul style="list-style-type: none"> <li>• Wykonuje analizę ryzyka i wdraża mechanizmy zapewniające ciągłość działania systemów chmurowych.</li> <li>• Przeprowadza przeglądy architektoniczne i ocenia zgodność wdrożenia z projektem, wymaganiami i standardami organizacji.</li> <li>• Przygotowuje pełną dokumentację wdrożeniową, operacyjną i bezpieczeństwa.</li> <li>• Współpracuje w zespole projektowym, konsultując i uzgadniając decyzje architektoniczne z interesariuszami technicznymi i biznesowymi.</li> </ul> <p>- zajęcia terenowe</p>
Projekt zespołowy	Organizacja projektu zespołowego i metodologia Scrum. Podział ról w zespole, wybór tematu projektu aplikacji chmurowej, analiza wymagań i projektowanie architektury.
	Konfiguracja środowiska deweloperskiego i infrastruktury. Wybór stosu technologicznego, utworzenie repozytoriów Git, konfiguracja CI/CD pipeline, zarządzanie projektami.
	Implementacja MVP (Minimum Viable Product). Rozwój podstawowej funkcjonalności aplikacji, integracja z usługami chmurowymi, pierwsze testy jednostkowe.
	Rozbudowa funkcjonalności i integracje. Implementacja zaawansowanych funkcji, integracja z zewnętrznymi API, system uwierzytelniania i autoryzacji.
	Testowanie, monitoring i dokumentacja. Implementacja testów automatycznych (unit, integration, e2e), konfiguracja monitoringu aplikacji, przygotowanie dokumentacji technicznej.
	Finalizacja i deployment produkcyjny. Optymalizacja wydajności, audyt bezpieczeństwa, wdrożenie na środowisko produkcyjne, przygotowanie dokumentacji końcowej.
Web Apps i aplikacje mobilne	Analiza wymagań i planowanie funkcjonalności aplikacji PWA i mobilnych – wprowadzenie do środowisk programistycznych.
	Tworzenie szkieletu aplikacji PWA (React) – konfiguracja projektu, podstawowa struktura aplikacji.
	Projektowanie i programowanie mobilnych aplikacji cross-platformowych w praktyce.
	Projektowanie i implementacja spójnego interfejsu użytkownika (UI) – stylizacja i responsywność.
	Programowanie i synchronizacja komunikacji między bazą danych a aplikacjami internetowymi, PWA i mobilnymi aplikacjami cross-platformowymi.
	Zabezpieczanie aplikacji PWA oraz mobilnych aplikacji cross-platformowych.
	Testowanie aplikacji PWA oraz mobilnych aplikacji cross-platformowych.
	Debugowanie i wdrażanie aplikacji PWA oraz mobilnych aplikacji cross-platformowych.
Zaawansowane przetwarzanie danych w chmurze	Wprowadzenie do chmury obliczeniowej i modeli usług (IaaS, PaaS, SaaS).
	Modele danych relacyjnych i nierelacyjnych. Języki zapytań SQL i NoSQL.
	Przetwarzanie wsadowe i strumieniowe. Apache Spark, Kafka, Azure Stream Analytics.
	Wzorce rozproszonych baz danych. Replikacja, partycjonowanie, sharding.
	Spójność danych ACID vs BASE. Twierdzenie CAP i ostateczna spójność.
	Hurtownie danych i jeziora danych. ETL/ELT procesy w chmurze.
	Big Data - Hadoop, Spark, architektura Lambda i Kappa.
	Machine Learning w chmurze - MLOps, AutoML, model serving.
	Konfiguracja środowiska chmurowego i podstawowe usługi.
	Praktyczna praca z bazami danych w chmurze.
	Implementacja pipeline'ów przetwarzania danych w czasie rzeczywistym.

	Konfiguracja klastrów bazodanowych i optymalizacja wydajności.
	Testowanie modeli spójności w systemach rozproszonych.
	Budowa data warehouse z wykorzystaniem platform chmurowych.
	Analiza big data z wykorzystaniem platform chmurowych.
	Deployment modeli ML w chmurze.
Zarządzanie tożsamością i dostępem do zasobów	Identyfikacja i klasyfikacja danych w praktyce. Ochrona zasobów w chmurze (np. szyfrowanie, polityki dostępu).
	Konfiguracja i udostępnianie zasobów obliczeniowych, pamięciowych i sieciowych (np. w Azure, AWS).
	Konfiguracja zarządzania tożsamością (np. Azure AD, IAM) i realizacja żądań dostępu.
	Implementacja mechanizmów autoryzacji – OAuth2, MFA.
	Praktyczna administracja usługami domenowymi, zarządzanie użytkownikami i zasobami.
	Symulacja incydentu bezpieczeństwa i prowadzenie działań naprawczych (backup, response plan).
	Praktyczne stosowanie szyfrowania i zarządzania kluczami (np. Key Vault).
	Realizacja scenariuszy bezpieczeństwa w środowiskach hybrydowych. Zarządzanie tożsamością między systemami lokalnymi i chmurowymi w praktyce.
	Identyfikacja i klasyfikacja danych. Zarządzanie i ochrona zasobów danych w chmurze.
	Rodzaje zasobów w środowisku chmurowym. Zasoby obliczeniowe, pamięciowe i sieciowe. Dostarczanie danych.
	Zarządzanie tożsamością. Żądania dostępu do zasobów.
	Kontrola dostępu. Autoryzacja i uwierzytelnienie. Zewnętrzne technologie autoryzacji. OAuth2.
	Tożsamość w chmurze na przykładzie usług domenowych. Administracja. Zarządzanie usługami, użytkownikami oraz zasobami.
	Zarządzanie podatnościami. Wykrywanie, reagowanie i odzyskiwanie danych po incydentach bezpieczeństwa.
Metody szyfrowania danych. Magazyny kluczy i dobre praktyki.	
Inne zabezpieczenia tożsamości i zasobów. Zarządzanie tożsamością i dostępem do zasobów w środowiskach hybrydowych.	
<b>Specjalność: Inżynier aplikacji mobilnych</b>	
Bezpieczeństwo aplikacji mobilnych	Architektura bezpieczeństwa systemów operacyjnych Android oraz iOS.
	Przegląd bezpieczeństwa aplikacji mobilnych. Współczesne problemy i zagrożenia.
	Analiza aplikacji i identyfikacja słabości, problemów z perspektywy programisty i użytkownika. Mechanizmy obrony.
	Ataki sprzętowe. Przelamywanie zabezpieczeń systemów.
	Bezpieczeństwo danych i komunikacji.
	Inżynieria wsteczna.
	Narzędzia i standardy OWASP.
	Techniki dodatkowego zabezpieczenia aplikacji oraz hardening.
Podstawy programowania na platformę Android	Wprowadzenie do programowania aplikacji mobilnych Przegląd systemów mobilnych Android.
	Struktura, układ projektu. Compose.
	Główne komponenty aplikacji Android. Layouty. Aktywności w aplikacjach Android. Android Manifest.

	<p>Cykl życia aktywności. Komunikacja pomiędzy zamiarem Intents i aktywnością.</p> <p>Nawigacja w aplikacjach Android : NavigationDraver, ActionBar.</p> <p>Style, motywy i szablony kolorów. Material Design.</p> <p>Komponenty ListView,GridView,RecyclerView,ScrollView oraz WebView.</p> <p>Obrazy i animacje. Splash screen oraz programowanie aplikacji dla różnych wielkości ekranu - Screen orientation.</p> <p>Instalacja i konfiguracja środowiska Android Studio.</p> <p>Pierwsza aplikacja Hello World. Korzystanie z emulatora oraz fizycznych urządzeń. Debugowanie oprogramowania.</p> <p>Implementacja aktywności, layoutów oraz innych elementów Compose.</p> <p>Implementacja mechanizmów przesyłania danych za pomocą Intent.</p> <p>Implementacja nawigacji w aplikacji Android.</p> <p>Implementacja stylów,motywów i kolorów zgodnych z Materiał Design.</p> <p>Implementacja komponentów ListView,GridView,RecyclerView,ScrollView oraz WebView.</p> <p>Implementacja dodatków oraz obsługa różnych wielkości ekranu.</p>
Podstawy programowania na platformę iOS	<p>Wprowadzenie do programowania w środowisku Xcode; praca z symulatorem urządzeń Apple.</p> <p>Wybrane koncepcje SwiftUI.</p> <p>Wybrane widoki SwiftUI.</p> <p>Layout aplikacji.</p> <p>Powiadomienia w aplikacjach iOS.</p> <p>Nawigacja oraz menu.</p> <p>Animacje i grafika.</p> <p>Praca z mapami i lokalizacją.</p> <p>Tworzenie aplikacji z wykorzystaniem aplikacji Xcode oraz podstawowych widoków dostępnych w SwiftUI; testowanie aplikacji z wykorzystaniem mechanizmu previews.</p> <p>Tworzenie aplikacji wymagających wykorzystania kontenerów takich jak HStack, VStack, ZStack oraz List.</p> <p>Tworzenie aplikacji zawierających formularze.</p> <p>Tworzenie aplikacji z wykorzystaniem typowych mechanizmów nawigacji.</p> <p>Tworzenie aplikacji wykorzystujących powiadomienia lokalne.</p> <p>Implementacja animacji.</p> <p>Tworzenie aplikacji wykorzystujących mapy i lokalizację użytkownika.</p>
Praktyka zawodowa II	<p>Moduły do wyboru:  <b>MODUŁ 1: Projektowanie i implementacja zaawansowanych aplikacji mobilnych:</b></p> <ul style="list-style-type: none"> <li>• Projektuje architekturę aplikacji mobilnych, uwzględniając warstwowy podział systemu, zarządzanie stanem, przepływ danych oraz odpowiednie wzorce projektowe.</li> <li>• Implementuje kluczowe funkcjonalności aplikacji, integrując logikę biznesową, interfejs użytkownika oraz komponenty systemowe urządzeń mobilnych.</li> <li>• Projektuje i wdraża interfejsy użytkownika zgodne ze standardami platform mobilnych, dbając o ergonomię, dostępność oraz spójność doświadczenia użytkownika.</li> <li>• Analizuje działanie aplikacji, identyfikuje problemy funkcjonalne lub wydajnościowe oraz proponuje i wdraża optymalizacje</li> <li>• Stosuje zasady inżynierii oprogramowania, w tym standardy jakości, procedury testowania oraz dobre praktyki programistyczne..</li> </ul>

