UNIWERSYTET WSB MERITO SZCZECIN

Uniwersytet WSB Merito w Poznaniu

Wydział Ekonomiczny w Szczecinie

Program studiów

Dla kierunku

„Informatyka”

Studia I stopnia -ścieżka online

inżynierskie

Studia: stacjonarne/niestacjonarne
(wskazać formę lub formy studiów)

Profil: praktyczny
(ogólnoakademicki / praktyczny)

Rok akademicki 2026/2027

I. OGÓLNA CHARAKTERYSTYKA KIERUNKU STUDIÓW

nazwa kierunku studiów	**INFORMATYKA**	
Poziom kształcenia (studia pierwszego stopnia / studia drugiego stopnia / jednolite studia magisterskie)	**Studia pierwszego stopnia**	
Profil kształcenia	praktyczny	
Forma studiów stacjonarne/niestacjonarne	**stacjonarne/niestacjonarne**	
Czas trwania studiów (w semestrach)	**7**	
Łączna liczba punktów ECTS dla danej formy studiów.	**210**	
Łączna liczba godzin określona w programie studiów	**Studia stacjonarne** **x**	**Studia niestacjonarne** **2 555**
Tytuł zawodowy nadawany absolwentom	**inżynier**	
Wymiar praktyk zawodowych.	**960 godzin**	
Język prowadzenia studiów	**polski**	
Rok rozpoczęcia cyklu kształcenia	**2026**	

II. EFEKTY UCZENIA SIĘ

symbol efektu	opis efektów uczenia się dla absolwenta studiów I stopnia na kierunku Informatyka	kod uniwersalnej charakterystyki poziomu drugiego dla kwalifikacji na poziomie VI	kod charakterystyki poziomu drugiego dla kwalifikacji na poziomie VI umożliwiających uzyskanie kompetencji inżynierskich
WIEDZA			
Absolwent zna i rozumie:			
Inf_I_W01	w zaawansowanym stopniu zagadnienia z zakresu algorytmów, struktur danych, inżynierii oprogramowania, języków programowania	P6S_WG	
Inf_I_W02	w zaawansowanym stopniu zagadnienia z zakresu architektury systemów komputerowych, systemów operacyjnych, systemów baz danych i hurtowni danych, sieci komputerowych, bezpieczeństwa systemów	P6S_WG	
Inf_I_W03	metody oraz zastosowanie narzędzi wykorzystywanych przy rozwiązywaniu zadań informatycznych	P6S_WG	
Inf_I_W04	w zaawansowanym stopniu zasady komunikacji człowiek-komputer	P6S_WG	
Inf_I_W05	w stopniu podstawowym prawa patentowe, autorskie, o ochronie danych osobowych oraz zagrożenia związane z przestępczością elektroniczną jak również zapisy kodeksów etycznych	P6S_WK	
Inf_I_W06	metody i zastosowanie narzędzi pozwalających opisywać procesy i zjawiska społeczne oraz gospodarcze	P6S_WG	
Inf_I_W07	podstawowe zasady organizowania i rozwoju form indywidualnej przedsiębiorczości	P6S_WK	P6S_WK
Inf_I_W08	podstawowe koncepcje dotyczące opisu i wyjaśniania rzeczywistości ekonomicznej	P6S_WG	
Inf_I_W09	metody matematyczne i statystyczne wykorzystywane w informatyce	P6S_WG	
Inf_I_W10	zasady etyki w biznesie	P6S_WK	P6S_WK
Inf_I_W11	zagadnienia związane z cyklami życia systemów informatycznych w tym oprogramowania	P6S_WG	P6S_WG
Inf_I_W12	ogólne zagadnienia nt algorytmów i ich oceny złożoności, paradygmatów programowania, podstawowych narzędzi informatycznych	P6S_WG	P6S_WG
Inf_I_W13	standardy i normy stosowane w przesyłaniu i przetwarzaniu danych oraz w inżynierii oprogramowania	P6S_WG	P6S_WG
Inf_I_W14	w stopniu zaawansowanym zagadnienia w zakresie pozyskiwania, przechowywania i przetwarzania danych multimedialnych	P6S_WG	
UMIEJĘTNOŚCI			
Absolwent potrafi:			

Inf_I_U01	pozyskiwać i integrować informacje z literatury oraz innych źródeł, dokonywać ich oceny oraz krytycznej analizy.	P6S_UU	
Inf_I_U02	porozumiewać się w środowisku zawodowym językiem ojczystym i językiem angielskim, na poziomie B2 Europejskiego Systemu Opisu Kształcenia Językowego, używając specjalistycznej terminologii oraz wykorzystując zaawansowane narzędzia informatyczne do komunikacji	P6S_UK	
Inf_I_U03	modelować i projektować systemy informatyczne, opisywać wymagania funkcjonalne i niefunkcjonalne, oceniać architekturę oprogramowania	P6S_UW	P6S_UW
Inf_I_U04	programować aplikacje użytkowe, formułować algorytmy, dokonywać właściwego doboru języka programowania, projektować graficznie interfejs użytkownika, dokumentować i systematycznie testować wytwarzane oprogramowanie, programować aplikacje WWW	P6S_UW	P6S_UW
Inf_I_U05	projektować relacyjne bazy danych, przetwarzać i analizować dane zgromadzone w bazach danych, programować aplikacje korzystające z baz danych	P6S_UW	P6S_UW
Inf_I_U06	montować i dokonywać obróbki danych multimedialnych oraz wykorzystywać je w aplikacjach użytkowych	P6S_UW	P6S_UW
Inf_I_U07	wykonywać typowe zadania związane z utrzymaniem systemów komputerowych, sieci komputerowych, zapewnianiem bezpieczeństwa systemów	P6S_UW	P6S_UW
Inf_I_U08	przygotować i wygłosić wystąpienie publiczne w języku polskim i języku angielskim, dotyczącej zagadnień z zakresu informatyki, z wykorzystaniem wiedzy zawodowej, terminologii fachowej oraz informacji pochodzących z różnych źródeł, a także uczestniczyć w debacie	P6S_UK	
Inf_I_U09	przygotować opracowanie problemów, także nietypowych oraz złożonych, dla informatyki z wykorzystaniem wybranej literatury przedmiotu i innych udokumentowanych źródeł informacji oraz baz danych lub informacji w języku polskim i języku angielskim	P6S_UW P6S_UK	
Inf_I_U10	planować i przeprowadzać eksperymenty obliczeniowe oraz symulacje komputerowe, z wykorzystaniem narzędzi informatycznych, interpretować uzyskane wyniki i wyciągać wnioski	P6S_UW	P6S_UW
Inf_I_U11	wykorzystywać do formułowania i rozwiązywania problemów informatycznych, także złożonych i nietypowych, właściwe metody analityczne, symulacyjne oraz eksperymentalne	P6S_UW	P6S_UW
Inf_I_U12	przy formułowaniu i rozwiązywaniu zadań informatycznych dostrzegać ich aspekty ekonomiczne, prawne i inne związane ze środowiskiem, w którym wdraża się te zadania	P6S_UW	P6S_UW
Inf_I_U13	pracować w środowisku przemysłowym stosując zasady bezpieczeństwa związane z tą pracą	P6S_UW	P6S_UW
Inf_I_U14	dokonać wstępnej analizy ekonomicznej podejmowanych działań inżynierskich	P6S_UW	P6S_UW
Inf_I_U15	w typowym zakresie technicznym obsługiwać systemy informatyczne działające w przedsiębiorstwach	P6S_UW	P6S_UW

Inf_I_U16	rozwiązywać typowe problemy informatyczne pojawiające się w przedsiębiorstwach	P6S_UW	P6S_UW
Inf_I_U17	wykorzystywać normy związane zarówno z przesyłaniem, przetwarzaniem danych jak i przygotowaniem oraz zarządzaniem projektami informatycznymi	P6S_UW	P6S_UW
Inf_I_U18	doskonalić się przez całe życie, poprzez planowanie i realizowanie pozyskiwania nowej wiedzy i umiejętności	P6S_UU	
Inf_I_U19	pracować i współdziałać w różnych grupach społecznych i w różnych rolach	P6S_UO	
Inf_I_U20	wybierać priorytety służące realizacji określonego przez siebie lub innych celu bądź zadania	P6S_UO	
KOMPETENCJE SPOŁECZNE			
Absolwent jest gotów do:			
Inf_I_K01	uznania konieczności uczenia się przez całe życie oraz krytycznej oceny posiadanej wiedzy i odbieranych treści	P6S_KK	
Inf_I_K02	identyfikowania i rozstrzygania dylematów związanych z wykonywaniem zawodu	P6S_KR	
Inf_I_K03	myślenia i działania w sposób przedsiębiorczy, także poprzez inicjowania działań na rzecz interesu publicznego	P6S_KO	
Inf_I_K04	uznania skutków pozatechnicznych swojej działalności	P6S_KO	
Inf_I_K05	odpowiedzialnego postępowania, poprzez propagowanie i przestrzeganie zasad etyki zawodowej	P6S_KR	
Inf_I_K06	komunikatywnego przedstawiania i wyjaśniania osiągnięć informatyki szerokiemu gronu odbiorców.	P6S_KR	

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

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

Symbol efektu	Inf_I_W01	Inf_I_W02	Inf_I_W03	Inf_I_W04	Inf_I_W05	Inf_I_W06	Inf_I_W07	Inf_I_W08	Inf_I_W09	Inf_I_W10	Inf_I_W11	Inf_I_W12	Inf_I_W13	Inf_I_W14	Inf_I_U01	Inf_I_U02	Inf_I_U03	Inf_I_U04	Inf_I_U05	Inf_I_U06	Inf_I_U07	Inf_I_U08	Inf_I_U09	Inf_I_U10	Inf_I_U11	Inf_I_U12	Inf_I_U13	Inf_I_U14	Inf_I_U15	Inf_I_U16	Inf_I_U17	Inf_I_U18	Inf_I_U19	Inf_I_U20	Inf_I_K01	Inf_I_K02	Inf_I_K03	Inf_I_K04	Inf_I_K05	Inf_I_K06
BHP																										X											X		
Komunikacja społeczna						X				X					X	X						X																	X
Matematyka dla inżynierów			X						X																X							X	X						
Matematyka dyskretna			X						X																X							X	X						
Narzędzia informatyki			X													X								X	X									X		X			X
Wprowadzenie do informatyki			X									X													X														
Podstawy programowania	X		X									X						X																X					X
Podstawy ekonomii						X		X		X																X		X					X						
Metodyka pracy projektowej						X																X																	
Wyzwania rynku pracy							X									X																	X	X	X		X		X	
Podstawy zarządzania						X										X										X									X	X	X		
Algorytmy i struktury danych	X		X						X			X			X			X															X	X					
Architektura komputerów		X																X			X								X	X	X			X					
Systemy operacyjne		X																			X								X	X	X			X					
Podstawy elektrotechniki												X	X		X									X	X		X						X		X				
Probabilistyka i statystyka			X			X			X																X							X							
Programowanie obiektowe	X																	X																X					
Sieci komputerowe		X			X								X					X			X								X	X	X				X		X		
Bazy danych		X																	X					X								X							
Laboratorium inżynierskie		X	X			X					X				X						X			X	X		X					X				X			
Język angielski																X																X	X	X					
Programowanie aplikacji internetowych	X												X					X																X					
Wprowadzenie do baz danych		X																	X					X								X							
Zarządzanie projektami informatycznymi			X					X			X						X							X								X	X						
Laboratorium nowych technologii	X			X								X	X					X							X		X		X				X			X		X	
Metody obliczeniowe									X			X												X	X														
Programowanie zaawansowane	X												X				X	X						X								X		X					
Podstawy ochrony danych		X	X	X									X								X		X			X			X		X								
Analiza i projektowanie systemów informatycznych	X										X						X	X						X		X					X				X				X
Przetwarzanie danych multimedialnych				X										X				X		X					X														
Software Engineering			X								X		X		X	X	X	X			X							X							X		X			
Testowanie oprogramowania				X	X	X								X										X															
Programowanie w zastosowaniach	X	X	X						X			X			X		X		X				X	X	X							X								
Cultural Differences						X				X						X																	X	X		X				
Seminarium dyplomowe			X		X										X	X						X	X										X					X	X
Praktyka zawodowa					X											X										X			X	X		X	X	X	X	X		X	

Symbol efektu	Inf_I_W01	Inf_I_W02	Inf_I_W03	Inf_I_W04	Inf_I_W05	Inf_I_W06	Inf_I_W07	Inf_I_W08	Inf_I_W09	Inf_I_W10	Inf_I_W11	Inf_I_W12	Inf_I_W13	Inf_I_W14	Inf_I_U01	Inf_I_U02	Inf_I_U03	Inf_I_U04	Inf_I_U05	Inf_I_U06	Inf_I_U07	Inf_I_U08	Inf_I_U09	Inf_I_U10	Inf_I_U11	Inf_I_U12	Inf_I_U13	Inf_I_U14	Inf_I_U15	Inf_I_U16	Inf_I_U17	Inf_I_U18	Inf_I_U19	Inf_I_U20	Inf_I_K01	Inf_I_K02	Inf_I_K03	Inf_I_K04	Inf_I_K05	Inf_I_K06
Specjalność: Cyberbezpieczeństwo																																								
Bezpieczeństwo oprogramowania		X	X								X		X				X				X				X														
Wprowadzenia do chmur obliczeniowych	X	X	X										X				X		X		X							X		X									X
Bezpieczeństwo i ochrona danych	X		X						X			X	X		X									X	X						X								
Elementy kryptografii	X		X										X		X									X	X						X				X				
Systemy zarządzania bezpieczeństwem informacji			X		X																X																		
Podstawy monitoringu systemów i aplikacji		X	X										X		X									X						X	X			X					
Wykrywanie i analiza zagrożeń w sieci		X											X								X										X								
Ochrona danych w chmurze		X	X		X																X														X			X	
Zarządzanie projektami bezpieczeństwa IT		X	X		X							X			X		X	X			X					X		X		X									
Specjalność: Programowanie																																								
Testowanie i jakość oprogramowania					X																			X																
Projektowanie i implementacja aplikacji rozproszonych	X		X						X																														
Programowanie w zespole – studium przypadku	X		X											X			X								X	X					X								
Wzorce projektowe	X								X								X								X						X								
Zaawansowana inżynieria kodu			X						X																X														
Integracja oprogramowania z platformą Azure	X	X	X										X						X												X								
Programowanie aplikacji internetowych MVC API	X								X				X					X							X														
Projekt systemu informatycznego	X								X								X	X							X														
Algorytmizacja	X								X			X						X							X														
Specjalność: Sztuczna inteligencja																																								
Podstawy uczenia maszynowego	x			x								x	x		x													x			x								x
Systemy wspomagania decyzji	x			x	x				x					x	x					x				x	x	x									x				x
Projekt zespołowy - rozwiązania AI							X										X	X	X												X								
Systemy ekspertowe																	X		X												X								
Zaawansowane uczenie maszynowe	x		x								x		x				x	x						x	x												x		x
Wizja komputerowa																		X													X								
Sieci neuronowe i głębokie uczenie	x					x			x		x	x					x		x		x					x				x							x		x
Przetwarzanie języka naturalnego				x	x				x			x			x									x	x	x									x				x
Optymalizacja stochastyczna			X			X			X									X												X								

