

UNIwersYTET W SIEDLCACH

INSTYTUT INFORMATYKI

Kierunek INFORMATYKA

INFORMATOR

SYLABUS

Obowiązuje w roku akademickim 2025/26

studia I stopnia

(inżynierskie, profil praktyczny)

czas trwania: 7 semestrów

Siedlce 2025

Semestr 6

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Seminarium dyplomowe I
Nazwa w języku angielskim:	Diploma seminar I	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	1	
Imię i nazwisko koordynatora przedmiotu:		dr hab. Stanisław Ambroszkiewicz
Imię i nazwisko prowadzących zajęcia:		dr hab. Stanisław Ambroszkiewicz
Założenia i cele przedmiotu:		Student posiada wiedzę nabytą w czasie studiów do przygotowania pracy inżynierskiej z Informatyki. Celem zajęć jest zapoznanie studenta z metodyką przygotowania pracy dyplomowej, w tym umiejętności wykonania przez studentów zaawansowanego projektu inżynierskiego z Informatyki.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	znaczenie Informatyki, zna i rozumie jej rolę w życiu społeczeństwa oraz zagrożenia związane z jej zastosowaniami i uwzględnia te aspekty w swojej pracy dyplomowej	K_W03
W_02	zagadnienia z zakresu ochrony własności intelektualnej	K_W04
U_01	UMIEJĘTNOŚCI Student potrafi:	
	pozyskiwać informacje z literatury, baz danych i innych, w tym zwłaszcza internetowych źródeł; potrafi analizować, interpretować oraz integrować uzyskane informacje, a także oceniać ich użyteczność w aspekcie wykonywanej pracy kwalifikacyjnej, wyciągać wnioski oraz formułować i uzasadniać opinie	
U_02	wykorzystać poznane metody i modele matematyczne — w razie potrzeby odpowiednio je modyfikując — do analizy i projektowania systemów informatycznych w zakresie Informatyki	K_U02, K_U09
U_03	posłużyć się właściwie środowiskami programistycznymi i narzędziami komputerowymi wspomagającymi proces projektowania, implementacji i testowania systemów informatycznych, oraz zaplanować realizację poszczególnych etapów rozwiązywania zadań z zakresu Informatyki	K_U13
U_04	ocenić i porównać rozwiązania projektowe systemów informatycznych ze względu na zadane kryteria użytkowe i ekonomiczne	K_U16

U_05	stosować w rozwiązywanych zadaniach z Informatyki standardy i normy inżynierskie i technologie informatyczne wykorzystując zdobyte doświadczenie w środowisku zajmującym się zawodowo działalnością inżynierską	K_U14
U_06	dostrzegać aspekty pozatechniczne, w tym środowiskowe, ekonomiczne i prawne przy projektowaniu, implementowaniu i wdrażaniu systemów informatycznych	K_U23
KOMPETENCJE SPOŁECZNE Student jest gotów do:		
K_01	odpowiedzialnego pełnienia roli zawodowej informatyka, w tym do przestrzegania zasad etyki zawodowej i wymagania tego od innych i do dbania o dorobek i tradycję zawodu informatyka	K_K04
K_02	krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu Informatyki	K_K01
K_03	myślenia i działania w sposób przedsiębiorczy	K_K03
Forma i typy zajęć:		Studia stacjonarne: 15 godz. Studia niestacjonarne: 9 godz.
Wymagania wstępne i dodatkowe:		
Znajomość treści nauczania w dotychczasowym przebiegu studiów		
Treści modułu kształcenia:		
Zawierają elementy wykładów przewidzianych kursem I stopnia studiów inżynierskich na kierunku Informatyka. Największy nacisk jest położony na podnoszenie umiejętności i kompetencji w zakresie opracowywania i prezentowania wyników oraz korzystania z fachowej literatury, także w języku angielskim. Główne zagadnienia: 1. Precyzowanie obszaru zainteresowań dyplomantów. Omówienie przez prowadzącego proponowanej tematyki prac dyplomowych. 2. Sposoby zbierania materiałów źródłowych. Omówienie przez prowadzącego źródeł informacji naukowej oraz sposobów jej wyszukiwania i wykorzystywania. 3. Omówienie przez prowadzącego zasad przygotowywania treści referatów i ich multimedialnej prezentacji. 4. Wybór kierownika i tematu pracy dyplomowej. 5. Opis tematyki i problemów związanych z wybranym tematem, w tym przegląd literatury dotyczący wybranego tematu. 6. Wstępny projekt: ➤ jeśli jest to aplikacja programowa, to wstępny projekt zawiera: wymagania funkcjonalne i nie-funkcjonalne, projekt bazy danych, abstrakcyjna architektura funkcjonalna (podział na komponenty i komunikacja pomiędzy nimi). ➤ jeśli, dodatkowo, projekt ma rozwiązywać problem z informatyki teoretycznej, to ten problem oraz ogólny proponowany sposób jego rozwiązania powinien się znaleźć w tym projekcie wstępnym. 7. Prezentacja przez studentów referatów z zakresu wstępnie wybranej tematyki i celów ich prac dyplomowych – z wykorzystaniem środków multimedialnych, dyskusja i ewentualna korekta tematów prac. 8. Dyskusja mająca na celu ostateczne sformułowanie tematów prac dyplomowych i określenie ich zakresu.		