	<ul style="list-style-type: none"> <li>• Pozyskuje, ocenia i selekcjonuje informacje techniczne oraz dokumentację branżową oraz stosuje dobre praktyki i odpowiednie normy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą wdrażanych rozwiązań, integracji systemów oraz zmian projektowych.</li> </ul> <p>MODUŁ 2: Integracja aplikacji mobilnych z usługami zewnętrznymi i środowiskami wielowarstwowymi:</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje komunikację aplikacji mobilnej z usługami zewnętrznymi, w tym systemami backendowymi, usługami sieciowymi oraz środowiskami chmurowymi.</li> <li>• Implementuje mechanizmy uwierzytelniania, autoryzacji i bezpiecznego przetwarzania danych w aplikacjach mobilnych, zgodnie z obowiązującymi standardami bezpieczeństwa.</li> <li>• Przeprowadza testy funkcjonalne, integracyjne i wydajnościowe aplikacji oraz interpretuje ich wyniki w celu zapewnienia jakości i niezawodności systemu.</li> <li>• Optymalizuje aplikacje mobilne pod kątem działania na urządzeniach o zróżnicowanych parametrach technicznych oraz w środowiskach wieloplatformowych.</li> <li>• Wdraża aplikacje mobilne w środowiskach testowych lub produkcyjnych oraz monitoruje ich działanie..</li> <li>• Pozyskuje, ocenia i selekcjonuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki i odpowiednie normy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą wdrażanych rozwiązań, integracji systemów oraz zmian projektowych.</li> </ul> <p>- zajęcia terenowe</p>
<p>Programowanie w języku Java</p>	<p>Wprowadzenie do programowania w języku Java Konfiguracja środowiska programistycznego oraz programowanie pierwszych programów w języku Java</p> <p>Zmienne i typy danych. Instrukcje warunkowe oraz pętle. Programowanie z użyciem instrukcji warunkowych oraz sterujących.</p> <p>Zbiory danych, kolekcje oraz funkcje. Programowanie z użyciem tablic, kolekcji oraz funkcji.</p> <p>Programowanie obiektowe. Struktury, klasy, metody. Programowanie w oparciu o struktury, klasy i metody w praktyce.</p> <p>Dziedziczenie, abstrakcja, polimorfizm. Programowanie z użyciem dziedziczenia, abstrakcji oraz polimorfizmu.</p> <p>Pakiety oraz interfejsy. Programowanie z wykorzystaniem pakietów oraz interfejsów.</p> <p>Obsługa wejścia, wyjścia. Obsługa wyjątków. Programowanie z użyciem obsługi wejścia, wyjścia oraz wyjątków.</p> <p>Język Kotlin jako następcę Javy. Podobieństwa i różnice. Programowanie pierwszych programów w języku Kotlin.</p>
<p>Programowanie w języku Swift</p>	<p>Składnia i elementy języka Swift; podstawy pracy w środowisku Xcode</p> <p>Zmienne i typy danych. Instrukcje warunkowe oraz pętle</p> <p>Zbiory danych, kolekcje oraz funkcje</p> <p>Struktury, klasy i dziedziczenie; programowanie obiektowe</p> <p>Protokoły i rozszerzenia protokołów</p> <p>Typy opcjonalne, typy generyczne i domknięcia</p> <p>Tworzenie bezpiecznego kodu i obsługa błędów</p> <p>Framework Foundation</p> <p>Tworzenie i edycja plików Xcode Playgrounds</p> <p>Definiowanie zmiennych i stałych z wykorzystaniem właściwych typów danych; tworzenie prostych rozwiązań z wykorzystaniem instrukcji warunkowych i pętli</p> <p>Tworzenie własnych funkcji</p>

	Definiowanie własnych struktur i klas, w tym wykorzystując dziedziczenie
	Tworzenie własnych protokołów oraz implementacja protokołów dostarczanych przez język Swift lub framework Foundation
	Tworzenie bezpiecznego i ekspresyjnego kodu z wykorzystaniem typów opcjonalnych, typów generycznych i domknięć
	Tworzenie kodu z wykorzystaniem mechanizmów obsługi błędów
	Tworzenie skryptów z wykorzystaniem frameworka Foundation
Testowanie aplikacji mobilnych	Pisanie testów oraz scenariuszy testowych. Weryfikowanie aplikacji.
	Praca z narzędziami testerskimi wbudowanymi w środowiska programistyczne Android Studio oraz Xcode.
	Testowanie oprogramowania mobilnego w praktyce z wykorzystaniem dostępnych darmowych i płatnych narzędzi IT.
	Testowanie manualne oraz automatyczne.
	Testowanie aplikacji za pomocą Espresso oraz Appium.
	Testowanie bezpieczeństwa oraz wydajności aplikacji.
	Testowanie aplikacji za pomocą narzędzi developerskich oraz urządzeń fizycznych.
	Raportowanie testów. Kontrola wersji testowych aplikacji.
Zaawansowane programowanie na platformę Android	Usługi i wątki. Powiadomienia i notyfikacje w aplikacjach Android. Broadcast Receiver.
	Przetwarzanie asynchroniczne AsyncTask. Uprawnienia aplikacji Android Studio.
	Model MVVM oraz klasa ViewModel. Wzorce architektoniczne i projektowe.
	Zapisywanie i odczytywanie danych. Shared Preferences. Praca z bazą danych SQLite.
	Usługi Google Firebase. Komunikacja sieciowa Biblioteka Retrofit. Praca z systemami autentykacji użytkowników Google.
	Usługi lokalizacyjne w Android. Mapy Google.
	Dodatkowe biblioteki aplikacji Android. Wysyłanie wiadomości oraz wykonywanie połączeń za pomocą aplikacji Android.
	Publikowanie aplikacji w sklepie Google Play oraz testowanie w symulatorach i urządzeniach fizycznych.
	Tworzenie aplikacji z użyciem Service, Thread, NotificationManager i BroadcastReceiver.
	Obsługa zadań asynchronicznych (np. pobieranie danych z internetu) i nadawanie uprawnień.
	Refaktoryzacja aplikacji do architektury MVVM z użyciem ViewModel i LiveData.
	Tworzenie aplikacji zapisującej dane w SharedPreferences i w bazie danych SQLite oraz Room.
	Integracja Firebase Authentication i pobieranie danych z REST API za pomocą Retrofit.
	Opracowanie aplikacji wskazującej aktualną lokalizację użytkownika oraz prezentującą ją na mapie.
	Implementacja funkcji wysyłania SMS i połączeń telefonicznych z poziomu aplikacji.
	Konfiguracja pliku build.gradle, przygotowanie release APK, testowanie i publikacja w Google Play (symulacja).
Zaawansowane programowanie na platformę iOS	Tworzenie zaawansowanych layoutów z wykorzystaniem SwiftUI
	Dystrybucja aplikacji
	Integracja z frameworkami

	Pobieranie danych z Internetu
	CoreData i SwiftData
	Swift Package Manager
	Dostępność
	System watchOS
	Tworzenie aplikacji wymagających zaawansowanego layoutu
	Tworzenie aplikacji pobierających dane z internetu (integracja z API)
	Tworzenie aplikacji wykorzystujących frameworki dostępne w ekosystemie firmy Apple
	Tworzenie aplikacji wykorzystujących formularze oraz framework CoreData lub SwiftData
	Tworzenie aplikacji wykorzystujących Swift Package Manager do zarządzania zależnościami
	Konfigurowanie projektu zgodnie z wymaganiami dot. wspieranych platform i ich wersji; uruchamianie aplikacji na fizycznych urządzeniach
	Weryfikowanie wsparcia dla mechanizmów dostępności we własnych aplikacjach oraz wprowadzanie wymaganych usprawnień
	Tworzenie aplikacji dla systemu watchOS
<b>Specjalność: Inżynieria programowania procesorów</b>	
Architektura mikroprocesorów i mikrokontrolerów	Klasyfikacja mikroprocesorów oraz mikrokontrolerów. Budowa i zasada działania mikroprocesorów oraz mikrokontrolerów.
	Klasyfikacja sygnałów. Parametry sygnałów deterministycznych.
	Przestrzenie sygnałów i transformacje - wprowadzenie do przestrzeni Hilberta, transformaty Fouriera.
	Dyskretna i szybka transformata Fouriera.
Modelowanie i przetwarzanie danych	Wprowadzenie do modelowania danych: pojęcia danych i informacji, modele danych (relacyjny, obiektowy, hierarchiczny).
	Wprowadzenie do modelowania danych: kluczowe elementy modelowania konceptualnego (diagramy ER).
	Technologie baz danych i NoSQL: przegląd systemów zarządzania bazami danych (RDBMS vs. NoSQL), przykłady nierelacyjnych modeli danych (dokumentowe, klucz-wartość, grafowe) i ich zastosowania w przetwarzaniu dużych zbiorów danych.
	Projektowanie relacyjnych baz danych: zależności funkcyjne.
	Projektowanie relacyjnych baz danych: normalizacja danych (postacie normalne).
	Projektowanie relacyjnych baz danych: tworzenie schematów relacyjnych na podstawie modelu ER.
	Procesy ETL i hurtownie danych: etapy ekstrakcji, transformacji, ładowania danych z różnych źródeł; koncepcja hurtowni danych i podstawy projektowania schematów hurtowni (schemat gwiazdy, płątka śniegu); rola narzędzi ETL.
	Analiza i wizualizacja danych: przygotowanie danych do analizy (czyszczenie, uzupełnianie braków), wprowadzenie do podstaw statystyki opisowej, prezentacja wyników – raporty i wizualizacje (np. wykresy) z użyciem dostępnych narzędzi.
Optymalizacja i analiza kodu	Wprowadzenie do analizy i optymalizacji kodu w systemach wbudowanych
	Metryki wydajności: czas wykonania, zużycie pamięci, zużycie CPU, profiling
	Techniki optymalizacji kodu: algorytmiczne, językowe, kompilatorowe
	Wpływ architektury procesora na wydajność: pipeline, cache, DSP, SIMD
	Optymalizacja na poziomie asemblera: analiza listingu, ręczna optymalizacja