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

ZAJĘCIA LUB GRUPY ZAJĘĆ	TREŚCI PROGRAMOWE
Matematyka dla inżynierów/ Matematyka dyskretna	Pojęcie bazy i wymiaru przestrzeni. Niezależność liniowa wektorów i jej badanie. Ortogonalność i równoległość wektorów. Macierze. Układy równań i nierówności liniowych i podstawowe metody ich rozwiązywania. Podstawy geometrii przestrzeni trójwymiarowej. Równanie prostej i płaszczyzny. Przedstawienie parametryczne krzywej. Zastosowania w grafice komputerowej. Funkcje elementarne, wykresy i ich własności. Granice ciągów liczbowych (potęgowych, wykładniczych, pierwiastkowych, zbieżnych do liczby e); twierdzenie o trzech ciągach. Kryteria zbieżności szeregów liczbowych. Granice funkcji. Pochodna funkcji jednej zmiennej. Przedziały monotoniczności oraz ekstrema funkcji jednej zmiennej. Całka nieoznaczona i metody jej obliczania. Całka oznaczona. Elementy logiki matematycznej. Prawa rachunku zdań. Tautologie. Funkcja zdaniowa. Rachunek kwantyfikatorów. Relacje. Podstawowe typy relacji. Relacja porządkująca. Relacja równoważności. Elementy kombinatoryki i ich zastosowanie. Podstawowe techniki zliczania. Zastosowanie zasady włączania i wyłączania oraz zasady szufladkowej Dirichleta. Rekurencja i zasada indukcji matematycznej. Kryteria poprawności algorytmów rekurencyjnych. Zasada indukcji matematycznej. Grafy nieskierowane i algorytmy przeszukiwania grafu.
Probabilistyka i statystyka	Pojęcie i własności prawdopodobieństwa. Zmienna losowa i jej własności. Rozkład normalny prawdopodobieństwa. Wprowadzenie do wnioskowania statystycznego. Próba statystyczna i rozkłady z próby. Ustalanie minimalnej liczebności próby. Opisowa analiza struktury zjawisk masowych. Korelacja i regresja liniowa. Analiza szeregów czasowych. Wyznaczanie trendu liniowego. Estymacja parametrów populacji. Przedziały ufności. Weryfikacja hipotez statystycznych.
Narzędzia informatyki	Arkusze kalkulacyjne, Wykorzystanie narzędzi klasy content curation do selekcjonowania, wyszukiwania, gromadzenia, współdzielenia informacji i komunikacji. Wykorzystanie czytników RSS do selekcjonowania źródeł i pozyskiwania informacji. Wykorzystanie zaawansowanych funkcji wyszukiwarek internetowych do skutecznego przeszukiwania zasobów sieci. Wykorzystanie narzędzi wyszukiwania treści w dokumentach. Edycja dokumentów tekstowych. Tworzenie prezentacji z wykorzystaniem elementów wbudowanych w narzędzie oraz zaczerpniętych z zewnątrz. Bazy danych – definicje, struktura, zapytania na potrzeby analiz danych.
Wprowadzenie do informatyki	Dziedziny informatyki i ich obszary zastosowań. Systemy informacyjne vs. systemy informatyczne. Model przepływu informacji i systemy przetwarzania danych. Reprezentacja liczb w komputerze, arytmetyka binarna. Modele logiczne komputera i ich klasyfikacja. Elementy historii informatyki i budowa współczesnych urządzeń komputerowych. Algorytmika. Podstawowe instrukcje na przykładzie języka C/C++. Podstawowe typy danych. Programowanie strukturalne a obiektowe. Metaprogramowanie przy zastosowaniu szablonów – omówienie biblioteki STL. Klasyfikacja oprogramowania komputerów. Zadania oprogramowanie systemowego. Typy oprogramowanie użytkowego. Licencjonowanie oprogramowania. System informacyjny w zarządzaniu. Cechy charakterystyczne systemów klasy ERP jako zintegrowanego systemu informacyjnego. Klasyfikacja i charakterystyka innych systemów do zarządzania przedsiębiorstwami oraz realizacji e-gospodarki. Definicja i cechy społeczeństwa informacyjnego. Sztuczna inteligencja i informatyka przyszłości.

Podstawy programowania	Wprowadzenie, środowisko programistyczne, struktura elementarnego programu, podstawowe typy, zmienne i instrukcje. Typy języka C#, zmienne, instrukcje, operatory, wyrażenia, platforma .NET, zarządzanie pamięcią operacyjną. Tablice, struktury, we/wy w konsoli, obsługa błędów we/wy, wyjątki, kod nienadzorowany. Programowanie strukturalne, podprogramy, przekazywanie parametrów, przestrzenie nazw. Pliki/strumienie, operacje na plikach i katalogach w systemie Windows, zasoby niezarządzane przez .NET. Manipulowanie łańcuchami tekstu, dynamiczny przydział pamięci, preprocesor, dokumentowanie i testowanie programów. Programy GUI, model programowania sterowanego zdarzeniami, standardowe elementy dialogowe, model SDI.
Algorytmy i struktury danych	Wprowadzenie do algorytmiki: problem a algorytm, metody zapisu algorytmów, ocena wydajności, klasyfikacja złożoności problemów i algorytmów. Pojęcie rekurencji, przykłady algorytmów rekurencyjnych, ocena i porównanie wydajności algorytmów iteracyjnych i rekurencyjnych rozwiązujących ten sam problem, przekształcanie do postaci iteracyjnej. Rekurencyjne struktury danych: listy, kolejki, drzewa. Algorytmy sortowania: definicja problemu sortowania, miary efektywności, metody proste, metody ulepszone, zasady doboru metod, ocena wydajności, sortowanie zewnętrzne. Grafy i algorytmy grafowe, metody reprezentacji grafów, przeszukiwanie grafów, podstawowe problemy grafowe i ich znaczenie praktyczne. Algorytmy wyszukiwania, przeszukiwanie tekstów. Zaawansowane techniki programowania, algorytmy zachłanne, programowanie dynamiczne, kompresja i szyfrowanie danych. Algorytmy numeryczne, specyfika obliczeń numerycznych, typy danych, metody konstruowania algorytmów, optymalizacja, aproksymacja. turystyce;
Programowanie obiektowe	Paradygmat programowania zorientowanego obiektowo: abstrakcja, hermetyzacja, polimorfizm statyczny/dynamiczny, dziedziczenie. Deklaracja klasy. Pola i metody. Klasy a obiekty. Klasy: stałe, pola, metody, konstruktory, destruktor, modyfikatory, właściwości. Wsparcie dla programowania zorientowanego obiektowo w Visual Studio. Konstruowanie hierarchii klas, polimorfizm statyczny i dynamiczny, operatory, indeksatory, delegacje. Wsparcie dla programowania zorientowanego obiektowo w Visual Studio. Programowanie z wykorzystaniem obiektów: struktury, typy wartościowe i referencyjne, pakowanie, odpakowywanie. Interfejsy: definicja, wykorzystanie, kontrakt, definicje, deklaracje, modyfikatory, dziedziczenie, składowe, metody, właściwości, zdarzenia, indeksatory. Programowanie w dużej skali: typy generyczne, biblioteki, moduły, atrybuty. Metodyka obiektowa: zasady projektowania klas, stosowanie dziedziczenia, analiza obiektowa, proces tworzenia oprogramowania. Podstawy języka UML 2.X, modelowanie struktury logicznej systemu: klasy i ich diagramy, związki między klasami, instancje obiektów. Wsparcie w Visual Studio i ArgoUML. Inne popularne języki obiektowe. C++ jako przejściowy język proceduralno-obiektowy: łączenie paradygmatów, szablony, dziedziczenie wielokrotne. Java a C#: podobieństwa i różnice. Odmienna koncepcja – Python.
Programowanie aplikacji internetowych	Wprowadzenie - prezentacja metod programowania stron internetowych. Język HTML (tworzenie dokumentów, HTML) Język CSS – Kaskadowe arkusze stylów (dołączanie arkuszy, CSS, selektory, efekty wizualne) Język PHP (składnia, programowanie zorientowane obiektowo, połączenia z bazą danych, obsługa formularzy, obsługa sesji i ciasteczek, frameworki) Język Javascript, (składnia, obiekty, obsługa DOM - Document Object Model, dołączanie skryptów, używanie bibliotek. Query, efekty wizualne biblioteki jQuery UI, AJAX, frameworki)

Programowanie zaawansowane / Programowanie w zastosowaniach	Podejście koncepcyjne do tworzenia oprogramowania: Architektura klient-serwer (C#) Wydzielanie kodu - biblioteki narzędziowe Testy jednostkowe Testy logiki biznesowej Dziennik zdarzeń aplikacji Metody przechowywania danych (C# / MySql*/Oracle*/MSSql*) Architektura serwera (Java) Frontend (dowolny wybrany przez prowadzącego)

Architektura komputerów	Zarys historii rozwoju systemów komputerowych. Budowa i zasada działania systemu komputerowego. Zasady działania podstawowych elementów komputera (pamięć operacyjna, pamięci masowe, podstawowe urządzenia I/O). Budowa i zasady działania procesora na przykładzie rodziny procesorów x86. Przegląd technik przyspieszania pracy procesorów: pamięć cache, architektura RISC, przetwarzanie potokowe, architektura superskalarna, procesory wielordzeniowe. Komputerowe reprezentacje danych. Podstawowe operacje arytmetyczno-logiczne. Wprowadzenie do techniki cyfrowej. Budowa i zastosowanie półsumatora, sumatora, inwertera, rejestrów zatrzaskowych, rejestru przesuwającego. Programowanie w języku asemblera (lista rozkazów, tryby adresowania, instrukcje sterujące, podprogramy, system przerwań). Budowa i elementy składowe komputera klasy PC. Sposoby identyfikacji awarii podstawowych elementów komputera (pamięć, dyski, procesor, zasilanie)
Systemy operacyjne	Wprowadzenie do systemów operacyjnych Powłoka systemu operacyjnego i środowiska graficzne System plików. Zarządzania procesami. Zarządzanie pamięcią operacyjną. Dobór systemu operacyjnego do potrzeb klienta. Ochrona i bezpieczeństwo.
Sieci komputerowe	Wprowadzenie (motywacja, rys historyczny, podstawowa terminologia, klasyfikacje sieci komputerowych , typowe usługi sieciowe, podstawowe modele komunikacji i rodzaje transmisji). Wybrane zagadnienia z zakresu przesyłania danych (kodowanie bitów, wykrywanie błędów transmisji, zapewnienie niezawodności transmisji, sterowanie dostępem do współdzielonego medium komunikacyjnego). Architektury sieci (idea modelu warstwowego, model odniesienia ISO/OSI, model protokołów Internetu, stosy protokołów, adresacja fizyczna i logiczna, topologie sieci komputerowych). Standardy sieci lokalnych (Ethernet, TokenRing, FDDI, ATM) Urządzenia sieciowe (domeny kolizyjne i rozgłoszeniowe, segmentacja ruchu, wzmacniak, koncentrator, most, przełącznik, router, brama sieciowa) Wirtualne sieci lokalne (zasada działania, zastosowanie, metody definiowania przynależności). Sieci bezprzewodowe (zasada działania, tryby pracy, zagrożenia) Usługi. Przegląd wybranych zagadnień z zakresu bezpieczeństwa sieci komputerowych.
Podstawy Ochrony danych	Bezpieczeństwo, przestępstwa, środki ochrony. Polityka bezpieczeństwa. Normy etyczne odnoszące się do korzystania z sieci komputerowych. Kontrola dostępu do systemu informatycznego. Dziennik zdarzeń. Programy szkodliwe. Składowanie danych. Zapory sieciowe. Systemy wykrywania włamań i systemy prewencyjne Zasilacze awaryjne. Steganografia i znakowanie cyfrowe plików. Podstawowy ochrony kryptograficznej; podpis cyfrowy i infrastruktura klucza publicznego.
Analiza i projektowanie systemów informatycznych	Modelowanie środowiska. Zarządzanie projektem. Tworzenie dokumentacji projektowej. Etapy projektu i zasady przechodzenia do kolejnych faz projektu. Modelowanie wymagań systemu na poziomie projektowym. Modelowanie aspektów strukturalnych systemu. Doskonalenie modelu logicznego struktury systemu. Budowa diagramów klas i diagramów obiektów. Modelowanie dynamiki systemu. Zachowanie. Modelowanie systemu z perspektywy zachowań. Diagramy stanów. Współczesne architektury systemów informatycznych. Tworzenie struktury pakietowej z wykorzystaniem abstrakcji. Charakterystyka fazy implementacji. Testowanie systemu. Zagadnienie ewolucji oprogramowania i refaktoryzacji kodu.
Bazy danych / Wprowadzenie do baz danych	Relacyjne bazy danych: motywacje i pojęcia podstawowe, serwer bazy danych Modelowanie danych: diagramy ER Relacyjny model danych, transformacja diagramów ER do schematów relacji Podstawy języka zapytań SQL: tworzenie relacji, proste zapytania, prosta modyfikacja danych. Normalizacja relacji Zarządzanie współbieżnym dostępem użytkowników do danych, transakcje, poziomy izolacji transakcji. Struktury fizyczne w bazach danych: plik sekwencyjny, plik posortowany, plik indeksowy, indeks wielopoziomowy, indeks haszowany. Język SQL: projekcja, selekcja, połączenia, operacje mnogościowe, podzapytania, wartości puste. Język SQL: ograniczenia integralnościowe, zarządzanie strukturą relacji, perspektywy, indeksy. Język SQL: zarządzanie transakcjami. Język SQL: zarządzanie kontami i uprawnieniami użytkowników Proceduralne rozszerzenia języka SQL: język programowania PL/pgSQL, kursory, wyjątki, funkcje składowane, procedury wyzwalane. Tworzenie aplikacji dla baz danych. Wdrażanie systemów baz danych

Zarządzanie projektami informatycznymi	Projekt informatyczny. Projekt jako realizacja strategii informatyzacji przedsiębiorstwa i instytucji. Cykl życia projektu informatycznego. Struktury realizacyjne. Aspekty projektu informatycznego wg metodyki PRINCE2. Studia przypadkuEtapy i fazy projektu informatycznego Planowanie zadań. Monitorowanie postępu prac. Informatyczne narzędzia wspomagające realizacje projektu. Tendencja rozwojowe w zakresie projektów informatycznych. Baza wiedzy projektu - zarządzanie wiedzą w projektach informatycznych,
Przetwarzanie danych multimedialnych	Obszary zastosowań multimediów. Interaktywność multimediów. Ochrona własności intelektualnej. Aspekty wytwarzania aplikacji multimedialnych. Operowanie obrazem. Użyteczność aplikacji multimedialnych i jej analizowanie. Podstawy kompresji multimediów, standaryzacja. Narzędzia obróbki i integracji obiektów multimedialnych. Tworzenie aplikacji multimedialnej.
Software Engineering	Organizacja procesu wytwarzania oprogramowania, modele cyklu życia oprogramowania, fazy procesu wytwarzania oprogramowania, dobór w zespoły programistyczne, określenie tematu projektu, wybór technologii i narzędzi, planowanie projektu. Specyfikacja wymagań funkcjonalnych i poz a funkcjonalnych. Modelowanie systemu informatycznego. Implementacja systemu, systemy kontroli wersji, zarządzanie konfiguracją, budowanie oprogramowania. Wzorce projektowe, koncepcja, zasady stosowania, omówienie często wykorzystywanych wzorców projektowych, rozpoznawanie wzorców w kodzie i adaptowanie we własnych implementacjach. Testowanie oprogramowania, automatyzacja procesu testowania, testy jednostkowe, testy akceptacyjne i wydajnościowe.
Laboratorium inżynierskie	Fizyka. Pomiary prądu elektrycznego, dobór odpowiednich parametrów zasilacza. Technologia. Zaprojektowanie i wykonanie elementu z wykorzystaniem mini obrabiarki CNC. Arduino. Wykonanie układów elektronicznych i programów z wykorzystaniem zestawu Arduino. Raspberry Pi. Programowanie. Sieci komputerowe. Wykonanie warstwy fizycznej sieci komputerowej, pomiar, spawanie okablowania światłowodowego.
Laboratorium nowych technologii	Programowanie obrabiarki CNC z wykorzystaniem 5 osi roboczych, jako przykład wykorzystania znajomości zagadnień IT w branży innej niż informatyczna. Wykorzystanie skanera 3D, do tworzenia rysunków trójwymiarowych i przygotowania plików *.stl do wydruku 3D. Zapoznanie z budową i zasadą działania drukarki 3d wykorzystującej żywice fotoutwardzalne. Programowanie układów automatyki na bazie Lego Mindstorms. Zapoznanie z budową, zasadą działania oraz integracja czytnika RFID z bazą danych lub arkuszem kalkulacyjnym. Zapoznanie z budową i zastosowaniem eyetrackera. Analiza stron internetowych i reklam pod kątem przekazu informacji i prawidłowości umiejscowienia kluczowych elementów. Wykonanie spoin z wykorzystaniem symulatora spawania jako przykład przemysłowego zastosowania Augmented Reality.
Metody obliczeniowe	Systemy liczbowe i błędy w obliczeniach. Rozwiązywanie równań nieliniowych. Mnożenie macierzy. Rozwiązywanie układów równań liniowych. Interpolacja wielomianowa. Całkowanie numeryczne. Generatory liczb pseudolosowych. Metoda Monte Carlo.
Podstawy elektrotechniki	Podstawowe pojęcia i prawa elektrotechniki. Obwody elektryczne. Elementy i podzespoły elektroniczne. Metody analizy obwodów. Pomiar i analiza sygnałów elektrycznych. Zagadnienia bezpieczeństwa w elektrotechnice. Wprowadzenie do elektroniki cyfrowej.