Literatura podstawowa:	
1. Specyficzna dla danej dziedziny i konkretnego tematu zlecona przez opiekuna/promotora pracy oraz samodzielnie wyszukane przez studenta w dostępnych źródłach (również elektronicznych) zarówno w języku polskim jak i obcym oraz, m.in.: Zarządzenie nr 26/2016 Rektora Uniwersytetu Przyrodniczo-Humanistycznego w Siedlcach z dnia 22 kwietnia 2016 roku w sprawie określenia warunków, jakim powinna odpowiadać praca dyplomowa oraz zasad ich archiwizowania 2. Sobaniec C.: Jak pisać pracę inżynierską/magisterską. www.cs.put.poznan.pl/sobaniec/edu/jak_pisacmgr.pdf 3. Starecki T.: Praca dyplomowa – jak realizować, jak pisać i dlaczego. www.ise.pw.edu.pl/impuls/Dyplom.pdf	
Literatura dodatkowa:	
1. Kwaśniewski A.: Jak pisać pracę dyplomową. http://zpt2.tele.pw.edu.pl/~andrzej/TP/wykład/wykład-pdf/TPpraca_dypl.pdf 2. Drozdowski M. Jak pisać prace dyplomową/magisterską: http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak_mgr.html .	
Planowane formy/działania/metody dydaktyczne:	
Dyskusja, prezentacja multimedialna, prezentacja ustna. Prezentacja celów pracy dyplomowej Prezentacja koncepcji pracy dyplomowej z zastosowaniem informatycznych programów, systemów oraz technik audiowizualnych	
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
Symbol efektu	Metody weryfikacji efektów uczenia się
W_01_W_02	w sposób ciągły przez prowadzącego seminarium dyplomowe, na podstawie udziału w dyskusji oraz podczas przedstawianych prezentacji na poszczególnych etapach pracy dyplomowej (I – uzasadnienie wyboru tematu, studia literaturowe, II – postawienie hipotezy, omówienie wybranej metody badawczej i przedstawienie uzyskanych wyników/projektów/rozwiązań, wnioski, weryfikacja postawionej hipotezy)
U_01-U_06	w sposób ciągły przez prowadzącego seminarium dyplomowe, na podstawie udziału w dyskusji oraz podczas przedstawianych prezentacji na poszczególnych etapach pracy dyplomowej (I – uzasadnienie wyboru tematu, studia literaturowe, II – postawienie hipotezy, omówienie wybranej metody badawczej i przedstawienie uzyskanych wyników/projektów/rozwiązań, wnioski, weryfikacja postawionej hipotezy)
K_01-K_03	w toku zajęć seminaryjnych na podstawie przygotowania, udziału i aktywności poszczególnych studentów w zajęciach
Forma i warunki zaliczenia:	
Moduł kończy się zaliczeniem. Wybór tematyki i przegląd literatury, akceptacja kierownika pracy, przygotowanie projektu wstępnego oraz prezentacji dotyczącej opracowywanej pracy inżynierskiej. Wszystko to wymaga akceptacji kierownika pracy dyplomowej. Zaliczenie odbywa się po przygotowaniu prezentacji zaakceptowanej przez opiekuna dyplomanta.	
Bilans punktów ECTS:	
Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w zajęciach	15 godzin

Przygotowanie prezentacji multimedialnej	5 godzin
Samodzielne studia literaturowe	5 godzin
Sumaryczne obciążenie pracą studenta	25
Punkty ECTS za przedmiot	1
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w zajęciach	9 godzin
Przygotowanie prezentacji multimedialnej	8 godzin
Samodzielne studia literaturowe	8 godzin
Sumaryczne obciążenie pracą studenta	25
Punkty ECTS za przedmiot	1

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Praktyka zawodowa III
Nazwa w języku angielskim:	Apprenticeship III	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	10	
Imię i nazwisko koordynatora przedmiotu:		dr Marcin Stępiak
Imię i nazwisko prowadzących zajęcia:		Osoba delegowana z firmy/instytucji
Założenia i cele przedmiotu:		Zakłada się, że student ma wiedzę o funkcjonowaniu, projektowaniu i implementacji systemów informatycznych. Dodatkowo ma on wiedzę i umiejętności i kompetencje społeczne nabyte podczas dotychczas odbytych praktyk zawodowych. Cele praktyki zawodowej III: • pogłębienie umiejętności zawodowych studenta, • pogłębienie specjalistycznej wiedzy, umiejętności i kompetencji związanych z funkcjonowaniem firmy/instytucji w zakresie stosowanych systemów informatycznych oraz rozwoju istniejących systemów lub wytwarzania nowych aplikacji, • pogłębienie i wykorzystanie w praktyce wiedzy i umiejętności nabytych podczas nauki oraz poprzednich etapów praktyki zawodowej, zwłaszcza tych związanych z wybraną specjalnością, • zdobycie wiedzy i umiejętności związanych ze sposobami organizacji pracy indywidualnej i zespołowej, • nawiązanie kontaktów zawodowych ułatwiających poszukiwanie pracy, firmy/instytucji do odbycia kolejnych etapów praktyk, czy realizacji prac dyplomowych.
Symbol efektu	Efekty uczenia się	
	WIEDZA	
		Symbol efektu kierunkowego