	Metody implementacji filtracji cyfrowej na DSP i mikrokontrolerach
	Metody implementacja estymacji i detekcji sygnałów w systemach czasu rzeczywistego
	Techniki analizy podobieństwa sygnałów i transformacje czasowo-częstotliwościowe w DSP
	Profiling kodu w systemie wbudowanym - pomiar czasu i zużycia zasobów
	Optymalizacja algorytmu przetwarzania sygnału (np. filtr FIR) pod kątem czasu i pamięci w praktyce
	Analiza listingu assemblerowego i ręczna optymalizacja fragmentu kodu
	Implementacja filtru cyfrowego na DSP/mikrokontrolerze
	Implementacja estymatora (np. średniej, wariancji, widma) na DSP
	Implementacja detektora sygnału w szumie na mikrokontrolerze
	Analiza podobieństwa sygnałów (korelacja, dopasowanie) w DSP
	Implementacja i optymalizacja algorytmu przetwarzania sygnału na wybranym DSP/mikrokontrolerze
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Programowanie niskopoziomowe i optymalizacja kodu dla systemów wbudowanych</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje oprogramowanie w językach niskopoziomowych przeznaczone dla wybranych architektur procesorów, wykorzystując różne modele adresowania I zestawy instrukcji</li> <li>• Analizuje zależności między architekturą procesora I organizacją rejestrów a wykonywaniem instrukcji</li> <li>• Analizuje wydajność kodu, wykorzystuje techniki jego optymalizacji pod względem szybkości działania i zużycia pamięci oraz mocy, identyfikując wąskie gardła, błędy I problemy.</li> <li>• Integruje oprogramowanie z komponentami sprzętowymi w systemach embedded.</li> <li>• Testuje I debuguje kod niskopoziomowy, identyfikując błędy logiczne, sprzętowe I wydajnościowe</li> <li>• Pozyskuje, ocenia I selekcjonuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki i odpowiednie normy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą wdrażanych rozwiązań.</li> </ul> <p>MODUŁ 2: Projektowanie i testowanie systemów opartych na architekturze procesorowej</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje moduły oprogramowania współpracujące bezpośrednio z procesorem, uwzględniające architecture sprzętu</li> <li>• Analizuje i optymalizuje przepływ danych, cykle rozkazowe oraz integrację z pamięcią w kontekście architektury procesora, celem optymalizacji wydajności systemu</li> <li>• Tworzy środowiska testowe oraz przeprowadza automatyczne testy funkcjonalne, jednostkowe i wydajnościowe systemów niskopoziomowych.</li> <li>• Stosuje debugowanie sprzętowe (np. JTAG, SWD) oraz analizę sygnałów z wykorzystaniem narzędzi pomiarowych (np. analizator stanów logicznych, oscyloskop).</li> <li>• Wyszukuje, pozyskuje, a także ocenia, analizuje i zestawia informacje stosownie do potrzeb klienta/kontrahenta.</li> <li>• Pozyskuje, ocenia I selekcjonuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki i odpowiednie normy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą wdrażanych rozwiązań.</li> </ul> <p>- zajęcia terenowe,</p>
Programowanie mikrokontrolerów	Wprowadzenie do mikrokontrolerów.
	Konfiguracja portów I/O i operacje bitowe.
	Timery i przerwania.
	Komunikacja szeregową UART.
	Komunikacja I2C i SPI.

	Przetwornik analogowo-cyfrowy (ADC) i czujniki.
	Sterowanie silnikami i PWM.
	Zintegrowany system mikrokontrolera.
	Konfiguracja projektu dla danego mikrokontrolera. Debugowanie.
	Komunikacja UART. Konfiguracja portów I/O i operacje bitowe.
	Konfiguracja i wykorzystanie timerów i przerwań.
	Konfiguracja i wykorzystanie komunikacji SPI i I2C
Programowanie procesorów sygnałowych	Modulacje cyfrowe
	Modulacje impulsowe
	Modulacje impulsowo-kodowe
	Kodowanie. Twierdzenie Shannona
	Rozpraszanie widma
	Zwielokrotnianie dostępu do kanału telekomunikacyjnego
	Architektura DSP i optymalizacja kodu
	Filtracja cyfrowa w czasie rzeczywistym
	Analiza spektralna - FFT i DFT
	Adaptacyjne przetwarzanie sygnałów
Projektowanie układów scalonych (VLSI)	Testowanie i weryfikacja układów scalonych: rodzaje symulacji (funkcjonalna, czasowa, symulacja mocy), testy wektorowe, generowanie wzorców testowych dla układów kombinacyjnych i sekwencyjnych; koncepcja DFT (Design for Testability) – wbudowane struktury ułatwiające testowanie (np. łańcuchy przesuwne, BIST); wstęp do analizy pokrycia testowego.
	Zagadnienia zaawansowane i trendy: przegląd wyzwań w nanometrowych technologiach (efekty pasożytnicze, integralność sygnałowa, wpływ temperatury), nowoczesne kierunki rozwoju projektowania układów (układy FPGA vs ASIC, projektowanie układów analogowo-cyfrowych i systemów w jednocukładowych SoC, wschodzące technologie jak układy neuromorficzne).
	Wprowadzenie do układów scalonych VLSI: pojęcie skali integracji (LSI, VLSI, ULSI), prawo Moore'a i jego konsekwencje, przegląd zastosowań układów scalonych o wysokiej skali integracji; wstęp do technologii CMOS – tranzystory MOSFET (NMOS, PMOS), implementacja bramek logicznych (inwertery, bramki NAND/NOR) w technice CMOS.
	Języki opisu sprzętu (HDL): wprowadzenie do VHDL i Verilog – składnia, modelowanie elementów logicznych, tworzenie modułów, symulacja behawioralna vs symulacja strukturalna; dobre praktyki w opisie sprzętu (podział na moduły, parametryzacja, unikanie konstrukcji niesynteżowalnych).
	Cyfrowe układy logiczne w układach scalonych: projektowanie bloków kombinacyjnych (sumatory, dekodery, multipleksery) i sekwencyjnych (przerzutniki, liczniki) na poziomie schematu oraz ich charakterystyki (opóźnienie propagacji, pobór mocy, ograniczenia fan-in/fan-out); pojęcie skalowania napięcia zasilania i wpływ na pobór mocy.
	Tapy projektowania układów VLSI: projektowanie na poziomie RTL (Register-Transfer Level) i synteza logiczna – zamiana kodu HDL na siatkę bramek logicznych; optymalizacja układu pod względem szybkości, powierzchni i mocy (trade-offs między tymi parametrami); projektowanie fizyczne – pojęcie floorplanu, rozmieszczanie komponentów (placement), prowadzenie połączeń (routing), weryfikacja reguł projektowych (DRC – Design Rule Check, LVS – Layout vs Schematic).
Systemy wbudowane	Wprowadzenie do systemów wbudowanych. Wybór platformy docelowej (Arduino, Raspberry Pi). Konfiguracja środowiska programistycznego (Arduino IDE lub Raspberry Pi OS + Python).
	Wprowadzenie do programowania wybranej platformy. GPIO - sterowanie diodami LED, czytanie przycisków.

	Praca z czujnikami analogowymi i cyfrowymi. Pomiar temperatury, wilgotności, natężenia światła. Wyświetlanie danych na LCD/OLED.
	Protokoły komunikacyjne: I2C i SPI. Komunikacja z czujnikami i modułami peryferyjnymi. Obsługa błędów komunikacji.
	Komunikacja szeregową UART. Przesyłanie danych do komputera, logowanie pomiarów, interfejs z PC.
	Komunikacja bezprzewodowa - WiFi lub Bluetooth. Wysyłanie danych do chmury, zdalne sterowanie urządzeniem.
	Automatyzacja i sterowanie. PWM, serwosilniki, silniki krokowe. Budowa systemu automatyki.
	Integracja systemów - łączenie komunikacji bezprzewodowej z automatyką.
Technika cyfrowa	Pojęcie systemu analogowego i cyfrowego.
	Bramki logiczne i minimalizacja funkcji. Układy kombinacyjne.
	Przerzutniki i układy sekwencyjne. Maszyny stanów
	Analiza sygnałów cyfrowych w dziedzinie czasu i częstotliwości.
	Charakterystyka i porównanie systemów analogowych i cyfrowych.
	Tworzenie układów kombinacyjnych z pomocą bramek logicznych i minimalizacji funkcji.
	Tworzenie maszyny stanów z wykorzystaniem przerzutników i układów sekwencyjnych.
	Wykorzystanie sygnałów cyfrowych w dziedzinie czasu i częstotliwości.
<b>Specjalność: Inżynieria sieci, systemów i bezpieczeństwa IT</b>	
Bezpieczeństwo usług serwerowych	Wzmocnienie tożsamości i polityki konta domenowego Windows Server
	Zabezpieczanie dostępu zdalnego (RDP) i serwera plików Windows
	Kontrola uprawnień stacji roboczych i zarządzanie Microsoft LAPS
	Zaawansowane zarządzanie GPO
	Monitorowanie, audyt zdarzeń w Windows Server
	Zabezpieczanie usług systemowych Linux
	Audyt uprawnień systemu Linux
	Zaawansowana kontrola dostępu (ACL) w systemach Linux
	Konfiguracja zapory sieciowej i audyt sieci systemu Linux
	Szyfrowanie dysków i dzienniki audytowe w Linux
	Podstawowe pojęcia i zagrożenia występujące w środowiskach serwerowych Windows i Linux
	Ochrona hostów oraz danych w systemach serwerowych
	Uwierzytelnienie i autoryzacja w systemach serwerowych
	Kontrola praw i uprawnień użytkowników
	Zapory sieciowe i ich konfiguracja
	Infrastruktura PKI oraz wykorzystanie certyfikatów w Windows Server
	Zabezpieczanie serwera aplikacji www IIS i Apache
	Kontrola dostępu do sieci w systemach Linux i Windows
Bezpieczne administrowanie serwerami	
Zabezpieczanie usług sieciowych	
Inżynieria sieci komputerowych I	Wprowadzenie do routingu: budowa routera.
	Routing statyczny: interfejsy routera i ich konfiguracja.
	Wprowadzenie do routingu dynamicznego: porównanie routingu statycznego i dynamicznego, protokoły IGP i EGP, metryki, odległości administracyjne.