Testowanie oprogramowania	Wprowadzenie do kluczowych koncepcji testowania. Podstawowe etapy procesu testowania. Testowanie w ramach cyklu życia oprogramowania. Wybrane techniki projektowania testów. Priorytetyzacja przypadków testowych. Wprowadzenie do testowania atrybutów jakościowych. Wybrane elementy testowania automatycznego przy użyciu Selenium lub alternatywnej platformy testowej. Wybrane elementy testów bezpieczeństwa. Wybrane metody zarządzania procesem testowania. Nietechniczne aspekty testowania.
Podstawy ekonomii	Wprowadzenie do ekonomii. Gospodarka rynkowa. Elastyczność cenowa popytu i przychody przedsiębiorstw. Podstawy decyzji ekonomicznych producenta. Koszty produkcji. Maksymalizacja zysku w konkurencji doskonałej i monopolu w gospodarce rynkowej. Produkcja i popyt globalny – podstawowe pojęcia i zależności. Wzrost gospodarczy. Polityka fiskalna. Pieniądz i polityka pieniężna Gospodarka otwarta wahania koniunkturalne. Rynek pracy

Komunikacja społeczna	Proces komunikowania się. Modele komunikowania. Kompetencja komunikacyjna. Komunikacja werbalna i niewerbalna. Funkcje języka. Sztuka publicznego przemawiania. Autoprezentacja. Retoryka. Erystyka. Komunikacja w organizacji Komunikacja interpersonalna i grupowa. Komunikacja międzykulturowa.
Przedsiębiorczość	Przedsiębiorczość w Polsce i na świecie; Osobowość przedsiębiorcy; Warsztaty twórczo-analityczne dot. pomysłu na własną działalność; Proces kształtowania i wprowadzania na rynek organizacji; Relacje organizacji z otoczeniem; Źródła finansowania start-up'ów, Biznesplan; Zarządzanie kryzysem w organizacji; Benchmarking, Formalno-prawne aspekty prowadzenia działalności gospodarczej
Podstawy zarządzania	Wprowadzenie do zarządzania. Środowiskowy kontekst zarządzania. Funkcja zarządzania: planowanie Funkcja zarządzania: organizowanie Funkcja zarządzania: motywowanie Funkcja zarządzania: kontrola
Język obcy (angielski)	Poszukiwanie pracy/stażu/praktyk – analiza ofert pracy, szukanie rekomendacji u znajomych, słownictwo związane z oferowanymi warunkami pracy (słuchanie, czytanie); gramatyka – okresy warunkowe; Rozmowa kwalifikacyjna – zachowania związane z rozmową kwalifikacyjną, szukanie silnych i słabych stron, mowa ciała, zarządzanie stresem, pisanie efektywnego CV (słuchanie, czytanie); gramatyka – czasy przeszłe, opisywanie przeszłych doświadczeń; Moje zadania w pracy, podział odpowiedzialności, struktura firmy, pisanie korespondencji formalnej (słuchanie, czytanie); gramatyka – rzeczowniki policzalne i niepoliczalne; Onboarding – wprowadzenie nowej osoby w struktury i zasady firmy; rozumienie regulaminów oraz instrukcji (czytanie, słuchanie), gramatyka – strona bierna; Organisation – organizacja; Funkcjonowanie firmy: obsługa klienta, Internet, technologie, reklama, zarządzanie projektem i komunikacja w firmie (czytanie i słuchanie), gramatyka – czasy Perfect; Prezentacje – techniki prezentacyjne i oddziaływanie na słuchacza, prezentowanie danych, skuteczny początek i mocne zakończenie prezentacji (czytanie i słuchanie); gramatyka – relative clauses; Techniki i strategie negocjacyjne, wady i zalety outsourcingu, negocjacje z klientem, wyrażanie argumentów i opinii (czytanie i słuchanie); gramatyka – mowa zależna; Podróże służbowe – organizacja podróży, środki transportu, rezerwowanie noclegu i reagowanie na problemy, korespondencja

	związana z podróżą służbową (czytanie i słuchanie); gramatyka – wyrażanie przyszłości
Język obcy (niemiecki)	Poszukiwanie pracy/stażu/praktyk – analiza ofert pracy, rekomendacje, słownictwo warunków pracy, gramatyka – tryb rozkazujący; Rozmowa kwalifikacyjna – zachowania, mocne i słabe strony, mowa ciała, zarządzanie stresem, pisanie CV, gramatyka – zdania pytające; Nowa praca – orientacja w nowym miejscu pracy, wideokonferencje, analiza e-maili, gramatyka – czasownik oddzielnie złożony; Spotkanie z klientami – ustalanie terminu i agendy, gramatyka – czasowniki modalne; Prezentacja produktów firmy – prezentacja usług i produktów, rozmowy pracowników, gramatyka – odmiana przymiotnika; Obsługa klienta – zapytania, rozmowy telefoniczne, gramatyka – Konjunktiv II; Zamawianie towarów – rozmowa o zamówieniu, gramatyka – liczby porządkowe; Zakończenie praktyk – analiza zaświadczenia, gramatyka – spójniki weil, denn, dass
Cultural Differences (ang.)	Introduction to Culture. Culture and its main characteristics. 4 cultural dimensions of G.Hofstede. Intercultural verbal and nonverbal communication. Culture Shock & Developing cross cultural competencies. Building successful intercultural relationship based on trust.

Metodyka pracy projektowej	Techniki studiowania; Tworzenie prezentacji; Wystąpienia publiczne i autoprezentacja; Współpraca w zespole; Umiejętność pisania; Praca metodą projektu; Design Thinking; Metodyka projektu
Seminarium dyplomowe	Przedmiot, cel i obszar badań w nauce; Główne procesy tworzenia wiedzy; Źródła wiedzy naukowej – przykłady; Etapy postępowania badawczego; Metody i techniki badań – przykłady praktyczne; Procedura postępowania badawczego – proces preparacji pracy projektowej; Motywy i racje podjęcia danego tematu; Harmonogram treści prac projektowych/konstrukcja projektu; Podstawowe sposoby przetwarzania materiałów będące funkcją procesów myślowych; Metody gromadzenia aktualnej wiedzy i jej przetwarzanie – z podziałem na członków grupy; Literatura przedmiotu – rodzaje publikacji ze względu na ich wartość naukową; Zasady sporządzania bibliografii – przykłady; Wymogi metodyczne stawiane pracom projektowym; Zasady wnioskowania; Ocena merytoryczna prac i ich stopnia zaawansowania
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
Wyzwania rynku pracy	Planowanie kariery zawodowej i metody aktywnego poszukiwania pracy; Sylwetka kandydata na rynku pracy; Analiza rynku pracy; Planowanie kariery; Zasady, techniki metody i narzędzia rekrutacji; Autoprezentacja kandydata; Dokumenty aplikacyjne
Praktyka zawodowa	Podstawy prawne i przedmiot działalności przedsiębiorstwa; Organizacja podmiotu gospodarczego; Przepisy wewnętrzne regulujące funkcjonowanie przedsiębiorstwa; Miejsce działu IT w strukturze organizacyjnej; Konfiguracja zasobów sprzętowych i oprogramowania; Programowanie; Projektowanie systemów informatycznych; Zaawansowane przetwarzanie danych; Projektowanie i utrzymanie baz danych; Wdrożenia IT; Administracja IT; Tworzenie grafiki komputerowej; Bezpieczeństwo IT
SPECJALNOŚĆ: CYBERBEZPIECZEŃSTWO	

Bezpieczeństwo programowania	1. Wprowadzenie do bezpieczeństwa oprogramowania • Pojęcie podatności (vulnerabilities) i zagrożeń • Cykle życia podatności: CVE, CWE, CVSS • Rola programisty w procesie zapewniania bezpieczeństwa 2. Zasady bezpiecznego kodowania • Zasady "secure by design" i "least privilege" • Kodowanie defensywne (defensive programming) • Input validation i sanitizacja danych wejściowych 3. Typowe podatności w kodzie • Przepełnienie bufora (*buffer overflow*) • SQL injection, XSS, CSRF • Insecure deserialization • Race conditions i błędy konkurencji 4. Bezpieczne zarządzanie pamięcią • Alokacja i dealokacja pamięci • Unikanie wycieków pamięci i użycia po zwolnieniu (*use-after-free*) • Praktyki programowania w językach niskiego poziomu (C/C++) 5. Bezpieczeństwo aplikacji webowych i API • OWASP Top 10 (2023) • Zabezpieczanie REST i GraphQL API • Uwierzytelnianie i autoryzacja (OAuth2, JWT) 6. Bezpieczne praktyki w środowisku programistycznym • Kontrola wersji i bezpieczeństwo repozytoriów (np. Git) • Bezpieczne biblioteki zewnętrzne i zależności • Narzędzia SAST i DAST 7. Testowanie bezpieczeństwa oprogramowania • Testy penetracyjne aplikacji (pentesty) • Analiza statyczna i dynamiczna kodu • Fuzzing i testy mutacyjne 8. Reagowanie na incydenty i zarządzanie podatnościami • Praktyki DevSecOps i CI/CD z elementami bezpieczeństwa • Raportowanie i naprawianie błędów bezpieczeństwa • Dokumentowanie i wersjonowanie poprawek 9. Standardy i regulacje • ISO/IEC 27001, OWASP ASVS, NIST 800-53 • Wymagania prawne: RODO, DORA, dyrektywa NIS2 (UE) 10. Przykłady ataków i analiza przypadków (case studies) • Studium przypadków znanych incydentów (np. Log4Shell, Heartbleed) • Analiza błędów w kodzie open-source
Wprowadzenie do chmur obliczeniowych	1. Podstawy chmury obliczeniowej • Definicja i historia cloud computingu • Modele usługowe: IaaS, PaaS, SaaS • Modele wdrożenia: chmura publiczna, prywatna, hybrydowa i community cloud 2. Architektura systemów chmurowych • Kluczowe komponenty i warstwy architektury chmurowej • Wirtualizacja, konteneryzacja, orkiestracja (np. Docker, Kubernetes) • Rozproszone systemy plików i bazy danych 3. Platformy chmurowe – przegląd • Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP) • Porównanie funkcji i zastosowań • Free tier – jak zacząć praktycznie 4. Usługi IaaS i PaaS w praktyce • Tworzenie i zarządzanie maszynami wirtualnymi • Przechowywanie danych: S3, Azure Blob, GCP Storage

	• Sieci wirtualne, load balancery, autoskalowanie 5. Bezpieczeństwo w chmurze • Modele odpowiedzialności (shared responsibility model) • Zabezpieczanie dostępu – IAM, klucze, tokeny • Szyfrowanie danych w spoczynku i w tranzycie 6. Zarządzanie i monitorowanie • Monitorowanie zasobów i usług (CloudWatch, Azure Monitor, Stackdriver) • Automatyzacja: Infrastructure as Code (Terraform, ARM templates) • Koszty i optymalizacja zasobów 7. Chmura a DevOps • Integracja narzędzi CI/CD (GitHub Actions, GitLab, Jenkins) z chmurą • Deployment pipelines i kontenery w środowisku cloud • Przykład prostego projektu z pełną automatyzacją 8. Przykładowe zastosowania chmur • Hostowanie aplikacji webowych • Big Data i analityka w chmurze • Uczenie maszynowe w środowiskach chmurowych 9. Zagadnienia prawne i etyczne • RODO i lokalizacja danych • Vendor lock-in i interoperacyjność • Certyfikacje i zgodność (compliance) 10. Laboratoria i projekty praktyczne • Deploy aplikacji Node.js lub Python w chmurze • Konfiguracja bazy danych w Azure/GCP • Analiza zużycia zasobów i kosztów
Bezpieczeństwo i ochrona danych	1. Wprowadzenie do bezpieczeństwa informacji • Definicje: dane, informacja, bezpieczeństwo informacji • Triada CIA: poufność, integralność, dostępność • Klasyfikacja danych i analiza ryzyka 2. Zasady ochrony danych osobowych • Podstawowe pojęcia RODO/GDPR (administrator, podmiot danych, przetwarzanie) • Zasady przetwarzania danych osobowych • Uprawnienia osób, których dane dotyczą • Obowiązki administratora danych 3. Zabezpieczenia techniczne i organizacyjne • Szyfrowanie danych (AES, RSA, TLS) • Kontrola dostępu (RBAC, ACL) • Bezpieczne przechowywanie i backup danych • Przykłady narzędzi: BitLocker, VeraCrypt, KeePass 4. Zarządzanie tożsamością i dostępem • Uwierzytelnianie i autoryzacja • Wieloskładnikowe uwierzytelnianie (MFA) • Single Sign-On (SSO) i federacja tożsamości 5. Bezpieczeństwo danych w sieciach i systemach • Ochrona danych w transmisji (VPN, HTTPS) • Ataki na dane: sniffing, phishing, ransomware • Firewalle, IDS/IPS, ochrona przed DDoS 6. Bezpieczeństwo danych w środowiskach chmurowych • Modele odpowiedzialności (Shared Responsibility Model) • Ochrona danych w usługach IaaS, PaaS i SaaS • Lokacja danych i zgodność z przepisami 7. Polityki i audyty bezpieczeństwa • Tworzenie polityki bezpieczeństwa danych