	Student zna i rozumie:	
W_01	Posiada wiedzę w zakresie organizacji i zarządzania czasem oraz budowy harmonogramu pracy indywidualnej i zespołowej.	K_W12
W_02	Posiada pogłębioną specjalistyczną wiedzę dziedzinową związaną z systemami i narzędziami informatycznymi wykorzystywanymi w firmie.	K_W12
W_03	w zaawansowanym stopniu zagadnienia związane z wytwarzaniem oprogramowania lub administrowaniem systemami informatycznymi lub wykorzystania narzędzi informatycznych w pracy.	K_W06, K_W07
W_04	Student orientuje się w potrzebach rynku pracy. Zna relację między wymaganiami pracodawców a wiedzą zdobytą w trakcie zajęć.	K_W12, K_W03
Symbol efektu	UMIEJĘTNOŚCI Student potrafi:	Symbol efektu kierunkowego
U_01	planować czas pracy, nadawać priorytety zadaniom i je terminowo realizować. Potrafi wykorzystać wiedzę teoretyczną, praktyczną oraz narzędzia informatyczne do realizacji postawionych zadań.	K_U05, K_U23, K_U24
U_02	zadbać o własny wizerunek zawodowy. Potrafi zidentyfikować kierunki dalszego rozwoju na podstawie pozyskanej wiedzy, umiejętności oraz doświadczeń zawodowych. Potrafi nawiązywać i utrzymywać kontakty zawodowe.	K_U04, K_U06, K_U23
U_03	wykorzystać w praktyce wiedzę i umiejętności związane z realizowaną specjalnością. Potrafi w praktyce stosować zasady bezpieczeństwa i ergonomii pracy.	K_U14, K_U23, K_U25
Symbol efektu	KOMPETENCJE SPOŁECZNE Student jest gotów do:	Symbol efektu kierunkowego
K_01	odpowiedzialnego pełnienia roli zawodowej informatyka, w tym do przestrzegania zasad etyki zawodowej i wymagania tego od innych i do dbania o dorobek i tradycję zawodu informatyka.	K_K04
K_02	Krytycznej oceny działań własnych, działania zespołów, którymi kieruje i organizacji, w której uczestniczy. Potrafi dokonać samooceny własnych kompetencji i doskonalenia swoich kwalifikacji zawodowych.	K_K01, K_K03
Forma i typy zajęć:	Studia stacjonarne i niestacjonarne: praktyka (320 godzin)	
Wymagania wstępne i dodatkowe:		
1. Podstawowa wiedza z zakresu funkcjonowania, projektowania i implementacji systemów informatycznych.		
2. Wiedza, umiejętności i kompetencje społeczne nabyte podczas praktyki po pierwszym i drugim roku studiów.		
Treści modułu kształcenia:		
Szczegółowe treści w zależności od miejsca wykonywania praktyk zawodowych. Podczas drugich praktyk mogą one obejmować między innymi:		

1. Poznanie zasad bezpieczeństwa i higieny pracy na danym stanowisku oraz uwarunkowania prawne i etyczne stosownie do wykonywanych obowiązków.
2. Poznanie specyfiki działania przedsiębiorstwa, w którym jest odbywana praktyka.
3. Rozpoznanie obszarów działalności firmy wspomaganych komputerowo.
4. Zapoznanie z systemami i narzędziami informatycznymi wspomagającymi działalność firmy. W szczególności należy:
 - a. zapoznać się z dokumentacją techniczną sprzętu i oprogramowania,
 - b. rozpoznawać i rozwiązywać problemy związane z eksploatacją sprzętu i oprogramowania,
 - c. studiować możliwości optymalizacji, rozbudowy i modyfikacji infrastruktury teleinformatycznej, zgodnie z aktualnymi tendencjami rozwojowymi.
5. Dokonanie oceny istniejącej infrastruktury i wykorzystywanych technologii informatycznych w przedsiębiorstwie pod kątem zgodności ze standardami oraz możliwości rozwoju i współpracy z innymi rozwiązaniami.
6. Ocena aktualnego stanu oraz przyszłych potrzeb systemów informatycznych.
7. Współudział w projektowaniu nowych i ulepszaniu istniejących systemów informatycznych, biorąc pod uwagę:
 - a. wymagania i cele stawiane przed systemem informatycznym,
 - b. politykę bezpieczeństwa oraz procedury organizacyjne dotyczące wykorzystania infrastruktury informatycznej,
 - c. napotymane ograniczenia techniczne i biznesowe,
 - d. zagadnienia związane ze zwrotem kosztów inwestycji.
8. Realizacja zadań związanych z kierunkiem informatyka, a w szczególności ze specjalnością wybraną przez studenta odbywającego praktyki.
9. Prowadzenie dokumentacji przebiegu praktyk.

Literatura podstawowa:

Literatura podana przez opiekuna praktyk z ramienia zakładu pracy: dokumenty wytworzone w danym zakładzie pracy; obowiązujące regulaminy, stosowane akty prawne i normatywne; instrukcje obsługi urządzeń; opisy stosowanych procedur.

Literatura dodatkowa:

Regulamin praktyk

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
---------------	--

W_01-W_04	Podstawą weryfikacji efektów uczenia się będą: ocena praktykanta dokonana przez opiekuna praktyki z ramienia zakładu pracy; ankieta wypełniona przez praktykanta i opiekuna praktykanta i dostarczona do uczelni; ocena formalna i merytoryczna dziennika praktyk prowadzonego przez praktykanta a dokonana przez opiekuna praktyki z ramienia UwS a także wyniki kontroli praktyk dokonywanych przez opiekuna praktyk ze strony UwS oraz innych osób – wyznaczonych przez Dziekana.
U_01- U_03	Podstawą weryfikacji efektów uczenia się będą: ocena praktykanta dokonana przez opiekuna praktyki z ramienia zakładu pracy; ankieta wypełniona przez praktykanta i opiekuna praktykanta i dostarczona do uczelni; ocena formalna i merytoryczna dziennika praktyk prowadzonego przez praktykanta a dokonana przez opiekuna praktyki z ramienia UwS a także wyniki kontroli praktyk dokonywanych przez opiekuna praktyk ze strony UwS oraz innych osób – wyznaczonych przez Dziekana.
K_01- K_02	Podstawą weryfikacji efektów uczenia się będą: ocena praktykanta dokonana przez opiekuna praktyki z ramienia zakładu pracy; ankieta wypełniona przez praktykanta i opiekuna praktykanta i dostarczona do uczelni; ocena formalna i merytoryczna dziennika praktyk prowadzonego przez praktykanta a dokonana przez opiekuna praktyki z ramienia UwS a także wyniki kontroli praktyk dokonywanych przez opiekuna praktyk ze strony UwS oraz innych osób – wyznaczonych przez Dziekana.