	<p>Protokoły routingu wektora odległości: wykrywanie sieci, pętle routingu. Protokół RIP: RIPv2, RIPv2, VLSM, CIDR</p> <p>Szczegóły tablicy routingu: struktura tablica, przeszukiwanie tablicy, routing klasowy i bezklasowy</p> <p>Protokół EIGRP.</p> <p>Protokoły routingu stanu łącza: OSPF.</p> <p>Implementacja podstawowych schematów adresowania, wyznaczanie trasy, metryka.</p> <p>Konfiguracja routingu statycznego. Badanie sieci bezpośrednio dołączonych, sumaryczne i domyślne trasy statyczne.</p> <p>Konfiguracja routingu dynamicznego.</p> <p>Konfiguracja protokołu RIP w routerze.</p> <p>Weryfikacja tablic routingu, analiza porównawcza tablic routingu statycznego i dynamicznego.</p> <p>Konfiguracja i wdrażanie protokołu EIGRP.</p> <p>Konfiguracja i wdrażanie protokołu OSPF.</p>
Inżynieria sieci komputerowych II	<p>Konfiguracja i analiza działania sieci VoIP z wykorzystaniem Packet Tracer i Wireshark.</p> <p>Implementacja i testowanie protokołu STP w sieci przełączanej.</p> <p>Zarządzanie sieciami VLAN przy użyciu protokołu VTP.</p> <p>Konfiguracja połączenia Frame Relay w sieci symulowanej.</p> <p>Tworzenie połączenia punkt-punkt z użyciem protokołu PPP.</p> <p>Uwierzytelnianie dostępu do sieci za pomocą serwera RADIUS.</p> <p>Konfiguracja translacji adresów NAT w sieci LAN-WAN.</p> <p>Konfiguracja przełącznika warstwy 3 i routing między VLAN-ami.</p> <p>Technologia VoIP.</p> <p>Protokół STP.</p> <p>Protokół VTP.</p> <p>Technologia Frame Relay.</p> <p>Protokół PPP.</p> <p>Protokół Radius.</p> <p>Translacja adresów.</p> <p>Przełączniki wielowarstwowe.</p>
Konfiguracja, zarządzanie i utrzymanie systemów Windows serwer	<p>Instalacja i konfiguracja środowiska AD DS w Windows Server.</p> <p>Zarządzanie usługami domenowymi w usłudze Active Directory w Windows Server.</p> <p>Konfiguracja i zarządzanie usługami sieciowymi w Windows Serwer.</p> <p>Zarządzanie i utrzymywanie infrastruktury Active Directory.</p> <p>Tworzenie i zarządzanie zasadami grup i preferencjami zasad grup.</p> <p>Zarządzanie replikacją w środowisku domenowym Active Directory.</p> <p>Konfiguracja i zarządzanie monitoringiem środowisk serwerowych.</p> <p>Konfiguracja bezpieczeństwa serwerowych systemów operacyjnych.</p>
Ochrona brzegowa sieci	<p>Rola ochrony brzegowej w architekturze sieci</p> <p>Klasyfikacja zagrożeń sieciowych</p> <p>Modele i strategie ochrony sieci</p> <p>Urządzenia klasy UTM – budowa i funkcjonalność</p> <p>Podstawowe mechanizmy zabezpieczania routerów UTM</p>

	Zaawansowane funkcje bezpieczeństwa UTM
	Listy kontroli dostępu w UTM
	Systemy IPS w UTM
	Mechanizmy filtrowania treści i ruchu
	Zarządzanie i raportowanie bezpieczeństwa
	Podstawowa konfiguracja urządzenia UTM
	Zabezpieczanie routera i UTM przed nieautoryzowanym dostępem
	Konfiguracja standardowych list kontroli dostępu
	Tworzenie zaawansowanych reguł ACL
	Konfiguracja i testowanie systemu IPS
	Filtrowanie URL w praktyce
	Filtrowanie i inspekcja ruchu SSL
	Integracja UTM z LDAP / Active Directory
	Konfiguracja połączeń VPN
	Raportowanie i analiza zdarzeń bezpieczeństwa
Planowanie, zarządzanie i bezpieczeństwo systemów klienckich	Standaryzacja systemów Windows Pro
	Konfiguracja Windows Server Update Services (WSUS) i wdrożenie na stacji roboczej
	Zarządzanie stacjami klienckimi przy pomocy Group Policy Objects (GPO)
	Konfiguracja polityk bezpieczeństwa przy użyciu GPO
	Konfiguracja ochrony systemu klasy Endpoint Security
	Wdrażanie polityk bezpieczeństwa przy pomocy systemu Endpoint Security
	Wdrażanie zadań ochrony stacji roboczych Endpoint Security
	Zarządzanie certyfikatami na stacji roboczej
	Praktyczne wykorzystanie własnej infrastruktury PKI (CA) na stacji roboczej
	Projektowanie, konfigurowanie i zarządzanie środowiskiem klienckim.
	Automatyczne wdrażanie systemów klienckich. Użycie zaawansowanych narzędzi do zarządzania systemem.
	Centralne zarządzanie ochroną typu „endpoint security”. Centralne zarządzanie funkcjonalnością systemów klienckich.
	Użycie zaawansowanych narzędzi do zarządzania systemem.
	Automatyzowanie zadań przy skryptów administracyjnych.
	Użycie wiersza poleceń cmd i PowerShell do zarządzania stacjami roboczymi.
	Wdrażanie konsoli ochrony typu endpoint security.
Wdrażanie i zarządzanie ochroną stacji roboczych typu endpoint security. Polityki ochrony endpoint security.	
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Konfiguracja i zarządzanie infrastrukturą sieciową i serwerową:</p> <ul style="list-style-type: none"> <li>• Konfiguruje i zarządza urządzeniami sieciowymi (routery, przełączniki, firewalle) z wykorzystaniem CLI oraz narzędzi zarządzania siecią</li> <li>• Tworzy i wdraża reguły routingu, VLAN-y, listy kontroli dostępu (ACL) oraz usługi sieciowe (np. DHCP, DNS, VPN).</li> <li>• Instaluje, konfiguruje i administruje systemami serwerowymi (Linux/Windows Server), w tym usługami katalogowymi, serwerami WWW i bazami danych</li> <li>• Monitoruje i diagnozuje ruch sieciowy, przy użyciu narzędzi takich, jak Wireshark, tcodump oraz wybranych systemów NMS</li> <li>• Obsługuje wirtualizację i/lub konteneryzację</li> <li>• Monitoruje wydajność i dostępność usług IT oraz odchylenia od normy</li> <li>• Wykorzystuje normy i standardy branżowe, dokumentację techniczną oraz dobre praktyki w procesie realizacji zadań</li> </ul>

	<ul style="list-style-type: none"> <li>• Przygotowuje dokumentację związaną z wykonanymi czynnościami administracyjnymi i/lub wdrożeniowymi</li> </ul> <p>MODUŁ 2: Bezpieczeństwo systemów informatycznych i reagowanie na incydenty:</p> <ul style="list-style-type: none"> <li>• Analizuje zagrożenia, identyfikuje podatności oraz ocenia ryzyko w systemach operacyjnych, sieciach i aplikacjach.</li> <li>• Konfiguruje i obsługuje systemy ochrony końcówek (EDR), zapory sieciowe, systemy detekcji intruzów (IDS/IPS), certyfikaty i mechanizmy szyfrowania.</li> <li>• Realizuje testy bezpieczeństwa oraz analizuje logi i zdarzenia z wykorzystaniem narzędzi takich jak Kali Linux, Nmap, Metasploit, SIEM (np. Wazuh, Splunk).</li> <li>• Bierze udział w procesach reakcji na incydenty oraz wspiera działania zgodnie z odpowiednimi procedurami SOC</li> <li>• Wdraża mechanizmy kopii zapasowych, procedur odzyskiwania danych oraz polityk bezpieczeństwa</li> <li>• Wykorzystuje normy i standardy branżowe, dokumentację techniczną oraz dobre praktyki w procesie realizacji zadań</li> <li>• Przygotowuje dokumentację związaną z wykrytymi incydentami, przeprowadzonymi analizami, lub testami bezpieczeństwa.</li> </ul> <p>- zajęcia terenowe</p>
<p>Usługi webowe i zaawansowana konfiguracja systemów Windows serwer</p>	<p>Instalacja internetowych usług informacyjnych oraz konfiguracja witryny domyślnej.</p> <p>Konfiguracja puli aplikacji, witryn internetowych i aplikacji internetowych.</p> <p>Zabezpieczanie witryn internetowych oraz transmisji danych.</p> <p>Instalacja oraz wdrażanie usługi FTP.</p> <p>Tworzenie kopii zapasowych oraz monitorowanie internetowych usług informacyjnych.</p> <p>Wdrażanie pamięci masowych.</p> <p>Konfiguracja równoważenia obciążenia sieciowego (NLB).</p> <p>Konfiguracja klastra pracy awaryjnej.</p>
<p>Wirtualizacja systemów informatycznych Hyper-V i VMware</p>	<p>Instalacja i konfiguracja środowiska Hyper-V oraz VMware Workstation/ESXi. Konfiguracja środowiska testowego, instalacja hypervisora, konfiguracja sieci i interfejsów zarządzania</p> <p>Tworzenie i konfiguracja maszyn wirtualnych</p> <p>Konfigurowanie i zarządzanie sieciami wirtualnymi</p> <p>Konfigurowanie magazynów danych w środowiskach zwirtualizowanych</p> <p>Testowanie dynamicznej alokacji zasobów (CPU, RAM), obserwacja zużycia w narzędziach monitorujących ESXi/Hyper-V</p> <p>Konfiguracja klastrów (Failover Cluster, HA w VMware), test migracji maszyn (Live Migration, vMotion)</p> <p>Wprowadzenie do wirtualizacji – historia, pojęcia podstawowe, typy hypervisorów (Type 1, Type 2), architektura systemów wirtualnych</p> <p>Architektura i funkcje systemów wirtualizacji: hosty, maszyny wirtualne, BIOS/EFI, snapshoty, szablony</p> <p>Sieci wirtualne – typy połączeń (Internal, External, Private, vSwitch, vLAN), integracja z infrastrukturą LAN</p> <p>Magazyny danych i przechowywanie - formaty dysków (VHDX, VMDK), protokoły iSCSI, NFS, Fibre Channel</p> <p>Zarządzanie zasobami, równoważenie obciążenia i monitoring - CPU, RAM, I/O. Mechanizmy QoS i dynamicznej alokacji zasobów</p> <p>Wysoka dostępność (HA, FT, DRS, vMotion, Live Migration) - klastrowanie i redundancja w środowiskach wirtualnych</p> <p>Kopie zapasowe oraz backup wirtualnych maszyn</p> <p>Przenoszenie plików maszyn wirtualnych, backup VMDK/VHDX</p>