	• Audyt wewnętrzny i zewnętrzny • Incydenty i zarządzanie naruszeniami danych 8. Prawo i normy w zakresie ochrony danych • RODO, ePrivacy, ustawa o KSC (PL) • Standardy i dobre praktyki: ISO/IEC 27001, ISO/IEC 27701, NIST SP 800-53 • Odpowiedzialność prawna za naruszenie ochrony danych 9. Etyczne aspekty ochrony danych • Etyka informacyjna • Prywatność jako wartość społeczna • Granice inwigilacji i profilowania 10. Zajęcia praktyczne i projektowe • Analiza ryzyka przetwarzania danych • Projekt polityki bezpieczeństwa dla firmy lub instytucji • Symulacja incydentu i plan reakcji
Elementy kryptografii	1. Wprowadzenie do kryptografii • Historia i rozwój kryptografii (od Cezara do kwantowej) • Kryptografia klasyczna vs nowoczesna • Pojęcia: szyfrowanie, deszyfrowanie, klucz, atak kryptograficzny 2. Systemy szyfrowania symetrycznego • Podstawowe algorytmy: AES, DES, 3DES • Tryby pracy szyfrów blokowych (ECB, CBC, CTR) • Generowanie i dystrybucja kluczy symetrycznych 3. Systemy szyfrowania asymetrycznego • Algorytmy: RSA, ElGamal, ECC (Elliptic Curve Cryptography) • Generowanie par kluczy: publiczny/prywatny • Zastosowanie w podpisie cyfrowym i wymianie kluczy 4. Funkcje skrótu (hashing) • Właściwości funkcji skrótu: jednoznaczność, odporność na kolizje • Algorytmy: SHA-2, SHA-3, MD5 (dla celów dydaktycznych) • Praktyczne zastosowania: integralność danych, hasła, blockchain 5. Podpisy cyfrowe • Zasada działania i algorytmy (RSA, DSA, ECDSA) • Weryfikacja tożsamości i integralności dokumentów • Przykład użycia w certyfikatach SSL/TLS 6. Protokoły kryptograficzne • Protokół Diffie-Hellmana (wymiana klucza) • SSL/TLS, HTTPS – ochrona transmisji w Internecie • Autoryzacja i uwierzytelnianie: OAuth, Kerberos 7. Infrastruktura klucza publicznego (PKI) • Certyfikaty cyfrowe, CA, CRL • Praktyczne aspekty używania certyfikatów (np. w ePUAP, podpis kwalifikowany) • Zarządzanie kluczami 8. Zarządzanie bezpieczeństwem kluczy i danych • Bezpieczne przechowywanie i rotacja kluczy • Smart cards, HSM, TPM, klucze FIDO • Przykłady narzędzi: OpenSSL, GnuPG 9. Kryptografia w praktyce • Zastosowanie w e-commerce, VPN, ePUAP, blockchain • Kryptowaluty i podpisy transakcji • Przykłady ataków kryptograficznych i ich przeciwdziałanie (np. padding oracle, brute-force) 10. Nowoczesne kierunki rozwoju kryptografii • Post-quantum cryptography (algorytmy odporne na komputery kwantowe) • Homomorficzne szyfrowanie i obliczenia na zaszyfrowanych danych

	• Kryptografia kwantowa i protokół BB84
Systemy zarządzania bezpieczeństwem informacji	1. Wprowadzenie do bezpieczeństwa informacji • Kluczowe pojęcia: informacja, bezpieczeństwo, aktywa informacyjne • Triada CIA: poufność, integralność, dostępność • Wartość informacji a ryzyko 2. Normy i standardy bezpieczeństwa informacji • ISO/IEC 27001: struktura i wymagania • ISO/IEC 27002: dobre praktyki zabezpieczeń • Inne normy: ISO 31000 (zarządzanie ryzykiem), ISO 22301 (ciągłość działania) 3. Projektowanie systemu zarządzania bezpieczeństwem informacji (SZBI) • Kontekst organizacji i analiza interesariuszy • Polityka bezpieczeństwa informacji • Określenie zakresu SZBI • Rola kierownictwa i przypisanie odpowiedzialności 4. Identyfikacja i ocena ryzyka • Metodyka analizy ryzyka (np. metoda szacowania ryzyka wg ISO 27005) • Klasyfikacja aktywów i zagrożeń • Ocena prawdopodobieństwa i skutków • Planowanie i wdrażanie zabezpieczeń 5. Środki bezpieczeństwa i mechanizmy kontroli • Techniczne, fizyczne i organizacyjne środki bezpieczeństwa • Zarządzanie tożsamością i kontrola dostępu • Zarządzanie incydentami i kopie zapasowe • Ochrona urządzeń mobilnych i zdalnej pracy 6. Monitorowanie, audyt i doskonalenie SZBI • Wewnętrzne audyty bezpieczeństwa informacji • Przegląd zarządzania, niezgodności, działania korygujące • Cykl PDCA (Plan-Do-Check-Act) w kontekście SZBI • Certyfikacja ISO/IEC 27001 – przebieg i wymagania 7. Ciągłość działania i zarządzanie incydentami • Tworzenie planów ciągłości działania (BCP) i planów odzyskiwania (DRP) • Scenariusze awaryjne i testy odporności • Zasady zarządzania incydentami (np. zgodnie z ISO/IEC 27035) 8. Zgodność z przepisami i wymaganiami prawnymi • RODO / GDPR – ochrona danych osobowych • Ustawa o krajowym systemie cyberbezpieczeństwa (PL) • Wymagania branżowe: DORA (dla instytucji finansowych), NIS2 (UE), HIPAA (USA) 9. Kultura bezpieczeństwa w organizacji • Świadomość pracowników i szkolenia • Rola liderów i komunikacja w zakresie bezpieczeństwa • Przykłady naruszeń wynikających z błędów ludzkich 10. Studia przypadków i dobre praktyki • Przykładowe wdrożenia SZBI w sektorze publicznym i prywatnym • Analiza incydentów i luk w systemach (np. wycieki danych) • Symulacje sytuacji kryzysowych i reakcji organizacji
Podstawy monitoringu systemów i aplikacji	1.Wprowadzenie do monitoringu IT • Cel i znaczenie monitoringu systemów i aplikacji • Typy monitoringu: systemowy, aplikacyjny, sieciowy, bezpieczeństwa • Różnica między monitoringiem a logowaniem 2. Podstawowe metryki i zdarzenia

	• Obciążenie CPU, zużycie pamięci RAM, dostępność dysku • Uptime, czas odpowiedzi, liczba błędów • Logi systemowe i aplikacyjne (syslog, journald, log4j) 3. Architektura rozwiązań monitorujących • Komponenty systemów monitoringu: agenty, kolektory, dashboardy • Push vs pull monitoring • Integracja z bazami danych, API, mikroserwisami 4. Narzędzia monitorujące – przegląd • Prometheus – monitoring metryk • Grafana – wizualizacja danych i alerty • Zabbix, Nagios – kompleksowe monitorowanie infrastruktury • Elastic Stack (ELK) – analiza logów • Datadog, New Relic, Azure Monitor – rozwiązania komercyjne i chmurowe 5. Logowanie i analiza logów • Zbieranie i przechowywanie logów (syslog, Filebeat, Fluentd) • Wzorce logowania w aplikacjach (log levels, struktura) • Detekcja błędów i anomalii w logach 6. Alertowanie i reakcja na incydenty • Tworzenie reguł i progów alertowych • Notyfikacje e-mail, Slack, SMS • Integracja z systemami zarządzania incydentami (np. Jira, PagerDuty) 7. Monitoring aplikacji webowych i API • Monitorowanie endpointów i dostępności usług • Analiza błędów HTTP, kodów statusu, czasu odpowiedzi • Uptime monitoring, synthetic monitoring 8. Monitoring systemów kontenerowych i chmurowych • Monitorowanie Docker i Kubernetes (np. kube-state-metrics, cAdvisor) • Zbieranie metryk z chmury (AWS CloudWatch, Azure Monitor, GCP Stackdriver) • Skalowalność i wysokodostępność rozwiązań monitorujących 9. Wydajność i profilowanie aplikacji • APM (Application Performance Monitoring) – podstawy i zastosowania • Trace'y, latency, bottlenecki • Narzędzia: Jaeger, OpenTelemetry, Zipkin 10. Bezpieczeństwo monitoringu • Monitoring a SIEM (Security Information and Event Management) • Wykrywanie ataków i nieprawidłowości • Zarządzanie dostępem do danych monitorujących
Wykrywanie i analiza zagrożeń w sieci	1. Wprowadzenie do zagrożeń w sieciach komputerowych • Klasyfikacja zagrożeń: pasywne i aktywne • Ataki na warstwy sieciowe (model OSI, TCP/IP) • Zagrożenia lokalne, zdalne, wewnętrzne i zewnętrzne 2. Podstawy przechwytywania i analizy ruchu sieciowego • Narzędzia: Wireshark, tcpdump • Analiza pakietów: struktura ramek Ethernet, IP, TCP/UDP • Wykrywanie anomalii w ruchu 3. Systemy wykrywania włamań (IDS/IPS) • Architektura i funkcje IDS/IPS • Porównanie systemów: Snort, Suricata, Zeek (Bro) • Różnice między detekcją opartą na sygnaturach i analizą behawioralną 4. Zbieranie i korelacja logów • Centralizacja logów: syslog, Rsyslog, Logstash • Korelacja zdarzeń i analiza wzorców • Wstęp do SIEM (Security Information and Event Management) 5. Analiza zagrożeń i incydentów

	• Analiza ataków typu DoS/DDoS, ARP spoofing, port scanning • Identyfikacja nieautoryzowanego dostępu • Analiza prób phishingu i malware w ruchu sieciowym 6. Threat intelligence i źródła IOC (Indicators of Compromise) • Bazy danych zagrożeń: VirusTotal, AbuseIPDB, AlienVault OTX • Wykorzystanie IOC do wykrywania kampanii ataków • STIX/TAXII – formaty wymiany informacji o zagrożeniach 7. Automatyzacja wykrywania zagrożeń • Skrypty i automatyczne reguły w Snort/Suricata • Integracja z narzędziami open source (np. TheHive, MISP) • Podstawy automatyzacji reagowania (SOAR) 8. Bezpieczeństwo protokołów i usług sieciowych • Wykrywanie podatności w protokołach (DNS, HTTP, SMB, FTP) • Skany podatności – np. Nmap, Nessus (podstawy) • Analiza sesji TLS/SSL – MITM, downgrade attacks 9. Symulacje i testy penetracyjne w środowisku sieciowym • Laboratoria typu CTF (Capture The Flag) • Ćwiczenia w środowiskach testowych (np. Kali Linux + Metasploitable) • Tworzenie raportów z incydentów 10. Reagowanie na zagrożenia • Proces obsługi incydentu bezpieczeństwa • Zbieranie dowodów (digital forensics – podstawy) • Współpraca z zespołami CSIRT/SOC
Ochrona danych w chmurze	1. Wprowadzenie do ochrony danych w chmurze • Definicja danych i ich wartości w środowiskach cloud • Modele usług chmurowych (IaaS, PaaS, SaaS) a odpowiedzialność za bezpieczeństwo • Modele wdrożeniowe: chmura publiczna, prywatna, hybrydowa 2. Zagrożenia dla danych w chmurze • Klasyfikacja zagrożeń (utrata danych, wycieki, nieautoryzowany dostęp) • Ataki charakterystyczne dla środowisk chmurowych (np. man-in-the-cloud, misconfigured buckets, side-channel attacks) • Ryzyka związane z przetwarzaniem i przechowywaniem danych poza granicami kraju 3. Szyfrowanie danych w chmurze • Szyfrowanie danych w spoczynku i w tranzycie (encryption at rest / in transit) • Algorytmy szyfrowania: AES, RSA, TLS, ECC • Zarządzanie kluczami szyfrującymi – KMS (Key Management Service) w AWS, Azure i GCP 4. Mechanizmy kontroli dostępu • Zarządzanie tożsamością i dostępem (IAM – Identity and Access Management) • Polityki uprawnień – Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC) • MFA (uwierzytelnianie wieloskładnikowe) i federacja tożsamości (SSO, SAML, OAuth) 5. Ochrona danych osobowych w środowiskach chmurowych • RODO/GDPR i dane osobowe w chmurze • Zasady minimalizacji, pseudonimizacji i anonimizacji • Umowy powierzenia przetwarzania danych (DPA) z dostawcami usług 6. Audyt i zgodność w chmurze • Standardy bezpieczeństwa: ISO/IEC 27017, ISO/IEC 27018, SOC 2, CSA STAR • Monitorowanie zgodności z przepisami i normami • Mechanizmy śledzenia zdarzeń i logowania aktywności użytkowników 7. Kopie zapasowe i odzyskiwanie danych

	• Backup danych w chmurze – typy, częstotliwość, przechowywanie • Planowanie odzyskiwania danych po awarii (Disaster Recovery, RTO/RPO) • Testowanie procedur backupu i odzyskiwania 8. Bezpieczne zarządzanie cyklem życia danych • Tworzenie, przechowywanie, archiwizacja i usuwanie danych • Mechanizmy automatycznego czyszczenia danych (data lifecycle policies) • Weryfikacja skuteczności usuwania (data remanence problem) 9. Odpowiedzialność dostawców i klientów • Shared Responsibility Model – kto za co odpowiada? • Ocena wiarygodności dostawców chmury • Umowy SLA i ich rola w kontekście bezpieczeństwa danych 10. Zajęcia praktyczne i case studies • Konfiguracja polityk bezpieczeństwa w AWS/Azure/GCP • Symulacja wycieku danych z nieprawidłowo skonfigurowanego zasobu (np. S3 bucket) • Przegląd zdarzeń w logach audytowych (CloudTrail, Azure Monitor)
Zarządzanie projektami bezpieczeństwa IT	1. Wprowadzenie do zarządzania projektami IT • Definicja projektu IT a projekt bezpieczeństwa informacji • Specyfika projektów w obszarze IT Security • Rola kierownika projektu i zespołu bezpieczeństwa 2. Planowanie projektów bezpieczeństwa IT • Określanie celów projektu: np. wdrożenie systemu DLP, SIEM, audytu bezpieczeństwa • Zakres, czas, koszty, jakość – model potrójnego ograniczenia (triple constraint) • Tworzenie harmonogramu i macierzy zadań (WBS – Work Breakdown Structure) 3. Analiza ryzyka i zarządzanie ryzykiem • Identyfikacja i klasyfikacja ryzyk w projektach bezpieczeństwa • Ocena ryzyk: prawdopodobieństwo, wpływ, priorytetyzacja • Reakcje na ryzyko: unikanie, minimalizacja, przeniesienie, akceptacja 4. Standardy i metodyki zarządzania projektami • PMBOK, PRINCE2, Agile w projektach bezpieczeństwa • Metody hybrydowe – kiedy klasyczne, kiedy zwinne podejście • Dokumentacja projektowa: plan projektu, karta projektu, raport ryzyk 5. Zarządzanie interesariuszami i komunikacją • Identyfikacja kluczowych interesariuszy (zarząd, IT, użytkownicy końcowi, audytorzy) • Komunikacja projektowa – plany komunikacji i raportowania • Zarządzanie oporem przed zmianą (Change Management) 6. Budżetowanie projektów bezpieczeństwa • Szacowanie kosztów wdrożeń (licencje, usługi, szkolenia, sprzęt) • Narzędzia do kontroli budżetu i analiza kosztów/koszyków bezpieczeństwa (np. TCO) • Przykład uzasadnienia budżetowego dla projektu IT Security (np. SOC-as-a-Service) 7. Wdrażanie projektów i kontrola postępów • Realizacja etapów projektu: od analizy do testów i wdrożenia • Monitorowanie wykonania zadań (Gantt, KPI, Milestones) • Narzędzia wspierające: JIRA, MS Project, Asana, Trello 8. Audyt i testowanie bezpieczeństwa • Fazy testów: testy penetracyjne, testy akceptacyjne, testy regresji • Audyt wdrożenia zgodnie z normami ISO/IEC 27001/27002 • Dokumentacja końcowa projektu: raporty, zalecenia, lessons learned 9. Zgodność, regulacje i etyka • RODO, NIS2, ustawa o Krajowym Systemie Cyberbezpieczeństwa