Forma i warunki zaliczenia:

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

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

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

Bilans punktów ECTS:

Studia stacjonarne

Aktywność

Obciążenie studenta

Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki

320 godzin

Bezpośrednia komunikacja z przełożonymi i członkami zespołu na terenie zakładu pracy – miejscu odbywania stażu	50 godz.
Sumaryczne obciążenie pracą studenta	320 godzin
Punkty ECTS za przedmiot	10
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki	270 godzin
Bezpośrednia komunikacja z przełożonymi i członkami zespołu na terenie zakładu pracy – miejscu odbywania stażu	50 godz.
Sumaryczne obciążenie pracą studenta	320 godzin
Punkty ECTS za przedmiot	10

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Uczenie maszynowe
Nazwa w języku angielskim:	Machine learning	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Piotr Świtalski
Imię i nazwisko prowadzących zajęcia:		dr Piotr Świtalski, mgr Mateusz Przychodzki
Założenia i cele przedmiotu:		Założeniem tego przedmiotu jest znajomość przez słuchaczy zagadnień związanych z metodami sztucznej inteligencji oraz analizy danych. Celem przedmiotu jest zaznajomienie studentów z problematyką metod uczenia maszynowego, m.in.: • metodami klasyfikacji danych, • metodami regresji danych, • typowymi algorytmami uczenia maszynowego, • wprowadzeniem do generatywnych sieci przeciwnych.
Symbol efektu	Efekty uczenia się	
	WIEDZA Student zna i rozumie:	
W_01	najważniejsze paradygmaty i metody uczenia maszynowego	K_W01, K_W10
W_02	algorytmy nadzorowanego i nienadzorowanego uczenia maszynowego	K_W10
W_03	jak ocenić i poprawić jakość modeli uczenia maszynowego	K_W01, K_W10
UMIEJĘTNOŚCI Student potrafi:		
U_01	wykorzystać biblioteki implementujące algorytmy uczenia maszynowego do rozwiązania postawionego problemu	K_U02, K_U22
U_02	zaprojektować i przygotować implementację danego algorytmu uczenia maszynowego, odpowiednią dla problemu i dopasowaną do posiadanych danych	K_U07, K_U09
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
Warunkiem uczestnictwa w zajęciach jest znajomość zagadnień z przedmiotów „Analiza danych” oraz „Sztuczna inteligencja” lub znajomość literatury obowiązującej w tych przedmiotach.		
Treści modułu kształcenia:		

Treści dotyczące wykładu:

1. **Wprowadzenie do uczenia maszynowego.** Rodzaje systemów uczenia maszynowego. Uczenie nadzorowane i uczenie nienadzorowane. Uczenie półnadzorowane. Uczenie przez wzmacnianie. Uczenie wsadowe i uczenie przyrostowe. Główne problemy uczenia maszynowego.
2. **Ocena przydatności danych.** Źródła danych. Przykład oceny przydatności danych źródłowych. Rozkład częstości zmiennych. Zmienne numeryczne i kategoriowe. Brakujące dane. Reprezentatywność danych. Duplikaty. Szeregi czasowe.
3. **Klasyfikacja.** Zbiór danych MNIST. Podział zbioru danych. Uczenie klasyfikatora binarnego. Miary wydajności. Macierz pomyłek. Precyzja klasyfikatora. Pełność. Wynik F1. Wykres krzywej ROC. Klasyfikacja wieloklasowa. Analiza błędów. Klasyfikacja wieloetykietowa. Klasyfikacja wielowyjściowa.
4. **Regresja.** Model regresji liniowej. Gradient prosty. Stochastyczny spadek wzdłuż gradientu. Regresja wielomianowa. Krzywe uczenia. Regresja logistyczna. Regresja softmax.
5. **Maszyny wektorów nośnych.** Liniowa klasyfikacja SVM. Klasyfikacja miękkiego marginesu. Nieliniowa klasyfikacja SVM. Jądro wielomianowe. Cechy podobieństwa. Gaussowskie jądro RBF. Regresja SVM. Funkcja decyzyjna i prognozy. Programowanie kwadratowe. Problem dualny. Kernelizowane maszyny SVM. Przyrostowe maszyny SVM.
6. **Drzewa decyzyjne i lasy losowe.** Uczenie i wizualizowanie drzewa decyzyjnego. Wyliczanie prognoz. Szacowanie prawdopodobieństw przynależności do klas. Algorytm uczący CART. Entropia jako miara zanieczyszczenia. Hiperparametry regularyzacyjne. Drzewa decyzyjne w regresji. Niestabilność. Lasy losowe. Istotność cech w lasach losowych.
7. **Techniki uczenia nienadzorowanego.** Klasyfikacja technik uczenia nienadzorowanego. Algorytm centroidów (algorytm k-średnich). Wyszukiwanie optymalnej liczby skupień. Analiza skupień w segmentacji obrazu. Analiza skupień w przetwarzaniu wstępnym. Analiza skupień w uczeniu półnadzorowanym. Algorytm DBSCAN. Inne algorytmy analizy skupień.
8. **Uczenie przez wzmacnianie.** Uczenie się optymalizowania nagród. Wyszukiwanie strategii. Wprowadzenie do narzędzia OpenAI Gym. Sieci neuronowe jako strategie. Ocenianie czynności: problem przypisania zadłości. Gradienty strategii. Procesy decyzyjne Markowa. Uczenie metodą różnic czasowych. Q-uczenie. Przegląd popularnych algorytmów RN. AutoML i AutoGluon.
9. **Przetwarzanie sekwencji za pomocą sieci rekurencyjnych i splotowych.** Neurony i warstwy rekurencyjne. Komórki pamięci. Sekwencje wejść i wyjść. Uczenie sieci rekurencyjnych. Prognozowanie szeregów czasowych. Obsługa długich sekwencji. Duże modele językowe LLM.
10. **Uczenie reprezentacji za pomocą autokoderów.** Efektywne reprezentacje danych. Analiza PCA za pomocą niedopełnionego autokodera liniowego. Autokodery stosowe. Autokodery splotowe. Autokodery rekurencyjne. Autokodery odsumiające. Autokodery rzadkie. Autokodery wariacyjne.
11. **Generatywne sieci przeciwstawne.** Generator i dyskryminator. Problemy związane z uczeniem sieci GAN. Głębokie splotowe sieci GAN. Rozrost progresywny sieci GAN. Sieci StyleGAN.