	Monitoring, diagnostyka i rozwiązywanie problemów w środowiskach Hyper-V i VMware
	Diagnozowanie błędów sieci, problemów z wydajnością, analiza logów systemowych i zdarzeń hypervisora w praktyce
<b>Specjalność: Programista aplikacji w chmurze z wykorzystaniem AI</b>	
Bazy danych i Big Data	Wstęp do Big Data (definicje, architektura, narzędzia wykorzystywane w obszarze baz danych i Big Data).
	Analiza dużych zbiorów danych.
	Bazy danych NoSQL (systemy baz danych NoSQL i wykorzystywane w nich języki zapytań).
	Cykl projektowy w Big Data.
	Analiza i projektowanie w Big Data.
	Przegląd wybranych metod analitycznych przy użyciu pakietu obliczeniowego R. Zaawansowane metody analityczne - teoria i przykłady.
	Praktyka wykorzystania zaawansowanych metod analitycznych.
	Platforma Hadoop, Apache HIVE, Apache PIG.
	Przetwarzanie równoległe i rozproszone w praktyce - Hadoop, MapReduce.
	Modele eksploracji danych w R.
	Wizualizacje w Big Data.
	Narzędzia informatyczne i ich efektywne wykorzystanie dla dużych zbiorów danych.
	Wykorzystanie narzędzi informatycznych do wizualizacji w Big Data.
	Wykorzystanie języków zapytań do baz danych.
Budowa i administracja aplikacji w chmurze	Konfiguracja środowiska deweloperskiego w chmurze i pierwszy deployment.
	Budowa aplikacji mikrousługowej z wykorzystaniem kontenerów.
	Implementacja CI/CD pipeline dla aplikacji chmurowej.
	Zarządzanie bazami danych w chmurze i backup/recovery.
	Monitoring, logging i alerting aplikacji w produkcji.
	Skalowanie automatyczne i zarządzanie ruchem.
	Bezpieczeństwo aplikacji - uwierzytelnianie i autoryzacja w chmurze.
	Serverless computing - budowa aplikacji bezserwerowej.
Podstawowe narzędzia: Python i open source	Wprowadzenie do środowiska open source. Ekosystem Pythona, licencje (MIT, GPL, Apache), modele rozwoju i współpracy
	Interaktywne notatniki - Jupyter Notebook, JupyterLab, ich rola w analizie danych i prototypowaniu
	Zarządzanie środowiskiem pracy - wirtualne środowiska (venv, conda), menedżery pakietów (pip, poetry), zarządzanie zależnościami
	Pakiety open source w Pythonie. Przegląd bibliotek (numpy, pandas, matplotlib, requests, flask, itp.), wybór i integracja w projektach
	Implementowanie funkcji i modułów. Dobre praktyki projektowania kodu, struktura projektu Pythonowego, dokumentacja (docstrings, README)
	Testowanie oprogramowania w oparciu o pakiety i narzędzia Python (pytest, unittest).
	Git i współpraca zespołowa. Podstawy Git, workflow (feature branch, pull request), GitHub/GitLab, współpraca w open source
	Potoki CI/CD (GitHub Actions, GitLab CI), automatyzacja budowania, testowania i wdrażania projektów Pythonowych.
	Konfiguracja środowiska. Instalacja Pythona, menedżerów pakietów, tworzenie wirtualnych środowisk (venv/conda)

	Praca z Jupyter Notebook. Tworzenie notatnika, analiza danych, wizualizacja, eksport wyników
	Instalacja, konfiguracja i wykorzystanie bibliotek (np. numpy, pandas, matplotlib) w analizie danych
	Implementacja modułów i funkcji. Tworzenie własnych pakietów
	Pisanie testów jednostkowych (pytest/unittest), uruchamianie testów, analiza pokrycia kodu
	Praca z systemem GitHub. Inicjalizacja repozytorium, commit, branch, pull request, współpraca w zespole
	Tworzenie README, docstrings, generowanie dokumentacji (np. Sphinx)
	Ćwiczenia praktyczne z automatyzacji budowania, testowania i wdrażania projektów Pythonowych
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Projektowanie i wdrażanie aplikacji chmurowych z elementami sztucznej inteligencji</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje aplikacje webowe lub mobilne działające w środowiskach chmurowych, z wykorzystaniem modelu usługowego chmury oraz architektur opartych na usługach i funkcjach uruchamianych na żądanie</li> <li>• Integruje komponenty sztucznej inteligencji z aplikacjami użytkownika przy użyciu interfejsów API, usług chmurowych oraz modeli AI as a Service</li> <li>• Opracowuje logikę aplikacji obejmującą przetwarzanie danych, uwierzytelnianie, autoryzację, zarządzanie sesjami oraz obsługę zdarzeń w środowisku chmurowym</li> <li>• Projektuje i wdraża procesy automatyzacji budowy, testowania i publikacji oprogramowania oraz monitoruje pracę aplikacji w środowisku chmury</li> <li>• Wyszukuje, pozyskuje, a także ocenia, analizuje i zestawia informacje stosownie do potrzeb klienta/kontrahenta.</li> <li>• Pozyskuje i selekcjonuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki projektowe i odpowiednie normy i standardy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą projektowanego i wdrażanego rozwiązania.</li> </ul> <p>MODUŁ 2: Budowa i integracja rozproszonych systemów AI z wykorzystaniem konteneryzacji i praktyk DevOps</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje komponenty systemów rozproszonych, w tym moduły wykorzystujące uczenie maszynowe i/lub analizę danych i uruchamia je w środowiskach chmurowych z użyciem konteneryzacji</li> <li>• Integruje modele sztucznej inteligencji z backendem aplikacji, umożliwiając automatyczne podejmowanie decyzji lub przetwarzanie danych w czasie rzeczywistym.</li> <li>• Buduje i konfiguruje potoki przetwarzania danych oraz potoki uczenia maszynowego, zapewniające automatyzację przygotowania danych, trenowania modeli oraz wdrażania nowych wersji.</li> <li>• Zarządza cyklem życia aplikacji i modeli AI, stosując praktyki automatyzacji, monitorowania oraz utrzymania ciągłości działania w środowiskach chmurowych.</li> <li>• Wyszukuje, pozyskuje, a także ocenia, analizuje i zestawia informacje stosownie do potrzeb klienta/kontrahenta.</li> <li>• Pozyskuje i analizuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki programistyczne i odpowiednie normy i standardy.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą projektowanego i wdrażanego rozwiązania.</li> </ul> <p>- zajęcia terenowe,</p>
Programowanie i architektura aplikacji w chmurze I	<p>Wprowadzenie do programowania w chmurze - modele usług i dostawcy.</p> <p>Wprowadzenie do architektury aplikacji webowych - klient-serwer, REST API.</p> <p>Bazy danych w chmurze - relacyjne vs NoSQL, Database as a Service.</p> <p>Uwierzytelnianie i autoryzacja w aplikacjach chmurowych - JWT, OAuth 2.0.</p> <p>Przechowywanie plików w chmurze - object storage, CDN.</p> <p>Deployment aplikacji - Platform as a Service (PaaS).</p>