	• Wymagania branżowe: DORA, ISO 22301, ISO 27005 • Odpowiedzialność kierownika projektu za zgodność z przepisami 10. Case studies i symulacje projektowe • Przykłady udanych i nieudanych projektów bezpieczeństwa IT • Studium wdrożenia SIEM, systemu zarządzania tożsamością (IAM), audytu RODO • Symulacja projektu – tworzenie i prezentacja własnego mini-projektu
SPECJALNOŚĆ: PROGRAMOWANIE	
Testowanie i jakość oprogramowania	1. Wprowadzenie do testowania oprogramowania • Cel i znaczenie testowania w cyklu życia oprogramowania • Podstawowe pojęcia: defekt, błąd, awaria, test case, test suite • Różnice między weryfikacją a walidacją 2. Rodzaje i poziomy testów • Testy jednostkowe (unit tests) • Testy integracyjne, systemowe i akceptacyjne • Testy manualne vs automatyczne • Testy funkcjonalne i niefunkcjonalne (np. wydajnościowe, bezpieczeństwa) 3. Techniki projektowania przypadków testowych • Techniki czarnoskrzynkowe (black-box): tablice decyzyjne, klasy równoważności, analiza wartości brzegowych • Techniki białoskrzynkowe (white-box): pokrycie instrukcji, gałęzi, ścieżek • Testy eksploracyjne i heurystyki testowania 4. Automatyzacja testów • Zasady automatyzacji: co warto automatyzować? • Narzędzia do testów automatycznych: Selenium, JUnit, TestNG, Postman (API) • CI/CD i testy automatyczne w potoku DevOps (np. GitHub Actions, Jenkins) 5. Testowanie aplikacji webowych i mobilnych • Testy interfejsu użytkownika (UI testing) • Testy REST API – narzędzia i metody • Testowanie responsywności i kompatybilności przeglądarek 6. Zarządzanie jakością oprogramowania • Modele jakości: ISO/IEC 25010, CMMI, TMMi • Zapewnienie jakości (Quality Assurance – QA) vs Kontrola jakości (Quality Control – QC) • Procesy zarządzania jakością w zwinnych i klasycznych metodykach 7. Narzędzia wspierające testowanie • Systemy zarządzania testami: TestLink, Zephyr, Xray (Jira) • Rejestracja i śledzenie błędów: Jira, Bugzilla, Mantis • CI/CD + testy: GitLab CI, Jenkins, Azure DevOps 8. Metodyki i strategie testowania • Testowanie w modelu kaskadowym vs w metodykach zwinnych (Scrum, Kanban) • Test-Driven Development (TDD) i Behavior-Driven Development (BDD) • Risk-Based Testing, exploratory testing 9. Zarządzanie defektami • Cykl życia defektu: zgłoszenie → weryfikacja → naprawa → ponowne testowanie → zamknięcie • Priorytety i poziomy ważności błędów • Raportowanie i analiza przyczyn błędów (Root Cause Analysis) 10. Jakość i bezpieczeństwo a satysfakcja użytkownika • UX a jakość • Wpływ błędów na reputację i zaufanie użytkowników • Audyty jakościowe i przeglądy kodu (code review)

Projektowanie i implementacja aplikacji rozproszonych	1. Wprowadzenie do systemów i aplikacji rozproszonych • Definicja i cechy systemów rozproszonych (brak wspólnej pamięci, niezależność węzłów) • Zalety i wyzwania architektury rozproszonej (skalowalność, odporność, złożoność) • Przykłady aplikacji rozproszonych: systemy bankowe, serwisy strumieniowe, chmury obliczeniowe 2. Modele komunikacji i architektury systemów rozproszonych • Architektura klient-serwer vs peer-to-peer • Komunikacja synchroniczna i asynchroniczna • Protokół RPC (Remote Procedure Call), REST, gRPC 3. Interfejsy API i komunikacja międzyprocesowa • Projektowanie RESTful API – dobre praktyki • JSON, XML, protobuf – formaty danych • Autoryzacja i uwierzytelnianie w rozproszonych środowiskach (OAuth2, JWT) 4. Zarządzanie stanem i spójnością danych • Problemy spójności danych (CAP theorem) • Techniki replikacji i partycjonowania • Bazy danych NoSQL w systemach rozproszonych (np. MongoDB, Cassandra) 5. Zarządzanie usługami i architektura mikroserwisowa • Mikroserwisy vs monolity – porównanie • Konteneryzacja aplikacji (Docker) i orkiestracja (Kubernetes) • Discovery services, load balancing, service mesh (np. Istio) 6. Zarządzanie błędami i odporność na awarie • Idempotencja, time-outy, retry, circuit breakers • Wzorce odpornościowe: fallback, bulkhead, timeout • Narzędzia: Hystrix (lub jego zamienniki), Resilience4j 7. Bezpieczeństwo w aplikacjach rozproszonych • Bezpieczna komunikacja między usługami (TLS, VPN, tokeny) • Zarządzanie tożsamością i uprawnieniami • Audyt i monitorowanie działań w systemie 8. Monitorowanie i skalowanie aplikacji rozproszonych • Logging, tracing (np. OpenTelemetry, Jaeger, Zipkin) • Prometheus i Grafana – monitorowanie metryk aplikacji • Dynamiczne skalowanie i autoskalery 9. Testowanie aplikacji rozproszonych • Testy jednostkowe i integracyjne w środowisku mikroserwisowym • Testy kontraktowe (contract testing) • Symulacja opóźnień i awarii – chaos engineering (np. Chaos Monkey) 10. Praktyczne wdrożenie projektu rozproszonego • Praca zespołowa nad implementacją wieloelementowej aplikacji (np. sklep internetowy, system rezerwacyjny) • Dokumentacja architektury, API i konfiguracji systemu • Wdrożenie w środowisku chmurowym (np. AWS/Azure/GCP lub lokalnie z Docker Compose)
Programowanie w zespole – studium przypadku	1. Wprowadzenie do pracy zespołowej w projektach IT • Rola zespołów programistycznych w procesie wytwarzania oprogramowania • Modele współpracy: zespoły zwinne, struktury tradycyjne, interdyscyplinarne • Przykładowe role w zespole: developer, lider, tester, analityk, DevOps 2. Zarządzanie projektem programistycznym • Wybór i omówienie metodyki prowadzenia projektu: Scrum, Kanban, Agile hybrid • Planowanie sprintów i retrospektywy • Narzędzia do zarządzania pracą zespołu: Jira, Trello, GitHub Projects

	3. Tworzenie zespołowego środowiska pracy • Systemy kontroli wersji – Git: workflow zespołowy (feature branches, pull requesty, code review) • Zasady pracy z repozytorium: commitowanie, branchowanie, rozwiązywanie konfliktów • Narzędzia komunikacyjne: Slack, Discord, MS Teams 4. Analiza wymagań i dokumentacja projektowa • Zbieranie i analiza wymagań od "klienta" (studium przypadku) • Tworzenie dokumentacji funkcjonalnej i niefunkcjonalnej • Diagramy UML (np. przypadków użycia, klas, sekwencji) 5. Projektowanie architektury aplikacji • Wybór stosu technologicznego (frontend + backend + baza danych) • Architektura warstwowa / mikroserwisowa / MVC • Projektowanie API (np. RESTful API) 6. Współpraca przy kodowaniu i dobre praktyki • Podział zadań i integracja kodu • Konwencje kodowania, przeglądy kodu (code review) • Refaktoryzacja i ciągła integracja (CI) 7. Testowanie i zapewnianie jakości • Pisanie testów jednostkowych i integracyjnych • Testowanie manualne i automatyczne • Integracja testów z CI/CD (np. GitHub Actions, GitLab CI) 8. Wdrożenie aplikacji • Przygotowanie aplikacji do wdrożenia • Wdrożenie lokalne / chmurowe / na serwerze testowym • Prezentacja działania aplikacji 9. Zarządzanie problemami i ryzykiem • Typowe wyzwania pracy zespołowej: konflikty, niekomunikatywność, opóźnienia • Techniki rozwiązywania problemów w zespole • Dokumentowanie problemów i prowadzenie retrospektyw 10. Podsumowanie projektu – prezentacja i ewaluacja • Prezentacja końcowa projektu (demo) • Ewaluacja pracy zespołu i członków • Wnioski, raport końcowy, refleksja nad procesem wytwarzania
Wzorce projektowe	1. Wprowadzenie do wzorców projektowych • Co to są wzorce projektowe i dlaczego są potrzebne? • Historia i klasyfikacja wzorców (Gang of Four) • Antywzorce i złe praktyki (anti-patterns) • Zasady SOLID jako fundament dobrego projektowania 2. Wzorce kreacyjne (Creational Patterns) • Singleton – zapewnienie pojedynczej instancji • Factory Method – dynamiczne tworzenie obiektów • Abstract Factory – tworzenie całych rodzin obiektów • Builder – konstruowanie złożonych obiektów krok po kroku • Prototype – klonowanie istniejących obiektów 3. Wzorce strukturalne (Structural Patterns) • Adapter – łączenie niekompatybilnych interfejsów • Decorator – dynamiczne dodawanie funkcjonalności • Composite – struktury drzewiaste i rekursywne • Proxy – podstawianie obiektu pośredniego • Bridge – oddzielenie abstrakcji od implementacji • Facade – uproszczony interfejs do złożonego systemu 4. Wzorce czynnościowe (Behavioral Patterns)

	• Observer – powiadamianie wielu obiektów o zmianach stanu • Strategy – dynamiczna zamiana algorytmu • Command – reprezentacja poleceń jako obiektów • State – zmiana zachowania obiektu w zależności od stanu • Template Method – szkielet algorytmu z możliwością modyfikacji kroków • Mediator, Chain of Responsibility, Iterator, Visitor 5. Zasady implementacji wzorców • Przykłady kodu w językach: Java, Python, C#, TypeScript • Wady i zalety konkretnych wzorców • Kiedy i gdzie stosować – dobre praktyki 6. Wzorce projektowe w praktyce • Stosowanie wzorców w projektach zespołowych • Wzorce w aplikacjach webowych i desktopowych • Analiza wzorców w bibliotekach open source (np. Spring, Django) 7. Testowanie i refaktoryzacja z użyciem wzorców • Ułatwienie testowania przez zastosowanie wzorców • Wzorce w kontekście TDD i BDD • Refaktoryzacja z użyciem wzorców: rozbijanie dużych klas, unifikacja interfejsów 8. Nowoczesne wzorce i podejścia architektoniczne • Wzorce w architekturze mikroserwisowej (np. Circuit Breaker, Event Sourcing) • Domain-Driven Design (DDD) i wzorce w domenie • Wzorce architektoniczne: MVC, MVVM, Hexagonal Architecture 9. Projekt zaliczeniowy • Analiza problemu projektowego i wybór odpowiednich wzorców • Dokumentacja decyzji projektowych • Implementacja rozwiązania z użyciem co najmniej 3 typów wzorców
Zaawansowana inżynieria kodu	1. Wprowadzenie do zaawansowanej inżynierii oprogramowania • Czym różni się „pisanie kodu” od jego „inżynierii”? • Kod produkcyjny vs kod akademicki • Cechy dobrego kodu: czytelność, testowalność, rozszerzalność, wydajność 2. Architektura i struktura dużych aplikacji • Modułowość, separacja odpowiedzialności (Separation of Concerns) • Architektura warstwowa, heksagonalna, DDD • Clean Architecture – zasady i przykłady implementacji 3. Zaawansowane techniki refaktoryzacji • Wykrywanie i eliminacja „code smells” • Refaktoryzacja bez zmiany zachowania – testy jako zabezpieczenie • Przykłady refaktoryzacji: zagnieżdżone warunki, długa klasa, duplikacja kodu 4. Wzorce projektowe i idiomy językowe • Integracja wzorców projektowych w dużych projektach • Idiomy specyficzne dla języków (Java, Python, C#) • Fluent API, Lazy Initialization, Dependency Injection 5. Zarządzanie złożonością • Redukcja złożoności cyklomatycznej • De-kompozycja problemów i funkcji • Narzędzia do analizy złożoności kodu 6. Praca z technicznym długiem • Identyfikacja długu technicznego • Dokumentowanie, priorytetyzacja i planowanie spłaty długu • Koszt zaniechania utrzymania jakości kodu 7. Zaawansowane testowanie i jakość kodu • Testy jednostkowe, integracyjne, mutacyjne • Pokrycie kodu testami – analiza, narzędzia (np. JaCoCo, Coverage.py) • Testowanie kodu legacy

	8. Narzędzia wspierające inżynierię kodu • Lintery i statyczna analiza kodu (SonarQube, ESLint, Pylint) • Code Review – procesy, dobre praktyki • CI/CD z automatyczną kontrolą jakości kodu (np. pre-commit hooks, pipelines) 9. Wydajność i optymalizacja kodu • Profilowanie aplikacji (memory, CPU, I/O) • Optymalizacja algorytmiczna i mikrooptymalizacje • Kiedy optymalizować, a kiedy refaktoryzować 10. Kultura inżynierii kodu • Dokumentowanie kodu – styl, standardy, narzędzia (np. Sphinx, Javadoc) • Współpraca w dużym zespole koderskim – merge requesty, review, pair programming • Mentoring i „code ownership”
Integracja oprogramowania z platformą Azure	1. Wprowadzenie do Microsoft Azure • Architektura i komponenty chmury Microsoft Azure • Modele usług: IaaS, PaaS, SaaS • Tworzenie i zarządzanie kontem, subskrypcje, zasoby (Azure Resource Manager) 2. Tworzenie i wdrażanie aplikacji w Azure • Hostowanie aplikacji webowych (Azure App Service) • Wdrażanie aplikacji z GitHub / Azure DevOps / lokalnych repozytoriów • Wersjonowanie i sloty wdrożeniowe (deployment slots) 3. Integracja z bazami danych i usługami przechowywania • Azure SQL Database i Cosmos DB – połączenia i zarządzanie • Azure Blob Storage, File Storage – operacje CRUD i integracja z aplikacjami • Bezpieczne przechowywanie danych konfiguracyjnych (Azure Key Vault) 4. Uwierzytelnianie i autoryzacja • Azure Active Directory (AAD) – integracja z aplikacjami • Logowanie zewnętrzne (Google, Facebook, Microsoft) • Role-Based Access Control (RBAC) i zarządzanie tożsamością 5. Usługi integracyjne Azure • Azure Logic Apps – budowanie procesów bez kodowania • Azure Functions – integracja w architekturze serverless • Azure API Management – tworzenie i zabezpieczanie interfejsów API 6. Integracja z usługami DevOps • Azure DevOps – CI/CD pipelines, Repos, Boards • Automatyzacja testów i wdrożeń • Monitorowanie jako element pipeline’u 7. Monitorowanie i diagnostyka aplikacji • Azure Monitor i Application Insights – logi, metryki, alerty • Analiza działania aplikacji i identyfikacja błędów • Integracja z Power BI dla wizualizacji danych 8. Bezpieczeństwo integracji i danych • Zarządzanie kluczami, certyfikatami i tożsamościami • Ochrona transmisji danych – TLS, szyfrowanie • Audyty i zgodność z normami (ISO, RODO, NIS2) 9. Zarządzanie środowiskiem i kosztami • Organizacja zasobów: grupy zasobów, tagowanie, subskrypcje • Szacowanie kosztów: Azure Pricing Calculator • Monitorowanie i optymalizacja zużycia zasobów 10. Projekt praktyczny • Projektowanie, implementacja i wdrożenie aplikacji korzystającej z Azure App Service, bazy danych i co najmniej jednej funkcji integracyjnej (np. Azure Functions + Storage + Monitoring) • Dokumentacja projektu i prezentacja integracji