Treści dotyczące laboratorium:

1. **Zastosowania Pythona w uczeniu maszynowym.** Pliki. Kolekcje. Typy danych. Menedżer pakietów pip. Struktury danych. Funkcje wyższego rzędu.
2. **Jupyter Notebook – NumPy, Pandas.** Instalacja Jupyter Notebook. Interfejs użytkownika. Komórki w Jupyter Notebook. Wykonywanie kodu w Jupyter Notebook. Zapisywanie i eksportowanie notatników. Biblioteka NumPy. Pandas – operacje na DataFrame.
3. **Wizualizacja danych z Matplotlib.** Wstęp do Matplotlib. Podstawowe funkcje i metody Matplotlib. Dostosowywanie wyglądu wykresów. Obiektowe podejście w Matplotlib. Rodzaje wykresów w uczeniu maszynowym.

4. **Analiza zbioru Titanic w kontekście uczenia maszynowego.** Pobieranie i eksploracja zbioru Titanic. Sprawdzanie struktury danych i wykrywanie brakujących wartości. Eksploracja danych. Kodowanie zmiennych kategoriycznych. Normalizacja i standaryzacja cech numerycznych. Porównanie metod skalowania. Analiza korelacji.
5. **Klasyfikacja.** Zbiór danych MNIST. Klasyfikacja binarna. Ocena modelu klasyfikacyjnego. Miary wydajności algorytmu. Klasyfikacja wieloetykietowa.
6. **Regresja liniowa.** Wprowadzenie do Scikit-learn. Przygotowanie danych. Budowa i trenowanie modelu regresji liniowej. Ocena jakości modelu.
7. **Maszyny wektorów nośnych.** Klasyfikacja liniowa SVM. Zbiór Iris. Wyjaśnienie wizualizacji granicy decyzyjnej w LinearSVC. Klasyfikacja nieliniowa SVM (jądro RBF). Zbiór Moons. Regresja SVM. GridSearchCV – wyszukiwanie optymalnych parametrów.
8. **Drzewa decyzyjne i lasy losowe.** Drzewa decyzyjne (CART) – klasyfikacja i regresja. Lasy losowe – zespoły drzew decyzyjnych. Hiperparametry. Wizualizacja drzewa i granic decyzyjnych. Drzewa w scikit-learn. Klasyfikacja w drzewach decyzyjnych. Regresja w drzewach decyzyjnych.
9. **Uczenie nienadzorowane.** Algorytm centroidów. Granice algorytmu centroidów. Algorytm DBSCAN.
10. **Uczenie przez wzmacnianie.** Wyszukiwanie strategii. OpenAI Gym. Gradienty strategii. Q-uczenie.
11. **Automatyzacja procesu uczenia się.** AutoML i AutoGluon. Presety w AutoGluon. Klasyfikacja danych tabelarycznych z AutoGluon. AutoML w scikit-learn: Auto-sklearn.
12. **Autokodery i generatywne sieci przeciwstawne.** Efektywne reprezentacje danych. Analiza PCA za pomocą niedopełnionego autokodera liniowego. Autokodery rekurencyjne. Autokodery wariacyjne. Generatywne sieci przeciwstawne: generator i dyskryminator.

Literatura podstawowa:

1. Géron A.: Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Wydanie III, Helion, 2023
2. Szeliga M.: Data science i uczenie maszynowe, PWN, 2017
3. Morzy T.: Eksploracja danych – metody i algorytmy, PWN, 2024

Literatura dodatkowa:

1. Raschka S. i Mirjalili V.: Python. Uczenie maszynowe. Wydanie 2, Helion, 2019
2. Moroney L.: Sztuczna inteligencja i uczenie maszynowe dla programistów. Praktyczny przewodnik po sztucznej inteligencji, Helion, 2021

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Przedstaw metodę klasyfikacji wieloetykietowej”.
W_02	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Opisz dwie techniki doboru właściwej liczby skupień podczas stosowania algorytmu centroidów”.
W_03	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Którego modelu uczącego regresji liniowej należy użyć, jeżeli zbiór danych uczących zawiera miliony cech? Odpowiedź uzasadnij”.