	Monitorowanie i logowanie aplikacji chmurowych.
	Wzorce projektowe dla aplikacji chmurowych - cache, circuit breaker.
	Konfiguracja kont i środowisk w platformach chmurowych (AWS/Azure/GCP).
	Tworzenie pierwszej aplikacji webowej z API REST.
	Integracja aplikacji z bazą danych w chmurze (RDS/Azure SQL/Cloud SQL).
	Implementacja systemu logowania z wykorzystaniem usług tożsamości chmury.
	Upload i serwowanie plików przez S3/Blob Storage z integracją CDN.
	Wdrożenie aplikacji na platformy PaaS (Heroku/App Service/App Engine).
	Konfiguracja monitoringu i analizy logów aplikacji.
	Implementacja cache'owania i obsługi błędów w aplikacji chmurowej.
Programowanie i architektura aplikacji w chmurze II	Dekompozycja aplikacji monolitycznej na mikrousługi z API Gateway.
	Konteneryzacja mikrousług i konfiguracja multi-container environment.
	Deploy aplikacji mikrousługowej na klaster Kubernetes.
	Konfiguracja service mesh dla zarządzania ruchem i bezpieczeństwem.
	Implementacja asynchronicznej komunikacji z użyciem Kafka/RabbitMQ.
	Budowa aplikacji serverless z Lambda/Azure Functions/Cloud Functions.
	Implementacja kompletnego pipeline CI/CD z automated testing i deployment.
	Konfiguracja kompleksowego monitoringu dla aplikacji rozproszonej.
	Architektura mikrousług - zalety, wyzwania, wzorce komunikacji.
	Konteneryzacja aplikacji - Docker, registry, best practices.
	Orkiestracja kontenerów - wprowadzenie do Kubernetes.
	Service mesh i komunikacja między mikrousługami - Istio, service discovery.
	Event-driven architecture - message queues, event streaming.
	Serverless computing - funkcje chmurowe, FaaS, BaaS.
CI/CD dla aplikacji chmurowych - GitOps, automatyczne testy, blue-green deployment.	
Observability i resilience - monitoring, tracing, chaos engineering.	
Projekt zespołowy	Organizacja projektu zespołowego i metodologia Scrum. Podział ról w zespole, wybór tematu projektu aplikacji chmurowej, analiza wymagań i projektowanie architektury.
	Konfiguracja środowiska deweloperskiego i infrastruktury. Wybór stosu technologicznego, utworzenie repozytoriów Git, konfiguracja CI/CD pipeline, zarządzanie projektami.
	Implementacja MVP (Minimum Viable Product). Rozwój podstawowej funkcjonalności aplikacji, integracja z usługami chmurowymi, pierwsze testy jednostkowe.
	Rozbudowa funkcjonalności i integracje. Implementacja zaawansowanych funkcji, integracja z zewnętrznymi API, system uwierzytelniania i autoryzacji.
	Testowanie, monitoring i dokumentacja. Implementacja testów automatycznych (unit, integration, e2e), konfiguracja monitoringu aplikacji, przygotowanie dokumentacji technicznej.
	Finalizacja i deployment produkcyjny. Optymalizacja wydajności, audyt bezpieczeństwa, wdrożenie na środowisko produkcyjne, przygotowanie dokumentacji końcowej.
Systemy rozproszone	Wprowadzenie, nazewnictwo w systemach rozproszonych
	Komunikacja jednostkowa i grupowa w systemach rozproszonych
	Procesy i wątki w systemach rozproszonych. Wędrowka procesów.
	Typy zegarów w systemach rozproszonych.

	Zagadnienia użycia wątków w poszczególnym procesie. Wywołanie przerw systemowych i ich priorytetyzacja.
	Synchronizacja procesów – podejście ogólne i funkcjonalne
	Elektroniczna wymiana danych, przegląd komunikatów
	Zagadnienia użycia wątków w poszczególnym procesie. Wywołanie przerw systemowych i ich priorytetyzacja.
	Konfiguracja środowiska (Python, biblioteki socket, threading, asyncio); proste połączenie klient–serwer TCP/UDP. Analiza adresacji i podstaw komunikacji.
	Implementacja komunikacji multicast; wysyłanie wiadomości do wielu hostów; porównanie z komunikacją unicast.
	Tworzenie procesów/wątków (threading, multiprocessing), praca w modelu producent–konsument; obserwacja użycia CPU i pamięci; omówienie migracji procesów.
	Badanie synchronizacji zegarów pomiędzy węzłami (algorytm Lamporta, Berkeley); pomiar różnic czasowych.
	Implementacja semaforów, blokad i barier w systemie wielowątkowym; eliminacja wyścigów danych.
	Analiza wywołań przerw systemowych, implementacja priorytetów procesów i wątków (np. nice, ps, top); pomiar czasu obsługi.
	Tworzenie bufora komunikatów między procesami, system kolejki zadań (np. queue, RabbitMQ); analiza niezawodności przesyłu.
	Opracowanie prostego systemu (np. czat P2P, rozproszony licznik zasobów, mini broker wiadomości).
Usługi i platformy deweloperskie dla aplikacji w chmurze	Praca z systemami przetwarzania w chmurze. Omówienie różnic pomiędzy IaaS, PaaS, SaaS i Serverless.
	Maszyny wirtualne. Apropowizacja i zarządzanie maszynami wirtualnymi.
	Tworzenie środowiska deweloperskie w platformach chmurowych Backend-As-A-Service BaaS.
	Konteneryzacja. Tworzenie i uruchamianie kontenerów. Docker.
	Wdrażanie aplikacji chmurowych na publicznych serwerach chmurowych.
	Analiza ruchu w aplikacji internetowej. Monitorowanie zużycia zasobów serwerowych. Praca z dodatkowymi usługami chmurowe np. Admob. Cloud Functions. CDN, Load Balancing, Firewall.
	Magazyny i bazy danych. Wykorzystywanie w praktyce baz nierelacyjnych w aplikacjach chmurowych. BLOB.
	Bezpieczeństwo danych. Wdrażanie mechanizmów uwierzytelniania i kontroli dostępu. Wdrażanie usług uwierzytelnienia typu OAuth2.
<b>Specjalność: Programista gier komputerowych</b>	
Architektura i języki programowania gier komputerowych	Konfiguracja środowiska programistycznego dla wybranego silnika.
	Implementacja prostego silnika gry lub wykorzystanie gotowego frameworka.
	Programowanie komponentów: ruch postaci, kolizje, UI.
	Zastosowanie wzorców projektowych w grze 2D.
	Tworzenie prostego systemu ECS.
	Ładowanie i zarządzanie zasobami (audio, grafika, tekstury).
	Optymalizacja kodu i analiza wydajności.
	Publikacja gry w Internecie.
	Wprowadzenie do architektury gier komputerowych.
	Przegląd języków programowania w produkcji gier (C++, C#, Python, Lua).
	Struktura silników gier: Unity, Unreal, Godot.
	Wzorce projektowe: Game Loop, Singleton, Observer, Factory.

	Projektowanie systemów ECS (Entity Component System).
	Zarządzanie zasobami i ładowanie danych.
	Programowanie wielowątkowe i optymalizacja.
	Wprowadzenie do skryptowania i języków domenowych w grach.
Praktyka zawodowa II	<p>Moduły do wyboru:</p> <p>MODUŁ 1: Projektowanie i implementacja mechaniki gier 2D/3D:</p> <ul style="list-style-type: none"> <li>• Projektuje i rozwija funkcjonalne prototypy gier komputerowych z wykorzystaniem silników gier (np. Unity, Unreal Engine) oraz języków programowania (np. C#, C++, Blueprint), testując różne warianty mechanik i iterując rozwiązania.</li> <li>• Projektuje i implementuje systemy rozrywki, w tym, logikę gry, interakcje gracza z otoczeniem, systemy fizyki kolizji, sterowania oraz zarządzania stanem gry i przepływem scen.</li> <li>• Tworzy i integruje interfejsy użytkownika oraz elementy (UI/UX) w grach, dbając o spójność interakcji i czytelność komunikacji z graczem.</li> <li>• Integruje i optymalizuje zasoby graficzne, dźwiękowe i animacje z kodem gry, zapewniając ich poprawną współpracę z logiką gry i wydajność działania.</li> <li>• Analizuje i dokumentuje działanie wdrażanych mechanik, wykorzystując narzędzia debugowania i profilowania w silniku gry.</li> <li>• Pozyskuje i selekcjonuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki projektowe i odpowiednie normy oraz standardy wykorzystywane w produkcji gier.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą projektowanego i wdrażanego rozwiązania.</li> </ul> <p>MODUŁ 2: Programowanie systemów sztucznej inteligencji i dynamicznych mechanik gry</p> <ul style="list-style-type: none"> <li>• Projektuje i implementuje systemy sztucznej inteligencji, planowania zachowań, algorytmów pathfindingu oraz mechanizmów decyzyjnych w grach (np. FSM, Behavior Trees, Pathfinding, Decision Making).</li> <li>• Tworzy proceduralne systemy generowania zawartości (np. poziomów, map, wydarzeń, parametrów rozgrywki) oraz rozwiązania adaptujące dynamikę gry do zachowań użytkownika.</li> <li>• Programuje zaawansowane zachowania postaci niezależnych (NPC) w grach jedno- i wieloosobowych integrując je z systemem animacji, fizyki, logiki oraz środowiskiem gry.</li> <li>• Optymalizuje działanie systemów AI i skryptów pod względem wydajności, zarządzania zasobami i stabilności działania gry</li> <li>• Integruje systemy AI z innymi komponentami gry, zapewniając ich spójne działanie.</li> <li>• Wyszukuje, pozyskuje i analizuje informacje techniczne oraz dokumentację branżową, stosuje dobre praktyki projektowe i odpowiednie normy oraz standardy projektowania i dobrej praktyki programistycznej.</li> <li>• Przygotowuje dokumentację techniczną dotyczącą projektowanych rozwiązań oraz wyników testów systemów AI.</li> </ul>
Programowanie gier mobilnych	<p>Wprowadzenie do projektowania gier mobilnych.</p> <p>Architektura gier mobilnych. Różnice pomiędzy grami mobilnymi a komputerowymi.</p> <p>Programowanie gestów. Programowanie różnych wielkości ekranów.</p> <p>Przygotowywanie rozgrywki.</p> <p>Wykorzystywanie zasobów urządzenia fizycznego.</p> <p>Połączenie gry z zasobami i usługami sieciowymi.</p> <p>Programowanie gier z wykorzystaniem VR oraz AR.</p> <p>Testowanie, optymalizacja i dystrybucja gier mobilnych.</p> <p>Tworzenie szkieletu gry mobilnej w Unity – scena, menu, podstawowe obiekty.</p> <p>Obsługa ekranów dotykowych i gestów – swipe, tap, multitouch w praktyce.</p> <p>Adaptacja interfejsu do różnych rozdzielczości ekranów.</p>