Programowanie aplikacji internetowych MVC API	1. Wprowadzenie do architektury MVC • Model-View-Controller – podział odpowiedzialności • Rola warstwy modelu, widoku i kontrolera • Przykłady implementacji MVC w różnych technologiach (ASP.NET Core, Spring MVC, Django, Laravel) 2. Tworzenie aplikacji webowej w wybranym frameworku • Struktura projektu MVC • Konfiguracja środowiska (np. .NET, Node.js, Java Spring, Django) • Routing i kontrolery – obsługa żądań HTTP 3. Modelowanie danych i integracja z bazą danych • Tworzenie modeli danych (ORM: Entity Framework, Hibernate, Django ORM) • Migrations i synchronizacja schematu bazy • Operacje CRUD (Create, Read, Update, Delete) 4. Projektowanie i tworzenie API (RESTful) • Podstawy REST: zasoby, metody HTTP, statusy odpowiedzi • Projektowanie endpointów API zgodnie z dobrymi praktykami • Serializacja danych: JSON, XML • API versioning i dokumentacja (np. Swagger / OpenAPI) 5. Autoryzacja i uwierzytelnianie użytkowników • System logowania (np. JWT, OAuth2) • Role i uprawnienia użytkowników • Middleware i zabezpieczanie endpointów 6. Frontend i konsumpcja API • Integracja frontendów SPA (np. React, Angular, Vue) z backendem API • AJAX / Fetch / Axios – komunikacja z API • Walidacja danych po stronie klienta i serwera 7. Walidacja i obsługa błędów • Walidacja danych wejściowych (np. FluentValidation, class-validator) • Obsługa wyjątków, kodów błędów i logowanie zdarzeń • Middleware do obsługi błędów 8. Testowanie aplikacji webowych i API • Testy jednostkowe kontrolerów i logiki biznesowej • Testy integracyjne i testy API (np. Postman, REST Assured, Supertest) • Pokrycie testami i automatyzacja (CI/CD) 9. Deploy aplikacji webowej • Wdrożenie aplikacji na serwerze (np. IIS, Nginx, Apache) • Hosting w chmurze (np. Azure App Service, Heroku, AWS) • Monitorowanie działania aplikacji i logi błędów 10. Projekt zespołowy / indywidualny • Opracowanie pełnej aplikacji webowej z API i bazą danych • Dokumentacja API i struktury projektu • Prezentacja działania aplikacji
Projekt systemu informatycznego	1. Wprowadzenie do projektowania systemów informatycznych • Czym jest system informatyczny: komponenty, cykl życia, interesariusze • Proces tworzenia systemów – od analizy wymagań po wdrożenie • Modele SDLC (System Development Life Cycle): kaskadowy, iteracyjny, zwinny 2. Analiza potrzeb i wymagań użytkowników • Identyfikacja interesariuszy • Wymagania funkcjonalne i niefunkcjonalne • Narzędzia: analiza SWOT, burze mózgów, wywiady, diagramy kontekstowe

	3. Specyfikacja i dokumentacja wymagań • Tworzenie specyfikacji SRS (Software Requirements Specification) • Modelowanie wymagań: przypadki użycia (use cases), user stories • Makiety, scenariusze użytkownika, prototypowanie 4. Projektowanie architektury systemu • Wybór architektury: monolit, warstwowa, mikroserwisowa, klient-serwer • Dobór technologii: backend, frontend, baza danych • Tworzenie diagramów UML: klas, sekwencji, komponentów, aktywności 5. Modelowanie danych i projekt bazy danych • Diagramy ERD (Entity-Relationship Diagram) • Normalizacja danych i projektowanie relacji • Projekt logiczny i fizyczny bazy danych 6. Interfejs użytkownika i UX • Projektowanie GUI: zasady projektowania interfejsów • Makietowanie (np. Figma, Balsamiq) • Testy użyteczności i walidacja koncepcji 7. Planowanie projektu i podział zadań • Harmonogramy (Gantt), zależności zadań, kamienie milowe • Tworzenie zespołu projektowego: role i odpowiedzialności • Narzędzia do zarządzania projektami (np. Jira, Trello, GitHub Projects) 8. Wdrożenie i integracja • Etapy wdrażania systemu (dev → staging → produkcja) • Testy integracyjne, walidacja funkcji, testy akceptacyjne • Dokumentacja techniczna i użytkownika 9. Bezpieczeństwo i niezawodność systemu • Mechanizmy uwierzytelniania, kontroli dostępu, logowania zdarzeń • Backup, plan przywracania, odporność na awarie • Zgodność z RODO i normami (np. ISO 27001) 10. Prezentacja i ewaluacja projektu • Demo systemu – funkcje, architektura, wartość biznesowa • Raport końcowy – analiza zgodności z wymaganiami, analiza ryzyk • Refleksja zespołu: co się udało, co można poprawić
Algorytmizacja	1. Wprowadzenie do algorytmiki • Pojęcie algorytmu, właściwości algorytmów (skończoność, jednoznaczność, poprawność) • Podstawowe struktury danych: tablice, listy, stosy, kolejki • Przedstawianie algorytmów: pseudokod, schematy blokowe, języki programowania 2. Złożoność obliczeniowa • Analiza czasowa i pamięciowa algorytmów (notacja O(n), Ω(n), Θ(n)) • Porównanie różnych klas złożoności • Analiza przypadków: najlepszy, średni, najgorszy 3. Techniki projektowania algorytmów • Dziel i zwyciężaj (*divide and conquer*) • Algorytmy zachłanne (*greedy algorithms*) • Programowanie dynamiczne (*dynamic programming*) • Przeszukiwanie z powrotami (*backtracking*) 4. Podstawowe algorytmy sortowania i wyszukiwania • Sortowanie: bąbelkowe, przez wstawianie, przez wybór, szybkie (*quicksort*), scalanie (*mergesort*) • Wyszukiwanie liniowe i binarne • Algorytmy sortowania stabilnego vs niestabilnego 5. Algorytmy operujące na ciągach i tekstach • Przeszukiwanie wzorca (np. algorytm Knutha-Morrisa-Pratta, Boyera-Moore'a)

	• Algorytmy porównywania napisów • Zastosowanie tablic prefiksowych i sufixowych 6. Algorytmy grafowe – podstawy • Reprezentacja grafów (listy sąsiedztwa, macierze sąsiedztwa) • Przeszukiwanie wszerz (BFS) i w głąb (DFS) • Najkrótsza ścieżka (Dijkstra, Bellman-Ford) • Minimalne drzewa rozpinające (Kruskal, Prim) 7. Algorytmy numeryczne i rekurencja • Algorytmy operujące na liczbach: największy wspólny dzielnik (NWD), liczby pierwsze, liczby Fibonacciego • Rekurencja: definicje rekurencyjne i ich iteracyjne odpowiedniki • Optymalizacja rekurencji (memoizacja) 8. Algorytmy na strukturach danych • Operacje na listach, stosach i kolejkach • Wyszukiwanie w drzewach binarnych • Wprowadzenie do algorytmów na kopcach i haszowaniu 9. Problemy NP-trudne i złożone algorytmicznie • Problemy NP i klasy NP-zupełne (np. problem komiwojażera) • Aproksymacje i heurystyki • Algorytmy probabilistyczne i losowe 10. Zastosowanie algorytmów w praktyce • Przykłady algorytmów z życia codziennego: logistyka, wyszukiwanie, planowanie • Algorytmy w grach, sztucznej inteligencji i systemach rekomendacyjnych • Projekt końcowy: rozwiązanie wybranego problemu algorytmicznego z analizą
SPECJALNOŚĆ: SZTUCZNA INTELIGENCJA	
Podstawy uczenia maszynowego	1. Wprowadzenie do uczenia maszynowego • Czym jest uczenie maszynowe (ML)? – definicje i klasyfikacja • Główne paradygmaty: uczenie nadzorowane, nienadzorowane i wzmacniane • Zastosowania ML: rozpoznawanie obrazów, analiza tekstu, predykcja 2. Podstawowe pojęcia i terminologia • Dane treningowe, cechy (features), etykiety (labels) • Modele, funkcje kosztu, predykcja, generalizacja • Overfitting, underfitting i walidacja krzyżowa 3. Uczenie nadzorowane: klasyfikacja i regresja • Regresja liniowa i wieloraka • Klasyfikacja binarna i wieloklasowa • Algorytmy: k-Nearest Neighbors, Decision Trees, Naive Bayes, Support Vector Machines (SVM) 4. Uczenie nienadzorowane • Grupowanie (clustering): k-means, DBSCAN • Redukcja wymiarowości: PCA (Principal Component Analysis) • Wykrywanie anomalii 5. Uczenie głębokie – wprowadzenie • Wstęp do sieci neuronowych: perceptron, warstwy, aktywacje • Architektury: MLP (Multilayer Perceptron) • Ramy działania: TensorFlow/Keras lub PyTorch 6. Inżynieria cech i przygotowanie danych • Skalowanie, normalizacja, kodowanie zmiennych (One-hot, LabelEncoder) • Podział zbioru danych (train/test/validation) • Walka z brakującymi i odstającymi danymi 7. Ocena jakości modeli

	• Metryki: accuracy, precision, recall, F1-score, ROC-AUC • Krzyżowa walidacja i Grid Search • Wykresy: confusion matrix, ROC curve, learning curve 8. Praca z bibliotekami ML • Scikit-learn: klasyfikacja, regresja, grupowanie • Pandas, NumPy – przygotowanie danych • Matplotlib/Seaborn – wizualizacja wyników 9. Etapy budowy modelu ML • Przebieg projektu ML: od problemu do wdrożenia • Eksperymentowanie z parametrami • Przykładowy pipeline: zbiór danych → preprocessing → trening → ewaluacja → predykcja 10. Projekt praktyczny • Realizacja prostego projektu ML w grupie lub indywidualnie • Dobór danych, analiza cech, wybór modelu, ocena skuteczności • Prezentacja wyników i raport końcowy
Systemy wspomagania decyzji	1. Wprowadzenie do systemów wspomagania decyzji • Definicja DSS i ich miejsce w systemach informacyjnych • Klasyfikacja DSS: model-driven, data-driven, knowledge-driven • Historia i rozwój DSS 2. Struktura i architektura DSS • Kluczowe komponenty: baza danych, baza modeli, interfejs użytkownika • Integracja z systemami ERP, CRM i BI • Przykłady architektur (centralna, rozproszona, chmurowa) 3. Modele decyzyjne • Typy decyzji: strukturalne, półstrukturalne, niestrukturalne • Drzewa decyzyjne, macierze decyzyjne, analiza wielokryterialna (MCDA) • Scenariusze i symulacje decyzyjne 4. Metody wspomagania decyzji • Analiza wrażliwości • Metody optymalizacji (np. programowanie liniowe) • Metody wielokryterialne (AHP, TOPSIS, ELECTRE) 5. Bazy danych i hurtownie danych w DSS • Rola danych w procesie podejmowania decyzji • Hurtownie danych i OLAP • Zapytania decyzyjne i przetwarzanie analityczne 6. Systemy Business Intelligence i ich funkcje • BI jako rozszerzenie DSS • Dashboardy, raporty, KPI • Integracja narzędzi BI (np. Power BI, Tableau, Qlik) 7. Sztuczna inteligencja i uczenie maszynowe w DSS • Systemy ekspertowe i regułowe • Proste algorytmy ML jako wsparcie decyzji (np. klasyfikacja klientów) • Wprowadzenie do inteligentnych systemów doradczych 8. Projektowanie i wdrażanie DSS • Proces budowy systemu DSS: od analizy potrzeb do prototypu • Rola analityka biznesowego i zespołu IT • Etapy projektowania interfejsu użytkownika 9. Zastosowania DSS w praktyce • Finanse, logistyka, marketing, medycyna, produkcja • Case studies: np. systemy scoringowe, logistyka dostaw, analiza ryzyka • DSS w administracji publicznej i edukacji 10. Etyka i bezpieczeństwo DSS • Automatyzacja decyzji a odpowiedzialność

	• Zaufanie do wyników systemu • Ochrona danych osobowych i analiza zgodności (RODO, etyka algorytmów)
Projekt zespołowy – rozwiązania AI	1. Wprowadzenie do pracy zespołowej w projektach AI • Metodyki prowadzenia projektów AI (Agile, Kanban, CRISP-DM) • Tworzenie zespołu: role (data scientist, developer, analityk, tester) • Komunikacja i narzędzia zarządzania projektem: Git, Trello, Jira, Slack 2. Definiowanie problemu i celu projektu AI • Określenie celu biznesowego lub społecznego projektu • Dobór odpowiedniego podejścia AI (ML, NLP, CV, predykcja, klasyfikacja) • Przegląd literatury i inspiracji (np. Kaggle, Google AI, open source) 3. Zbieranie i przygotowanie danych • Wybór zbiorów danych (gotowe lub stworzone przez zespół) • Czyszczenie, przekształcanie, eksploracyjna analiza danych (EDA) • Wstępna wizualizacja danych i wybór cech 4. Budowa i trening modeli AI • Dobór modeli: regresja, drzewa decyzyjne, SVM, sieci neuronowe • Trenowanie modeli i walidacja (train/test/val split, cross-validation) • Weryfikacja poprawności działania (overfitting, underfitting) 5. Integracja modeli z aplikacją / systemem • Implementacja modelu w aplikacji webowej, mobilnej lub desktopowej • Użycie bibliotek: scikit-learn, TensorFlow, Keras, PyTorch, spaCy • Interfejs API dla modelu (np. FastAPI, Flask) 6. Wizualizacja i interpretacja wyników • Interaktywne dashboardy (Power BI, Streamlit, Dash) • Tłumaczalność modelu (Explainable AI): SHAP, LIME • Ocena działania – metryki, wykresy, raporty 7. Testowanie i optymalizacja rozwiązania • Testy jednostkowe i integracyjne aplikacji z modelem • Tuning hiperparametrów (Grid Search, Random Search) • Optymalizacja wydajności i kosztów (np. inference time) 8. Bezpieczeństwo i etyka AI • Ryzyka związane z błędnymi predykcjami • Ochrona danych osobowych (RODO) • Etyczne aspekty automatyzacji i decyzji podejmowanych przez model 9. Prezentacja i dokumentacja projektu • Przygotowanie dokumentacji technicznej i użytkownika • Demo działania aplikacji / systemu • Opis modelu, danych, sposobu walidacji i wniosków 10. Ewaluacja projektu i refleksja zespołowa • Ocena współpracy w zespole (peer review) • Lessons learned – co się udało, co można poprawić • Przygotowanie do publikacji open source lub udziału w konkursach AI
Systemy ekspertowe	1. Wprowadzenie do systemów ekspertowych • Definicja systemu ekspertowego i jego miejsce w dziedzinie AI • Cechy charakterystyczne: regułowość, wnioskowanie, baza wiedzy • Różnice między systemem ekspertowym a klasyczną aplikacją decyzyjną 2. Architektura systemu ekspertowego • Kluczowe komponenty: baza wiedzy, baza faktów, mechanizm wnioskowania, interfejs użytkownika • Rola silnika reguł, modułu wyjaśniającego i interfejsu akwizycji wiedzy • Modele hybrydowe (np. systemy ekspertowe z elementami ML) 3. Reprezentacja wiedzy