U_01	Efekt realizowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Zainstaluj bibliotekę scikit-learn. Na podstawie zbioru Titanic zbuduj model regresji liniowej, który przewiduje cenę biletu (`fare`) pasażera na podstawie samodzielnie dobranej kombinacji cech.”.
U_02	Efekt realizowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Dobierz klasyfikator rozróżniający dwa gatunki irysów: Iris-versicolor oraz Iris-virginica. W tym celu wybierz dwie cechy opisujące płatki kwiatu. Przygotuj zbiór danych zawierający tylko powyższe dwie klasy.”.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Ocena z przedmiotu składa się z dwóch ocen częściowych: • oceny z zajęć laboratoryjnych, • oceny z egzaminu końcowego. Na ocenę z zajęć laboratoryjnych składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać sumarycznie 50 pkt. Zaliczenie zajęć laboratoryjnych możliwe po uzyskaniu co najmniej 51% liczby punktów z tej formy zaliczenia Egzamin jest egzaminem pisemnym. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na egzaminie można uzyskać maksymalnie 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 51% liczby punktów z tej formy zaliczenia. Ocena końcowa, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS): • 0 – 50 pkt: niedostateczna (F), • 51 – 60 pkt: dostateczna (E), • 61 – 70 pkt: dostateczna plus (D), • 71 – 80 pkt: dobra (C), • 81 – 90 pkt: dobra plus (B), • 91 – 100 pkt: bardzo dobra (A).	
Bilans punktów ECTS:	
Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Przygotowanie się i udział w egzaminie	20 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	33 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta

Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Przygotowanie się i udział w egzaminie	30 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	38 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Software testing
Nazwa w języku angielskim:	Software testing	
Język wykładowy:	angielski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		Pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Marcin Stępnia
Imię i nazwisko prowadzących zajęcia:		dr Marcin Stępnia, mgr Łukasz Marchel
Założenia i cele przedmiotu:		Zakłada się, że student ma wiedzę i umiejętności z przedmiotów: Programowanie obiektowe, Wprowadzenie do technologii WWW, Inżynieria oprogramowania, Platformy programowania. Celem przedmiotu jest zapewnienie studentom znajomości inżynierii oprogramowania i programowania. Przedmiot ma na celu zapoznanie studentów z kluczowymi koncepcjami testowania, różnymi rodzajami testowania, tj. testowaniem funkcjonalnym i нефункциональным, testowaniem integracyjnym i testowaniem systemowym, a także z koncepcjami TDD (Test-Driven Development), modelem ciągłego dostarczania (Continuous Delivery Model) i jego wpływem na testowanie.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	Posiada uporządkowaną i teoretycznie ugruntowaną wiedzę na temat podstawowych koncepcji testowania, typów testowania, metod i celów odpowiednich dla modelu cyklu życia oprogramowania w projekcie.	K_W06
W_02	Posiada uporządkowaną i teoretycznie ugruntowaną wiedzę na temat Test-Driven Development i modelu ciągłego dostarczania oraz ich wpływu na testowanie.	K_W06
	UMIEJĘTNOŚCI Student potrafi:	
U_01	zaplanować i zastosować odpowiedni poziom testowania w kontekście rozwoju oprogramowania. Potrafi projektować konkretne i mierzalne przypadki testowe, aby zapewnić pokrycie i identyfikowalność wymagań.	K_U12, K_U19
U_02	korzystać z technik raportowania problemów, metryk i raportów o stanie testów, aby przekazywać wyniki testów współpracownikom, menedżerom i użytkownikom końcowym.	K_U12, K_U19

U_03	stosować zasady i praktyki TDD (test-driven development) w celu poprawy jakości testów i skrócenia czasu dostarczania oprogramowania. Potrafi integrować procesy testowania w ramach modelu ciągłego dostarczania oprogramowania.	K_U12, K_U19
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	krytycznej oceny swojej wiedzy i jest gotowy uznać znaczenie wiedzy w rozwiązywaniu problemów poznawczych i praktycznych w dziedzinie informatyki.	K_K01
Forma i typy zajęć:		Studia stacjonarne: wykłady (21 hours), laboratoria (24 hours) Studia niestacjonarne: wykłady (15 hours), laboratoria (15 hours)
Wymagania wstępne i dodatkowe:		
Warunkiem uczestnictwa w zajęciach jest wcześniejsze ukończenie następujących modułów: • Programowanie obiektowe • Wprowadzenie do technologii WWW • Inżynieria oprogramowania • Platformy programowania lub znajomość literatury z tych modułów. Dodatkowo wymagana jest znajomość języka angielskiego.		
Treści modułu kształcenia:		
Lectures: 1. Overview of the software testing process. Testing techniques. Functional and non-functional testing. 2. Test management and planning. Testing phases. Testing roles 3. Unit testing – approach and testing techniques, data requirements. Test engineer role, test cases and defect lifecycles 4. Integration testing – approach and testing techniques, data requirements, test analytics 5. System testing and system integration testing. Test process evaluation 6. User acceptance testing. Test plan example. Operations acceptance testing. 7. Regression testing. Software delivery concepts and configuration management. Metrics. Continuous Integration process 8. Agile testing. Test strategy implementation 9. Deployment pipeline. Automated acceptance test suite. Commit stage, commit stage test suite 10. Automated acceptance testing. Testing nonfunctional requirements 11. Test data management. Continuous Delivery management		
Laboratory classes: 1. Basic unit testing 2. Basic integration testing 3. Advanced unit testing 4. Advanced integration testing 5. Decision tables, defect reporting, requirements traceability matrix 6. Postman 7. JMeter 8. TDD 9. Regression testing 10. Code review 11. Selenium 12. Jenkins		
Literatura podstawowa:		