	<p>Tworzenie podstawowej rozgrywki – ruch, kolizje, warunki zwycięstwa/przegranaj.</p> <p>Implementacja komponentów fizycznych – żyroskop, akcelerometr, GPS.</p> <p>Integracja z serwerem – pobieranie danych JSON z API.</p> <p>Wprowadzenie do AR/VR w Unity – tworzenie prostego doświadczenia AR.</p> <p>Profilowanie i optymalizacja gry – testy wydajności na urządzeniu.</p>
Programowanie gier w środowisku Unity	<p>Wprowadzenie do środowiska Unity 3D.</p> <p>Praca z komponentami, zasobami oraz sceną.</p> <p>Renderer, Collider, Rigidbody, Transform.</p> <p>Modele, materiały i tekstury. Teksturowanie obiektów. Shadery i materiały.</p> <p>Praca z Assets oraz Sprite.</p> <p>Kamera i światło.</p> <p>Korzystanie z elementów fizycznie symulowanych Kolizje obiektów.</p> <p>Prefabrykaty. Praca z animacjami. Animator.</p> <p>Efekty dźwiękowe. Tworzenie interfejsu użytkownika. Tworzenie menu w grze.</p> <p>Praca z narzędziami i bibliotekami z Unity Store. Zapis gry do formy wykonywalnej. Publikowanie gry.</p>
Programowanie gier w środowisku Unreal	<p>Instalacja i konfiguracja Unreal Engine.</p> <p>Tworzenie projektu gry w UE.</p> <p>Tworzenie i programowanie aktorów w C++.</p> <p>Interakcje gracza z otoczeniem.</p> <p>Projektowanie poziomów i interfejsu użytkownika.</p> <p>Implementacja systemu dźwięków i efektów.</p> <p>Optymalizacja i testowanie gry.</p> <p>Publikacja gry i przygotowanie do prezentacji.</p>
Programowanie systemów Motion Capture	<p>Zapoznanie z systemem motion capture i jego interfejsem.</p> <p>Konfiguracja systemu – ustawienie kamer/sensorów, kalibracja przestrzeni.</p> <p>Przygotowanie sesji z aktorem – stroje, markery, synchronizacja.</p> <p>Rejestrowanie danych ruchu – sesja nagraniowa, planowanie scen. Nagranie krótkiej sekwencji ruchu i jej analiza.</p> <p>Obsługa wirtualnej kamery i podstawowe zasady reżyserii ruchu.</p> <p>Tworzenie rigowanej postaci w środowisku 3D (Blender, Maya) i implementacja w Unity/Unreal.</p> <p>Integracja danych motion capture z animacją postaci w grze.</p> <p>Przygotowanie miniprojektu: fragment rozgrywki wykorzystujący dane motion capture (scena dialogowa, animacja walki, itp.).</p>
Programowanie sztucznej inteligencji w grach	<p>Implementacja prostego FSM (finite state machine) – zachowanie przeciwnika patrolującego teren.</p> <p>Tworzenie przeciwnika z zachowaniem opartym na behavior tree – wykrywanie gracza, ucieczka, atak.</p> <p>Projektowanie mechaniki gry z wykorzystaniem AI w praktyce – współpraca NPC, kontrola zasobów.</p> <p>Proceduralne rozmieszczanie postaci oraz generowanie mapy terenu w praktyce – losowość vs kontrola.</p> <p>Implementacja sensorów i kolizji do interakcji AI z otoczeniem – pole widzenia, dźwięk.</p> <p>Implementacja systemu dynamicznego poziomu trudności – AI dostosowujące się do umiejętności gracza.</p>

	<p>Symulacja prostych interakcji AI–AI oraz AI–gracz w świecie gry – komunikacja między botami.</p> <p>Budowa prostego bota do gry sieciowej (offline test) – działania drużynowe, wsparcie gracza.</p> <p>Wprowadzenie do sztucznej inteligencji w grach.</p> <p>Algorytmy sztucznej inteligencji w grach.</p> <p>Projektowanie mechaniki i grywalności gry. Sztuczna inteligencja dla ruchu i nawigacji</p> <p>Rozmieszczanie postaci na planszy, generowanie świata i terenu gry.</p> <p>Interakcje między postaciami a światem gry.</p> <p>Wyważanie gry, zarządzanie poziomami trudności rozgrywki.</p> <p>Sztuczna inteligencja w grach sieciowych.</p> <p>Uczenie maszynowe w grach oraz generatywna AI w grach.</p>
Rzeczywistość wirtualna i rozszerzona	<p>Wprowadzenie do technologii VR, AR i MR – przegląd i konfiguracja sprzętu (np. Oculus, HoloLens, ARCore).</p> <p>Komunikacja i integracja z urządzeniami VR/AR/MR w praktyce – API, SDK, OpenXR.</p> <p>Projektowanie sterowania i poruszania się w środowisku VR i MR.</p> <p>Praktyczne kreowanie świata gry w Unity pod kątem immersji – projektowanie przestrzeni, interakcji, dźwięku 3D.</p> <p>Programowanie logiki rozgrywki z uwzględnieniem interfejsów dotykowych, gestów i śledzenia ruchu.</p> <p>Optymalizacja gier immersyjnych – techniki renderingu w czasie rzeczywistym, akceleracja graficzna, ray tracing, DLSS.</p> <p>Praca z technikami Raytracing. Akceleracja grafiki. Deep Learning Super Sampling.</p> <p>Integracja AI z rzeczywistością rozszerzoną w praktyce.</p>
Testowanie gier komputerowych	<p>Analiza cyklu życia gry na wybranym przykładzie (Unity/Unreal) – identyfikacja punktów testowych.</p> <p>Przygotowanie strategii testowania gry 2D/3D – identyfikacja ryzyk, wybór typów testów.</p> <p>Testowanie manualne podstawowych mechanik gry – kontrola postaci, kolizje, UI.</p> <p>Implementacja prostych testów automatycznych (np. w Unity Test Framework, Appium) – regresja, UI.</p> <p>Tworzenie przypadków i scenariuszy testowych dla wybranej gry – testowanie gameplayu.</p> <p>Raportowanie błędów w praktyce, dokumentowanie i zgłaszanie defektów (np. JIRA, TestRail)</p> <p>Testowanie gier mobilnych w emulatorze oraz w urządzeniu fizycznym, testy zgodności.</p> <p>Symulacja projektu mini-kampanii testowej – zespół testujący pełną grę (symulacja procesu QA).</p> <p>Wprowadzenie do testowania gier. Cykl życia testów w produkcji gry i typy testów w grach.</p> <p>Projektowanie niezawodnych gier – aspekty techniczne i użytkowe (UX, UI).</p> <p>Proces QA w praktyce: Zgłaszanie i zarządzanie błędami. Testy regresji, smoke testing.</p> <p>Testowanie logiki gry i mechaniki rozgrywki – kolizje, AI, interakcje.</p> <p>Testowanie gier mobilnych, desktopowych i sieciowych – różnice i wyzwania.</p> <p>Zarządzanie procesem testowania gier – role w zespole QA, workflow, narzędzia zarządzania (np. JIRA).</p>

	Dokumentacja testowa w projektach gier – przypadki testowe, scenariusze testowe, raportowanie błędów.
	Testy wydajności i bezpieczeństwa w grach – optymalizacja, stres testy, ochrona przed exploitami i cheatami.

*\*Treści programowe mogą ulegać modyfikacjom w procesie doskonalenia programów studiów, w celu zapewnienia ich aktualności oraz dostosowania do oczekiwań rynku pracy.*

## IV. PROGRAM STUDIÓW

Specjalności proponowane na kierunku Informatyka

- AI & Data Science
- AI w grafice komputerowej i technologiach multimedialnych
- Architekt rozwiązań IT w chmurze obliczeniowej
- Inżynier aplikacji mobilnych
- Inżynieria programowania procesorów (tylko forma niestacjonarna)
- Inżynieria sieci, systemów i bezpieczeństwa IT
- Programista aplikacji w chmurze z wykorzystaniem AI
- Programista gier komputerowych
- Software Development

### A) PRZYPORZĄDKOWANIE KIERUNKU STUDIÓW DO DYSYCYPLIN NAUKOWYCH

Dyscypliny naukowe	% PUNKTÓW ECTS
Informatyka techniczna i telekomunikacja	90%
Nauki o zarządzaniu i jakości	10%

### B) PODSTAWOWE WSKAŹNIKI ECTS OKREŚLONE DLA PROGRAMU STUDIÓW

Nazwa wskaźnika	Liczba punktów ECTS	
Łączna liczba punktów ECTS, jaką student musi uzyskać w ramach zajęć prowadzonych z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia	STUDIA STACJONARNE 105,2	STUDIA NIESTACJONARNE 76,5
Łączna liczba punktów ECTS przyporządkowana zajęciom kształtującym umiejętności praktyczne	STUDIA STACJONARNE 137	STUDIA NIESTACJONARNE 138,6
Łączna liczba punktów ECTS, jaką student musi uzyskać w ramach zajęć z dziedziny nauk humanistycznych lub nauk społecznych – w przypadku kierunków studiów przyporządkowanych do dyscyplin w ramach dziedzin innych niż odpowiednio nauki humanistyczne lub nauki społeczne	5	
Łączna liczba punktów ECTS przyporządkowana zajęciom do wyboru	71	
Łączna liczba punktów ECTS przyporządkowana praktykom zawodowym	40	

### C) WYMIAR, ZASADY I FORMY ODBYWANIA PRAKTYK ZAWODOWYCH

Zgodnie z Regulaminem studiów Uniwersytetu WSB Merito we Wrocławiu, praktyki zawodowe są obowiązkowe a zasady ich realizacji, treści programowe, metody kształcenia, efekty uczenia się, czy metody weryfikacji osiągnięcia efektów uczenia się etc. określają, podobnie jak w przypadku innych zajęć przewidzianych w programie studiów, karty przedmiotów „Praktyka zawodowa”.