	• Reguły produkcyjne (IF–THEN), ramy (frames), drzewa decyzyjne • Logika rozmyta i niepewność w reprezentacji wiedzy • Fakty, wnioskowanie, hierarchie i sieci semantyczne 4. Mechanizmy wnioskowania • Wnioskowanie w przód (*forward chaining*) i wstecz (*backward chaining*) • Konflikt reguł i strategie rozwiązywania (np. priorytetyzacja, specyficzność) • Przykładowe silniki wnioskowania: CLIPS, Jess, Drools 5. Budowa i rozwój bazy wiedzy • Pozyskiwanie wiedzy od ekspertów dziedzinowych • Strukturalizacja wiedzy i walidacja poprawności • Rola inżyniera wiedzy w procesie budowy systemu 6. Języki i narzędzia do budowy systemów ekspertowych • CLIPS, Prolog, Jess – przykłady zastosowań • Silniki reguł w językach wysokiego poziomu (np. Python + PyKnow, Java + Drools) • Środowiska graficzne i eksperymentalne (np. Exsys, ESWin) 7. Zastosowania systemów ekspertowych • Medycyna (np. diagnozowanie chorób) • Rolnictwo, przemysł, logistyka, edukacja • Systemy doradcze i wspomagania decyzji (np. systemy kredytowe, eksperci HR) 8. Ocena jakości i skuteczności systemu ekspertowego • Metody testowania i walidacji reguł • Miary efektywności: trafność, kompletność, wiarygodność • Rola symulacji i testów użytkowników 9. Ograniczenia i wyzwania systemów ekspertowych • Problemy ze skalowalnością i aktualizacją wiedzy • Ograniczenia logiki binarnej – wprowadzenie do logiki rozmytej • Kwestie etyczne i odpowiedzialność za decyzje systemu 10. Projekt praktyczny • Budowa mini systemu ekspertowego w wybranym narzędziu (np. system doradczy, diagnozujący, klasyfikujący) • Opracowanie bazy reguł, logiki wnioskowania i interfejsu • Dokumentacja działania oraz testy przypadków użytkownika
Zaawansowane uczenie maszynowe	1. Przegląd klasycznych i nowoczesnych algorytmów ML • Przypomnienie podstaw (regresja, drzewa decyzyjne, SVM, k-NN) • Ensemble learning: Random Forest, Gradient Boosting (XGBoost, LightGBM) • Wyważenie dokładności vs złożoności modelu 2. Uczenie głębokie (Deep Learning) • Sieci neuronowe: architektura, aktywacje, funkcje strat • Konwolucyjne sieci neuronowe (CNN) – rozpoznawanie obrazów • Rekurencyjne sieci neuronowe (RNN, LSTM) – sekwencje i dane czasowe 3. Transfer learning i uczenie z niewielką ilością danych • Modele wstępnie wytrenowane (np. ResNet, BERT, GPT) • Fine-tuning vs feature extraction • Praktyczne zastosowania: klasyfikacja obrazów, analiza tekstu 4. Uczenie nienadzorowane i redukcja wymiarowości • Zaawansowane metody grupowania: Spectral Clustering, Gaussian Mixture Models • Autoenkodery i ich zastosowania • T-SNE, UMAP – wizualizacja danych wysokowymiarowych 5. Uczenie przez wzmacnianie (Reinforcement Learning) • Agent, środowisko, nagrody, strategie • Algorytmy: Q-learning, SARSA, Deep Q-Networks (DQN) • Przykłady: gry, robotyka, zarządzanie ruchem 6. Uczenie półnadzorowane i samo-nadzorowane

	• Semi-supervised learning: algorytmy propagacji etykiet • Contrastive learning – koncepcje i zastosowania • Nowoczesne techniki reprezentacji danych (np. SimCLR, BYOL) 7. Generatywne modele uczenia maszynowego • Modele probabilistyczne: Naive Bayes, Hidden Markov Models (HMM) • Generative Adversarial Networks (GAN) – architektura, zastosowania, problemy uczenia • Modele autoregresyjne i sekwencyjne (np. Transformer, GPT) 8. Zaawansowane metody ewaluacji modeli • Weryfikacja krzyżowa strat wieloklasowych i wieloetykietowych • Metryki niestandardowe: Log Loss, AUC-PR, Cohen's Kappa • Uczenie z niezrównoważonymi danymi (undersampling, SMOTE) 9. Objaśnialność i interpretowalność modeli (Explainable AI) • SHAP, LIME – wyjaśnianie decyzji modelu • Metody globalne i lokalne interpretacji • Problemy z "czarnymi skrzynkami" w deep learningu 10. Skalowanie i wdrażanie modeli ML • Przetwarzanie dużych zbiorów danych: ML z wykorzystaniem Spark, Dask • API modeli: FastAPI, Flask, TensorFlow Serving • Wdrażanie modeli w chmurze: Azure ML, AWS SageMaker, Vertex AI
Wizja komputerowa	1. Wprowadzenie do wizji komputerowej • Definicja i zastosowania wizji komputerowej • Różnica między przetwarzaniem obrazów a rozpoznawaniem wizualnym • Przegląd aplikacji: biometria, przemysł, motoryzacja, medycyna, rozrywka 2. Podstawy obrazów cyfrowych • Reprezentacja obrazów: piksele, przestrzenie barw (RGB, HSV, YCbCr) • Histogramy, jasność, kontrast • Operacje wstępne: progowanie, filtrowanie, normalizacja 3. Przetwarzanie i analiza obrazów • Filtrowanie liniowe i nieliniowe: medianowe, Sobela, Gaussa • Wykrywanie krawędzi i konturów (Canny, Laplace, Sobel) • Segmentacja obrazu: Watershed, thresholding, k-means 4. Wykrywanie i rozpoznawanie obiektów • Ekstrakcja cech: SIFT, SURF, ORB • Detekcja obiektów: Haar cascades, HOG + SVM • Śledzenie obiektów: optical flow, MeanShift, Kalman filter 5. Wprowadzenie do deep learningu w wizji komputerowej • Konwolucyjne sieci neuronowe (CNN) – podstawy • Architektury: LeNet, AlexNet, VGG, ResNet • Wykorzystanie modeli wstępnie wytrenowanych (transfer learning) 6. Rozpoznawanie obrazów i klasyfikacja • Klasyfikacja obrazów z użyciem CNN • Detekcja obiektów: YOLO, SSD, Faster R-CNN • Segmentacja semantyczna i instancyjna: U-Net, Mask R-CNN 7. Wizja komputerowa w czasie rzeczywistym • Przetwarzanie z kamery: OpenCV, MediaPipe • Śledzenie twarzy, dłoni, sylwetek • Optymalizacja modeli do działania na urządzeniach mobilnych i IoT 8. Rekonstrukcja i przetwarzanie 3D • Głębia i stereo vision • Chmury punktów i rekonstrukcja 3D (SfM, SLAM) • Kamery głębi (Intel RealSense, Kinect) 9. Uczenie modelu wizji komputerowej • Przygotowanie zbioru danych (np. COCO, ImageNet, własne dane)

	• Augmentacja danych i detekcja błędów anotacji • Ewaluacja modeli: precision, recall, IoU, mAP 10. Projekt praktyczny • Praca zespołowa lub indywidualna: np. rozpoznawanie twarzy, analiza ruchu, klasyfikacja zdjęć medycznych, analiza wideo • Przygotowanie dokumentacji technicznej, prezentacji i demo działania aplikacji
Sieci neuronowe i głębokie uczenie	1. Wprowadzenie do sieci neuronowych • Biologiczna inspiracja i podstawy teoretyczne • Sztuczny neuron i perceptron • Architektura sieci: warstwy, wagi, funkcje aktywacji 2. Propagacja i uczenie sieci • Forward propagation i funkcje strat • Algorytm backpropagation – aktualizacja wag • Gradient descent i jego warianty (SGD, Adam, RMSprop) 3. Podstawowe architektury sieci • Multilayer Perceptron (MLP) • Funkcje aktywacji: sigmoid, tanh, ReLU, leaky ReLU • Regularizacja: dropout, L1/L2, batch normalization 4. Konwolucyjne sieci neuronowe (CNN) • Warstwy konwolucyjne i poolingowe • Detekcja cech w obrazach (feature maps, kernels) • Przykładowe architektury: LeNet, AlexNet, VGG, ResNet 5. Rekurencyjne sieci neuronowe (RNN) • Przetwarzanie danych sekwencyjnych: tekst, sygnały, czas • Problemy RNN: zanikanie i eksplozja gradientu • LSTM i GRU – nowoczesne rozwiązania rekurencyjne 6. Transformery i modele językowe • Architektura Transformer: self-attention, encodery i decodery • Modele BERT, GPT i ich zastosowania • Transfer learning w NLP 7. Generatywne sieci neuronowe • Autoenkodery: kompresja i rekonstrukcja danych • Generative Adversarial Networks (GAN): generator vs dyskryminator • Przykłady zastosowań: generowanie obrazów, danych, głosów 8. Zaawansowane techniki treningowe • Tuning hiperparametrów i eksperymentowanie (grid/random search) • Augmentacja danych, early stopping, cross-validation • Monitorowanie treningu i metryk (TensorBoard, Weights & Biases) 9. Implementacja i narzędzia • Frameworki: TensorFlow, PyTorch, Keras • Tworzenie własnych warstw, modeli i funkcji strat • Eksport modeli, integracja z API (ONNX, Flask, FastAPI) 10. Zastosowania głębokiego uczenia • Wizja komputerowa (CV): klasyfikacja, detekcja obiektów • Przetwarzanie języka naturalnego (NLP): klasyfikacja tekstu, chatboty • Predykcja szeregów czasowych, analiza danych medycznych
Przetwarzanie języka naturalnego	1. Wprowadzenie do NLP • Czym jest NLP i gdzie się je stosuje? • Struktura języka: morfologia, składnia, semantyka, pragmatyka • NLP jako połączenie lingwistyki i sztucznej inteligencji 2. Reprezentacja tekstu • Tokenizacja, stemming, lematyzacja

	• Reprezentacja tekstu: Bag of Words, TF-IDF • Modele wektorowe słów: Word2Vec, GloVe, FastText 3. Analiza składniowa i morfologiczna • Rozpoznawanie części mowy (POS tagging) • Parsowanie składniowe – drzewa zależności i konstytuentów • Analiza fleksyjna i morfologiczna (np. dla języka polskiego) 4. Przetwarzanie semantyczne • Rozpoznawanie bytów nazwanych (NER – Named Entity Recognition) • Rozumienie relacji semantycznych: synonimia, antonimia, hiper/hiponimia • Rozwiązywanie koreferencji i analiza zależności kontekstowych 5. Klasyfikacja i ekstrakcja informacji z tekstu • Klasyfikacja dokumentów (spam, tematyka, emocje) • Wydobywanie kluczowych informacji (keyword extraction, relation extraction) • Analiza sentymentu i opinii (np. recenzje, media społecznościowe) 6. Modele językowe i sieci neuronowe w NLP • RNN, LSTM, GRU – przetwarzanie sekwencyjne • Transformery: architektura i zasada działania • Modele: BERT, RoBERTa, GPT, T5 – porównanie i zastosowania 7. Uczenie i fine-tuning modeli NLP • Uczenie nadzorowane na korpusach tekstowych • Transfer learning i fine-tuning pretrenowanych modeli (Hugging Face Transformers) • Zagadnienia związane z annotacją i etykietowaniem danych 8. Dialog i generowanie języka • Systemy pytanie-odpowiedź (QA), chatboty, asystenci głosowi • Generowanie tekstu: od n-gramów po modele generatywne (GPT, LLM) • Summarization, translation, paraphrasing 9. Wyzwania i etyka NLP • Uprzedzenia i dyskryminacja w modelach językowych (bias, fairness) • Wielojęzyczność, zasoby językowe niskozasobowe (low-resource languages) • Ochrona prywatności, RODO a teksty użytkownika 10. Projekt zaliczeniowy • Zespół lub osoba przygotowuje rozwiązanie NLP: klasyfikator, ekstraktor informacji, chatbot, analizator emocji • Dane rzeczywiste lub zbiór otwarty (np. IMDB, PolEmo, CoNLL) • Trening modelu, prezentacja wyników, dokumentacja
Optymalizacja stochastyczna	1. Wprowadzenie do optymalizacji stochastycznej • Różnice między optymalizacją deterministyczną a stochastyczną • Przykłady zastosowań: finanse, zarządzanie, AI • Klasyfikacja metod stochastycznych 2. Modele probabilistyczne i niepewność • Zmienne losowe i rozkłady • Funkcje celu zależne od oczekiwanych wartości • Ograniczenia probabilistyczne (chance constraints) 3. Optymalizacja oparta na symulacji • Monte Carlo i jego warianty • Symulacja scenariuszy • Szacowanie wartości oczekiwanej i przedziałów ufności 4. Programowanie stochastyczne • Modele jedno- i dwuetapowe • Podejście scenariuszowe • Przykłady: planowanie inwestycji, produkcji, portfela 5. Stochastic Gradient Descent (SGD) i jego warianty • Klasyczny SGD: idea, implementacja, zastosowania

	• Warianty: Momentum, Adam, RMSProp • Przykłady: regresja logistyczna, sieci neuronowe 6. Metaheurystyki stochastyczne • Algorytmy genetyczne • Symulowane wyżarzanie (Simulated Annealing) • Optymalizacja rojem cząstek (PSO) 7. Optymalizacja Bayesowska • Modele procesów Gaussowskich • Eksploracja vs eksploatacja • Funkcje akwizycji: UCB, EI, PI • Tuning hiperparametrów modeli ML 8. Programowanie dynamiczne i uczenie wzmacniane • Dynamic Programming (DP) i Approximate DP • Wstęp do Reinforcement Learning • Modelowanie decyzji sekwencyjnych pod niepewnością 9. Zastosowania praktyczne • Optymalizacja portfela inwestycyjnego • Harmonogramowanie z niepewnością • Uczenie maszynowe i dostrajanie modeli (model selection) 10. Projekt praktyczny / analiza przypadków • Definicja problemu optymalizacyjnego • Dobór metod • Implementacja i raportowanie wyników

IV. PROGRAM STUDIÓW

W ramach studiów I stopnia na kierunku Informatyka oferowane są następujące specjalności:

- Programowanie
- Cyberbezpieczeństwo
- Sztuczna Inteligencja

A) PRZYPORZĄDKOWANIE KIERUNKU STUDIÓW DO DYSYCYPLIN NAUKOWYCH

L.p.	Dyscypliny naukowe	% PUNKTÓW ECTS
1.	Informatyka techniczna i telekomunikacja	90
2.	Matematyka	10

B) PODSTAWOWE WSKAŹNIKI ECTS OKREŚLONE DLA PROGRAMU STUDIÓW

Nazwa wskaźnika	Liczba punktów ECTS/Liczba godzin
Łączna liczba punktów ECTS, jaką student musi uzyskać w ramach zajęć prowadzonych z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia	STUDIA NIESTACJONARNE 73,8
Łączna liczba punktów ECTS przyporządkowana zajęciom kształtującym umiejętności praktyczne	STUDIA NIESTACJONARNE 121,7
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	7
Łączna liczba punktów ECTS przyporządkowana zajęciom do wyboru	99
Łączna liczba punktów ECTS przyporządkowana praktykom zawodowym	38

C) WYMIAR, ZASADY I FORMY ODBYWANIA PRAKTYK ZAWODOWYCH

Praktyki zawodowe są integralną częścią procesu dydaktycznego, co zgodnie z wymaganiami programowymi dla studiów I stopnia, jest odzwierciedleniem ich zawodowego charakteru. Zgodnie z obowiązującym regulaminem studiów Wyższej Szkoły Bankowej w Poznaniu, Wydziału Ekonomicznego w Szczecinie, praktyki zawodowe są obowiązkowe (są przedmiotem).

- **Wymiar praktyk zawodowych**

Szczegóły związane z odbywaniem praktyk określa Dziekan Wydziału. Dla kierunku Informatyka I stopnia przewidziane są następujące regulacje: student ma możliwość realizacji praktyki w trakcie całego przebiegu studiów, już od pierwszego roku studiów. Od roku akademickiego 2019/20, zgodnie z regulacjami ustawy Prawo o szkolnictwie wyższym i nauce z dn. 20 lipca 2018, obowiązujący wymiar godzin praktyk na studiach pierwszego stopnia wynosi: **6 miesięcy/24 tygodnie/960 godzin.**

- **Zasady i formy odbywania praktyk zawodowych**

1) Biuro Karier i Praktyk (dalej: BKiP) jest organizatorem i koordynatorem praktyki zawodowej dla studentów studiów I i II stopnia;
2) BKiP wspiera studenta i doradza w zakresie poszukiwania miejsca praktyk;
3) BKiP prowadzi monitoring realizowanych praktyk;
4) Student ma możliwość zorganizowania praktyki:
   a) za pośrednictwem BKiP,
   b) samodzielnie.
5) Jeżeli student chce zorganizować praktykę **za pośrednictwem Biura Karier i Praktyk**, zobowiązany jest do:
   a) wypełnienia deklaracji udostępnionej w Extranecie w wersji elektronicznej lub osobiście w Biurze Karier i Praktyk w wersji papierowej w terminie określonym przez Biuro Karier i Praktyk, nie później niż na 2 tygodnie przed terminem rozpoczęcia praktyk,
   b) dostarczenia do Biura Karier i Praktyk CV w wersji papierowej lub elektronicznej.