Wymiar praktyk zawodowych dla studiów I stopnia wynosi min. 960 godzin dydaktycznych/24 tygodnie. Praktyki realizowane i zaliczane są w semestrach, w których w programie studiów przewidziany jest przedmiot „Praktyka zawodowa”.

Student organizuje praktyki indywidualnie i jest zobowiązany do złożenia deklaracji, na której pracodawca potwierdza możliwość realizacji programu praktyk w danej placówce/firmie/instytucji. Student ma możliwość skorzystania z pomocy uczelni w przygotowaniu do rekrutacji na praktyki.

Z programu praktyk oraz założeń przedmiotu praktyka zawodowa określonych w karcie przedmiotu a także z Regulaminu praktyk zawodowych wynikają bezpośrednio miejsca, w których realizowane są praktyki. Uczelnia monitoruje miejsca praktyk pod kątem ich przystosowania do osiągania efektów uczenia się przypisanych do praktyk na danym kierunku, możliwości realizacji programu praktyk oraz predyspozycji i preferencji studenta. Praktyki realizowane są w podmiotach, które zapewniają praktykantom opiekuna praktyk, odpowiednie stanowiska pracy odpowiadające zakresowi przyszłej działalności zawodowej (dostęp do komputera, Internetu, profesjonalne oprogramowania etc.).

Procesem organizowania i koordynowania praktyk zajmują się dedykowani poszczególnym kierunkom pracownicy Biura Karier (BK). Nadzór merytoryczny nad realizacją praktyk zawodowych sprawuje opiekun praktyk zawodowych z ramienia Uczelni.

#### **D) SPOSOBY WERYFIKACJI I OCENY EFEKTÓW UCZENIA SIĘ OSIĄGANÝCH PRZEZ STUDENTA W TRAKCIE CAŁEGO CYKLU KSZTAŁCENIA**

Weryfikacja efektów uczenia się stanowi uniwersalny system umożliwiający monitorowanie, sprawdzanie i ocenianie procesu uczenia się studenta w trakcie całego cyklu kształcenia w uczelni. W doborze metod weryfikacji uwzględnia się rodzaje efektów (wiedza, umiejętności, kompetencje społeczne), etapy kształcenia (I stopień, II stopień), kierunki/programy studiów (merytoryka), a także treści (teoretyczne, praktyczne) i formy zajęć (wykład, ćwiczenia, lektorat, konwersatorium, laboratorium, seminarium, praktyka zawodowa). W uczelni przyjmuje się określone sposoby weryfikacji efektów uczenia się: egzaminy (ustne lub pisemne), prace kontrolne, kolokwia, projekty, a także inne aktywności zlecone przez dydaktyka, takie jak np.: ćwiczenia/zadania indywidualne i grupowe, case study, dyskusje dydaktyczne/debaty, prezentacje, gry dydaktyczne. Zróżnicowanie metod weryfikacji pozwalana na całościowe kontrolowanie postępów w procesie uczenia się studenta. Szczegółowe informacje, co do zasad i sposobów weryfikacji i oceny osiągnięcia efektów uczenia się w odniesieniu do poszczególnych kursów/przedmiotów, zamieszczone są w kartach przedmiotów. Poziom osiągnięcia efektów uczenia się studenta dokumentuje się:

- w przypadku wykładu, ćwiczenia, lektoratu, konwersatorium, laboratorium, seminarium – w protokole egzaminu/zaliczenia,
- w przypadku praktyki zawodowej – w protokole zaliczenia praktyki,
- w przypadku egzaminu dyplomowego – w protokole egzaminu dyplomowego.

Sposoby weryfikacji i oceny efektów uczenia się podlegają stałej kontroli Metodyka oraz Wydziałowego Zespołu ds. Jakości Kształcenia.

#### **E) WYKAZ ZAJĘĆ LUB GRUPY ZAJĘĆ Z PRZYPISANIEM PUNKTÓW ECTS**

**Studia stacjonarne**

L.p.	Przedmiot	Semestr	ECTS
1.	BHP	I	0
2.	Deep Impact / Tiefer Einfluss	IV	2
3.	Ekonomia	I	3
4.	Etyka	II	2
5.	Język obcy	I	3
6.	Język obcy	II	3
7.	Język obcy	III	3
8.	Język obcy	IV	3
9.	Metody efektywnego uczenia się	I	2
10.	Microsoft Office 365 i Google Apps SCRIPT	I	3
11.	Podstawy komunikacji społecznej	I	2
12.	Podstawy zrównoważonego rozwoju	II	1
13.	Prawo nowych technologii	IV	2
14.	Proseminarium	IV	1
15.	Repetitorium z matematyki	I	0
16.	Różnice kulturowe	III	3
17.	Wprowadzenie do studiowania na kierunku	I	0
18.	Wychowanie fizyczne	III	0
19.	Wychowanie fizyczne	IV	0
20.	Administrowanie systemem Linux	III	3
21.	Algorytmy i struktury danych	I	5
22.	Architektura systemów komputerowych	III	4
23.	Blockchain i jego zastosowanie	VII	1
24.	Elektronika	II	3
25.	Fizyka	II	3
26.	Funkcjonalność zintegrowanych systemów informatycznych SAP S/4HANA	VI	3
27.	Inżynieria bezpieczeństwa IT	V	2
28.	Inżynieria Internetu rzeczy	IV	3
29.	Inżynieria oprogramowania	V	3
30.	Inżynieria oprogramowania w środowisku MS .NET	VII	2
31.	Inżynieria serwerowych systemów operacyjnych Windows	IV	3
32.	Inżynieria zachowań interdyscyplinarnych w organizacji	VI	2
33.	Komunikacja i efektywność pracy inżynierów	V	2
34.	Matematyka dyskretna	III	4
35.	Matematyka I	I	5
36.	Matematyka II	II	4
37.	MVP – prototypowanie aplikacji z wykorzystaniem AI	II	2
38.	Narzędzia do automatyzacji budowy oprogramowania	IV	3
39.	Platformy Apple	IV	3
40.	Praktyka zawodowa I	VI	20
41.	Programowanie aplikacji low-code/no-code	V	2
42.	Programowanie obiektowe w języku Python	II	3
43.	Programowanie w języku Python	I	5
44.	Projektowanie i tworzenie stron WWW	VI	2
45.	Systemy baz danych	IV	3
46.	Systemy operacyjne	II	3
47.	Techniczny projekt nowatorski (praca przejściowa)	V	3
48.	Techniczny projekt nowatorski (praca przejściowa)	VI	3
49.	Techniczny projekt nowatorski (praca przejściowa)	VII	4
50.	Technologie sieciowe	III	4

51.	Testowanie oprogramowania	II	3
52.	Uczenie maszynowe	IV	4
53.	Współczesne systemy telekomunikacyjne	III	2
54.	Wykorzystanie sztucznej inteligencji w IT	III	3
55.	Zaawansowane techniki programowania w języku Python	III	3
56.	Zarządzanie projektami informatycznymi	V	3
57.	Przedmioty specjalnościowe	V, VI, VII	47
łącznie:			210

### Studia niestacjonarne

L.p.	Przedmiot	Semestr	ECTS
1.	BHP	I	0
2.	Deep Impact / Tiefer Einfluss	IV	2
3.	Ekonomia	I	3
4.	Etyka	II	2
5.	Język obcy	II	6
6.	Język obcy	III	6
7.	Metody efektywnego uczenia się	I	2
8.	Microsoft Office 365 i Google Apps SCRIPT	I	3
9.	Podstawy komunikacji społecznej	I	2
10.	Podstawy zrównoważonego rozwoju	II	1
11.	Prawo nowych technologii	IV	2
12.	Proseminarium	IV	1
13.	Repetitorium z matematyki	I	0
14.	Różnice kulturowe	III	3
15.	Wprowadzenie do studiowania na kierunku	I	0
16.	Administrowanie systemem Linux	III	3
17.	Algorytmy i struktury danych	I	5
18.	Architektura systemów komputerowych	III	4
19.	Blockchain i jego zastosowanie	VII	1
20.	Elektronika	II	3
21.	Fizyka	II	3
22.	Funkcjonalność zintegrowanych systemów informatycznych SAP S/4HANA	VI	3
23.	Inżynieria bezpieczeństwa IT	V	2
24.	Inżynieria Internetu rzeczy	IV	3
25.	Inżynieria oprogramowania	V	3
26.	Inżynieria oprogramowania w środowisku MS .NET	VII	2
27.	Inżynieria serwerowych systemów operacyjnych Windows	IV	3
28.	Inżynieria zachowań interdyscyplinarnych w organizacji	VI	2
29.	Komunikacja i efektywność pracy inżynierów	V	2
30.	Matematyka dyskretna	III	4
31.	Matematyka I	I	5
32.	Matematyka II	II	4
33.	MVP – prototypowanie aplikacji z wykorzystaniem AI	II	2
34.	Narzędzia do automatyzacji budowy oprogramowania	IV	3
35.	Platformy Apple	IV	3
36.	Praktyka zawodowa I	VI	20
37.	Programowanie aplikacji low-code/no-code	V	2
38.	Programowanie obiektowe w języku Python	II	3
39.	Programowanie w języku Python	I	5

40.	Projektowanie i tworzenie stron WWW	VI	2
41.	Systemy baz danych	IV	3
42.	Systemy operacyjne	II	3
43.	Techniczny projekt nowatorski (praca przejściowa)	V	3
44.	Techniczny projekt nowatorski (praca przejściowa)	VI	3
45.	Techniczny projekt nowatorski (praca przejściowa)	VII	4
46.	Technologie sieciowe	III	4
47.	Testowanie oprogramowania	II	3
48.	Uczenie maszynowe	IV	4
49.	Współczesne systemy telekomunikacyjne	III	2
50.	Wykorzystanie sztucznej inteligencji w IT	III	3
51.	Zaawansowane techniki programowania w języku Python	III	3
52.	Zarządzanie projektami informatycznymi	V	3
53.	Przedmioty specjalnościowe	V, VI, VII	47
łącznie:			210