6) Jeżeli student chce zorganizować praktykę **samodzielnie,** zobowiązany jest do wypełnienia deklaracji w wersji elektronicznej lub papierowej potwierdzonej przez praktykodawcę w terminie określonym przez Biuro Karier i Praktyk, jednak nie później niż na 2 tygodnie przed terminem rozpoczęcia praktyk.
7) Miejsce odbywania praktyki zatwierdza opiekun merytoryczny praktyk wyznaczony przez Dziekana Wydziału. Opiekun merytoryczny w razie wątpliwości co do miejsca odbywania praktyk przeprowadza szczegółową rozmowę ze studentem i opiekunem wyznaczonym ze strony firmy odnośnie kryteriów jakościowych doboru miejsca odbywania praktyk przez studenta oraz infrastruktury i wyposażenia miejsca odbywanych praktyk.
8) Po otrzymaniu przez studenta pozytywnej oceny dot. miejsca praktyki zawodowej przez opiekuna merytorycznego, BKiP przygotowuje dokumentację kierującą na praktykę zawodową.
9) Praktyka jest realizowana zgodnie z programem praktyk dla danego kierunku studiów.
10) Uczelnia nie pokrywa kosztów związanych z praktykami (np. ubezpieczenie NNW, OC, dojazdu, noclegu).
11) Student zobowiązany jest do rozliczenia praktyki zawodowej zgodnie z regulaminem praktyk w ciągu dwóch tygodni od dnia zakończenia praktyki zawodowej.
12) Dokumentacja z odbytej praktyki podlega ocenie formalnej przez BKiP oraz ocenie merytorycznej przez opiekuna kierunku.
13) Opiekun merytoryczny praktyk na podstawie dzienniczka praktyk oraz oceny opiekuna praktyk u praktykodawcy weryfikuje, czy student osiągnął zakładane efekty uczenia się i na tej podstawie zalicza praktykę zawodową.
14) Decyzję końcową o zaliczeniu praktyki zawodowej podejmuje Dziekan Wydziału.
15) Zaliczenie przez studenta praktyki w pełnym wymiarze jest warunkiem dopuszczenia studenta do egzaminu dyplomowego.

Zasady zaliczania praktyk na podstawie aktywności zawodowej i potwierdzonych efektów uczenia się.

1) Na pisemny wniosek student może ubiegać się o częściowe lub całkowite zaliczenie praktyk na podstawie wykonywanej pracy zawodowej trwającej minimum:
    a) 3 miesiące zatrudnienia (dotyczy studentów, którzy rozpoczęli studia do 30 września 2019 r. oraz wszystkich studentów studiów II stopnia),
    b) 6 miesięcy zatrudnienia (dotyczy studentów studiów I stopnia, którzy rozpoczęli studia po 01 października 2019 r.).
2) O wymiarze zaliczenia praktyk w całości lub części na podstawie wykonywanej pracy zawodowej decyduje Dziekan na podstawie złożonej dokumentacji. Decyzja jest podejmowana w przeciągu 2 tygodni od momentu złożenia w BKiP kompletnej dokumentacji. Przy ustaleniu zmniejszonego wymiaru praktyk brany jest pod uwagę staż pracy oraz jej zgodność z kierunkiem studiów lub specjalnością.
3) O zaliczenie praktyk może ubiegać się student, który:
    a) wykonuje lub wykonywał pracę bądź odbywał staż - w tym przypadku do wniosku należy dołączyć aktualne zaświadczenie o zatrudnieniu lub świadectwo pracy wraz z zakresem obowiązków oraz z potwierdzeniem realizacji efektów uczenia się w wykonywanej pracy zawodowej,
    b) pracuje (współpracuje) lub pracował (współpracował) w ramach własnej działalności gospodarczej – w tym przypadku do wniosku należy dołączyć zaświadczenie o prowadzeniu działalności gospodarczej wraz z potwierdzeniem realizacji efektów uczenia się w wykonywanej pracy zawodowej oraz aktualny wydruk Centralnej Ewidencji Informacji o Działalności Gospodarczej (CEIDG), Biuro Karier i Praktyk umawia studenta na rozmowę z opiekunem merytorycznym praktyk w celu potwierdzenia efektów uczenia się przez opiekuna merytorycznego. W trakcie spotkania, opiekun merytoryczny wypełnia formularz w którym zatwierdza zaliczenie praktyk i efektów uczenia się na podstawie rozmowy i dokumentacji przedstawionej przez studenta.

c) wykonuje lub wykonywał inne aktywności zawodowe - w tym przypadku do wniosku należy dołączyć dokument potwierdzający aktywność zawodową (np. referencje, zaświadczenie) oraz potwierdzenie realizacji efektów uczenia się w wykonywanej aktywności zawodowej podpisane przez uprawnioną do tego osobę Studentom będącym pracownikami służb mundurowych w uzasadnionych przypadkach związanych z koniecznością zachowania poufności informacji Dziekan może zaliczyć praktykę bez przekładania wszystkich lub części wymaganych dokumentów.

4) W przypadku częściowego zaliczenia praktyk student ma obowiązek zaliczenia pozostałej części zgodnie z programem praktyk, co jest warunkiem dopuszczania studenta do egzaminu dyplomowego.
5) W przypadku studentów I stopnia, gdzie wymiar praktyk wynosi 960 godzin, student może wnioskować o zaliczenie częściowe w wymiarze 160 godzin (1 miesiąc) co daje możliwość zaliczenia podstawowych modułów z programu praktyk jakąkolwiek aktywnością zawodową. Natomiast 800 godzin należy zrealizować zgodnie z kierunkiem studiów tak, aby student osiągnął efekty uczenia się założone w modułach programowych praktyk.

D) SPOSOBY WERYFIKACJI OCENY EFEKTÓW UCZENIA SIĘ OSIĄGANYCH PRZEZ STUDENTA W TRAKCIE CAŁEGO CYKLU KSZTAŁCENIA

Walidacja efektów uczenia się założonych w programie studiów, uszczegółowionych w kartach przedmiotu poprzez przedmiotowe efekty uczenia się, dotyczy trzech obszarów: wiedzy, umiejętności i kompetencji społecznych. Niektóre z metod weryfikacji efektów uczenia się pozwalają na ocenę w więcej niż jednym obszarze.

Metody weryfikacji oceny efektów uczenia się:

Kategoria	Wiedza	Umiejętności	Kompetencje społeczne
Metody:	- Egzaminy ustne – standaryzowane - Egzaminy pisemne – pytania otwarte, testy jedno –, bądź wielokrotnego wyboru, tekst z lukami, mini – testy, zadania, zadania rachunkowe - Ocena prac pisemnych, indywidualnych lub zespołowych, np.: projekty, scenariusze działań, analizy przypadku, symulacje procesów, recenzje artykułów - Ocena prezentacji projektu zespołowego lub indywidualnego w oparciu o prezentacje multimedialne, scenariusze, symulacje etc.	- Egzaminy ustne i pisemne - Obserwacja wykonania zadania lub projektu indywidualnego lub zespołowego - Ocena pracy indywidualnej lub zespołowej podczas zajęć - Ocena aktywności podczas działań praktycznych - Ocena prezentacji/projektu rozwiązującego problem inżynierski - Obserwacja i analiza prac lub innych wyników działań studenckich	- Obserwacja i analiza projektów lub zadań pod kątem gotowości do podejmowania działań zgodnych ze wskazanymi kompetencjami społecznymi, - Obserwacja zachowań i kompetencji społecznych podczas działań praktycznych - Samoocena - Ocena aktywności poza zajęciami – udział w kołach zainteresowań, konferencjach naukowych, konkursach, projektach

Wskazane metody weryfikacji wykorzystywane są również w trybie zdalnym.

E) PLANY STUDIÓW

Uniwersytet WSBM w Poznaniu, Wydział Ekonomiczny w Szczecinie Plan studiów dla naboru 2026/2027

Studia niestacjonarne online - I stopnia - **Informatyka**

Lp.	Przedmiot	L godz	Ects	Forma zaliczenia			Rok I												Rok II										Rok III										ROK IV					
							Sem 1						Sem 2						Sem 3						Sem 4						Sem 5						Sem 6						Sem 7					
				E	Zo	Zz	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS	W	C	L	P	E	ECTS
1	BHP	4	0			x					4	0																																		
2	Komunikacja społeczna	16	2		x			16				2																																		
3	Podstawy ekonomii	38	2	es	x			6			32	2																																		
4	Matematyka dyskretna	30	5	es	x			24			6	5																																		
5	Narzędzia informatyki	16	2		x						16	2																																		
6	Wprowadzenie do informatyki	16	1	es	x						16	1																																		
7	Podstawy programowania	48	5	es	x				24		24	5																																		
8	Matematyka dla inżynierów	80	8	es	x			24			16	4		24			16	4																												
9	Metodyka pracy projektowej	116	7		x						42	2					32	2					32	2					10	1															
10	Wyzwania rynku pracy	20	1		x									4			16	1																											
11	Podstawy zarządzania	48	2		x									8			40	2																											
12	Algorytmy i struktury danych	32	4	es	x										16		16	4																											
13	Architektura komputerów	32	2	es	x										16		16	2																											
14	Systemy operacyjne	32	3	es	x								16		16			3																											
15	Podstawy elektrotechniki	21	2		x										16		5	2																											
16	Probabilistyka i statystyka	30	4	es	x																24		6	4																					
17	Programowanie obiektowe	40	4	es	x																24		16	4																					
18	Sieci komputerowe	32	3	es	x														16		16			3																					
19	Wprowadzenie do baz danych	32	3		x																16		16	3																					
20	Laboratorium inżynierskie	29	3		x																24		5	3																					
21	Język obcy	150	10		x	x														15			60	5		15			60	5															
22	Programowanie aplikacji internetowych	32	3	es	x																						32			3															
23	Bazy danych	32	3	es	x																						32			3															
24	Zarządzanie projektami informatycznymi	24	3	es	x																								24	3															
25	Laboratorium nowych technologii	29	3		x																						24		5	3															
26	Metody obliczeniowe	32	3		x																						16		16	3															
27	Programowanie zaawansowane	32	3	es	x																											24		8	3									
28	Podstawy ochrony danych	32	2	es	x																										8		16		8	2									
29	Analiza i projektowanie systemów informatycznych	40	4	es	x																											24		16	4									
30	Przetwarzanie danych multimedialnych	32	3	es	x																															24		8	3							
31	Software Engineering	32	2	es	x																															16		16	2							
32	Testowanie oprogramowania	16	2		x																															8		8	2							
33	Programowanie w zastosowaniach	24	3		x																																						8		16			3
34	Cultural Differences	26	2		x																																						6			20	2
35	Egzamin semestralny	12	12	x				2				2		2				2		2				2		2				2		2				2		2								
36	Przedmioty specjalnościowe	288	36		x																										24		48		24	12	24		48		24	12	24		48		24	12
37	Seminarium dyplomowe	50	15			x																										10						4		20				5		20				6
38	Praktyka zawodowa	960	38			x																																		480					19		480				19
	Liczba godzin zorganizowanych	2 555	210				0	72	24	0	156	25	16	38	64	0	141	22	16	17	104	0	135	26	0	17	104	0	115	23	32	12	112	0	56	27	24	502	96	0	56	45	32	506	64	0	44	42
	W tym liczba godzin kontaktowych	1 852					252						259						272						236						212						678						646					
	Liczba godzin kontaktowych bez praktyk	892																																														

Plan studiów w UWSBM w Poznaniu Wydział Ekonomiczny w Szczecinie

Studia niestacjonarne (online)- I stopnia (studia inżynierskie)- **Informatyka (dla naboru 2026/2027)**

specjalność: Cyberbezpieczeństwo

L.P.	PRZEDMIOT	Ilość godzin	Forma zaliczenia			ROK III												ROK IV					
						sem 5						sem 6						sem 7					
			E	Zo	Zz	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS
1	Bezpieczeństwo oprogramowania	**40**		x				16		24	4												
2	Wprowadzenia do chmur obliczeniowych	**16**		x				16			4												
3	Bezpieczeństwo i ochrona danych	**40**		x		24		16			4												
4	Elementy kryptografii	**28**		x										16		12	4						
5	Systemy zarządzania bezpieczeństwem informacji	**40**		x								24		16			4						
6	Podstawy monitoringu systemów i aplikacji	**28**		x										16		12	4						
7	Wykrywanie i analiza zagrożeń w sieci	**16**		x																16			4
8	Ochrona danych w chmurze	**16**		x																16			4
9	Zarządzanie projektami bezpieczeństwa IT	**64**		x														24		16		24	4
	RAZEM	**288**				**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**
	RAZEM w semestrze	**288**				**96**						**96**						**96**					
	ECTS w semestrze	**36**				**12**						**12**						**12**					
	RAZEM godziny kontaktowe w semestrze	**216**				**72**						**72**						**72**					

Plan studiów w UWSBM w Poznaniu Wydział Ekonomiczny w Szczecinie

Studia niestacjonarne (online)- I stopnia (studia inżynierskie)- **Informatyka (dla naboru 2026/2027)**

specjalność: Programowanie

L.P.	PRZEDMIOT	Ilość godzin	Forma zaliczenia			ROK III												ROK IV					
						sem 5						sem 6						sem 7					
			E	Zo	Zz	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS
1	Testowanie i jakość oprogramowania	**40**		x				16		24	4												
2	Projektowanie i implementacja aplikacji rozproszonych	**16**		x				16			4												
3	Programowanie w zespole – studium przypadku	**40**		x		24		16			4												
4	Wzorce projektowe	**28**		x										16		12	4						
5	Zaawansowana inżynieria kodu	**40**		x								24		16			4						
6	Integracja oprogramowania z platformą Azure	**28**		x										16		12	4						
7	□ Programowanie aplikacji internetowych MVC API	**16**		x																16			4
8	Projekt systemu informatycznego	**16**		x																16			4
9	Algorytmizacja	**64**		x														24		16		24	4
	RAZEM	**288**				**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**
	RAZEM w semestrze	**288**				**96**						**96**						**96**					
	ECTS w semestrze	**36**				**12**						**12**						**12**					
	RAZEM godziny kontaktowe w semestrze	**216**				**72**						**72**						**72**					

Plan studiów w UWSBM w Poznaniu Wydział Ekonomiczny w Szczecinie

Studia niestacjonarne (online)- I stopnia (studia inżynierskie)- **Informatyka (dla naboru 2026/2027)**

specjalność: Sztuczna inteligencja

L.P.	PRZEDMIOT	Ilość godzin	Forma zaliczenia			ROK III												ROK IV					
						sem 5						sem 6						sem 7					
			E	Zo	Zz	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS	W	Ć	L	P	E	ECTS
1	Podstawy uczenia maszynowego	**40**		x				16		24	4												
2	Systemy wspomagania decyzji	**16**		x				16			4												
3	Projekt zespołowy - rozwiązania AI	**40**		x		24		16			4												
4	Systemy ekspertowe	**28**		x										16		12	4						
5	Zaawansowane uczenie maszynowe	**40**		x								24		16			4						
6	Wizja komputerowa	**28**		x										16		12	4						
7	Sieci neuronowe i głębokie uczenie	**16**		x																16			4
8	Przetwarzanie języka naturalnego	**16**		x																16			4
9	Optymalizacja stochastyczna	**64**		x														24		16		24	4
	RAZEM	**288**				**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**	**24**	**0**	**48**	**0**	**24**	**12**
	RAZEM w semestrze	**288**				**96**						**96**						**96**					
	ECTS w semestrze	**36**				**12**						**12**						**12**					
	RAZEM godziny kontaktowe w semestrze	**216**				**72**						**72**						**72**					