1. Watkins J., Mills S., Testing IT: An Off-the-Shelf Software Testing Process, 2nd edition, Cambridge University Press, 2011
2. Black, R., Agile Testing Foundations, BCS Learning & Development Ltd: Swindon UK, 2017
3. Whittaker J., Arbon J., Carollo J., How Google Tests Software, Addison-Wesley, 2012

Literatura dodatkowa:

1. Roman, A., Testowanie i jakość oprogramowania. Modele, techniki, narzędzia, PWN, Warszawa, 2015
2. Smilgin, R., Zawód tester. Od decyzji do zdobycia doświadczenia, Wydawnictwo Naukowe PWN, Warszawa, 2016
3. Weinberg, G., Perfect Software and Other Illusions about Testing, Dorset House, NY, 2008

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny, wspomagany technikami multimedialnymi. Laboratorium – zajęcia praktyczne z wykorzystaniem języków Java, C#, Python, JavaScript oraz wybranych środowisk programistycznych (np. IntelliJ Idea, Visual Studio, PyCharm, WebStorm lub alternatywnych) z odpowiednimi wtyczkami i rozszerzeniami wspomagającymi proces testowania oraz dedykowanymi narzędziami zewnętrznymi (np. Selenium, Postman, Jenkins, JMeter).

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Weryfikowany podczas kolokwium. Przykładowe pytania: • Porównaj poziomy testów pod względem celów, podstaw testowania, elementów testowych, typowych defektów i awarii, podejść i odpowiedzialności • Omów cechy, podobieństwa i różnice między technikami testowania czarnej skrzynki, białej skrzynki i opartego na doświadczeniu • Wyjaśnij, jak można wyprowadzić przypadki testowe z przypadku użycia
W_02	Weryfikowany podczas kolokwium. Przykładowe pytania Wyjaśnij cykl Test-Driven Development (TDD) i omów, jak wpływa on na jakość oprogramowania. • Porównaj i zestaw tradycyjne metody tworzenia oprogramowania z Test-Driven Development (TDD) pod względem praktyk testowania. • Opisz rolę automatycznego testowania w ciągłym dostarczaniu i wyjaśnij, dlaczego jest ono kluczowe dla tego modelu.
U_01	Weryfikowany na laboratorium. Przykładowe zadanie: „Zaplanuj wykonanie danego zestawu przypadków testowych, biorąc pod uwagę priorytety oraz zależności techniczne i logiczne”
U_02	Weryfikowany na laboratorium. Przykładowe zadanie: Sporządź raport o zawierający informacje o usterkach wykrytych podczas testów”
U_03	Weryfikowany na laboratorium. Przykładowe zadanie: „Opracowanie uwierzytelnianie przy użyciu programowania sterowanego testami dla aplikacji internetowej”
K_01	Weryfikowany na laboratorium w czasie dyskusji.

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem na ocenę. Aby go zaliczyć, wymagane jest zaliczenie części praktycznej i teoretycznej. Za część praktyczną modułu (ćwiczenia laboratoryjne, prace domowe, zadania) można uzyskać maksymalnie 60 punktów, a za część teoretyczną maksymalnie 40 punktów (podczas kolokwium). Każda część zostanie zaliczona po uzyskaniu ponad połowy maksymalnej liczby punktów (odpowiednio 30 i 20 punktów). Ocena końcowa zależy od sumy punktów zdobytych w obu częściach i przedstawia się następująco (w nawiasach podano ocenę ECTS):

- 0 – 50 pkt.: 2 ndst (F),
- 51 – 60 pkt.: 3 dst (E),
- 61 – 70 pkt.: 3,5 dst+ (D),
- 71 – 80 pkt.: 4 db (C),
- 81 – 90 pkt.: 4,5 db+ (B),
- 91 – 100 pkt.: 5 bdb (A).

Poprawa wyniku z części praktycznej: najpóźniej w ciągu 3 tygodni od daty rozpoczęcia zajęć/zadania. Po zakończeniu semestru poprawa nie jest możliwa.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 h
Udział w zajęciach laboratoryjnych	24 h
Samodzielne przygotowanie do zajęć laboratoryjnych, prace domowe i zadania	20 h
Udział w konsultacjach	6 h
Indywidualne studia nad tematami poruszonymi na wykładach	14 h
Przygotowanie do kolokwium	15 h
Całkowity nakład pracy studenta	100 h
Punkty ECTS	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 h
Udział w zajęciach laboratoryjnych	15 h
Samodzielne przygotowanie do zajęć laboratoryjnych, prace domowe i zadania	27 h
Udział w konsultacjach	3 h
Indywidualne studia nad tematami poruszonymi na wykładach	20 h
Przygotowanie do kolokwium	20 h
Całkowity nakład pracy studenta	100 h
Punkty ECTS	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Zintegrowane systemy informatyczne zarządzania
Nazwa w języku angielskim:	Integrated management information systems	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:		szósty
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Marek Pilski
Imię i nazwisko prowadzących zajęcia:		dr Marek Pilski
Założenia i cele przedmiotu:		Zakłada się, że student posiada umiejętność opisywania procesów biznesowych. Celem kursu jest przedstawienie zagadnień związanych z zintegrowanymi systemami zarządzania, min. to: struktura, architektura, standardy oraz nabycie praktycznych umiejętności związanych z wykorzystaniem wybranych ZSiZ.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	pojęcie organizacji, rolę systemu informatycznego w organizacji oraz sposoby (etapy, metody) wprowadzania TI do organizacji.	K_W05
W_02	zagadnienia z zakresu typologii, struktury i standardów typowych systemów informatycznych zarządzania.	K_W05, K_W09
W_03	zagadnienia związane zarządzaniem przedsięwzięć gospodarczych w tym dotyczące zadań, zasobów, optymalizacji oraz narzędzi wspierających.	K_W13
W_04	architekturę zintegrowanych systemów informatycznych zarządzania.	K_W05
	UMIEJĘTNOŚCI Student potrafi:	
U_01	zbudować harmonogram prac uwzględniając jego zadania, czas, budżet, zasoby i zarządzać realizacją przedsięwzięcia wykorzystując do tego dedykowane oprogramowanie.	K_U02, K_U13
U_02	w sposób praktyczny wykorzystać wybrany SiZ do realizacji określonych zadań biznesowych.	K_U02
U_03	przygotować prezentację dotyczącą charakterystyki i właściwości funkcjonalno-użytkowych wybranego SiZ, przedstawić jego wady i zalety oraz przeprowadzić merytoryczną dyskusję.	K_U04
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	wyrażenia oceny zasadności zastosowania SiZ dla danego sektora lub branży.	K_K01
Forma i typy zajęć:		Studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.)

	Studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:	
1. Znajomość procesów biznesowych na ogólnym poziomie w różnych sektorach lub branżach	
Treści modułu kształcenia:	
Treści dotyczące wykładu:	
Zarządzanie przedsięwzięciami gospodarczymi. Projekt, zarządzanie przedsięwzięciem, zadanie, czas trwania zadania, typy zadań i ich właściwości, ścieżka krytyczna i jej właściwości, zadanie krytyczne, zapas czasu, punkty kontrolne, zasoby projektowe, nadmierna alokacja zasobów, bilansowanie, wykres Gantta, harmonogram i jego budowa. Zarządzanie przedsięwzięciami gospodarczymi. Fazy przedsięwzięcia, współzależności pomiędzy zadaniami, właściwości zasobów, przypisanie zasobów do zadań, kalendarze, diagram sieciowy PERT, trójkąt projektu. Rola zasobów informacyjnych w przedsiębiorstwie. System organizacji, zasoby informacyjne w przedsiębiorstwie, procesy informacyjno-decyzyjne, pętla OODA, informacja a procesy podejmowania decyzji. Model zastosowań technologii informatycznej w organizacji. Zastosowanie pojedynczych aplikacji IT, wewnętrzna integracja, reorganizacja procesów biznesowych, reorganizacja sieci gospodarczej, zmiana zakresu działalności gospodarczej. Typologia i ewolucja informatycznych systemów zarządzania. Klasyfikacja systemów informatycznych zarządzania, cechy zintegrowanych systemów informatycznych, standardy systemów informatycznych zarządzania. Struktury i architektura systemów informatycznych zarządzania. Struktury SIŻ: funkcjonalna, informacyjna, organizacyjno-przestrzenna, techniczna. Warstwowa architektura SIŻ. Informacyjna architektura przedsiębiorstwa. Siatka Zachmana. Model TOGAF. Metoda macierzowa BSP. Podejście procesowe w zarządzaniu i jego wpływ na kierunki rozwoju SIŻ. Istota podejścia procesowego w zarządzaniu, proces biznesowy, podejście funkcjonalne w przedsiębiorstwie. Związek podejścia procesowego z koncepcjami współczesnych SIŻ. Złożoność realizacyjna SIŻ i scenariusze realizacji SIŻ. Czynniki występujących podczas realizacji systemu. Składniki systemu jako produktu końcowego. Tworzenie ZSI od podstaw przez służby informatyczne przedsiębiorstwa. Tworzenie ZSI od podstaw przez zewnętrzne firmy informatyczne. Wybór, zakup i wdrożenie wyrobu gotowego. Architektura zintegrowanego systemu zarządzania. Charakterystyka wybranych podsystemów: środki trwałe, logistyka, zarządzanie produkcją, zasoby ludzkie. Architektura zintegrowanego systemu zarządzania. Charakterystyka wybranych podsystemów: finanse i księgowość, controlling, CRM, BI.	
Treści dotyczące laboratorium:	
MS Project. Wprowadzenie podstawowych danych o projekcie, definiowanie zadań, szacowanie czasu trwania zadań, edycja zadań. MS Project. Definiowanie zależności pomiędzy zadaniami, tworzenie harmonogramu prac, definiowanie zadań sumarycznych i punktów kontrolnych, ustalanie ścieżki krytycznej, dodanie zadania cyklicznego. MS Project. Definiowanie zasobów przedsięwzięcia, definiowanie dni wolnych od pracy, zmiana czasu pracy, definiowanie nowego kalendarza, przydzielanie zasobów do zadań, definiowanie specyficznych stawek zasobów.	

4. **MS Project.** Optymalizacja projektu. Lokalizowanie zasobów z nadmierną alokacją, bilansowanie zasobów, śledzenie realizacji projektu, budowa wariantów planu, aktualizacja postępu prac, symulacja postępu na dany dzień, generowanie diagramu sieciowego PERT.
5. **MS Project.** Prace nad uzupełnieniem i rozbudową projektu o nowe zadania i zasoby.
6. **MS Project.** Opracowanie harmonogramu dla wybranego przedsięwzięcia. Sprawdzian.
7. **ERP Optima.** Prace konfiguracyjne. Założenie nowej Bazy Danych dla nowej firmy, konfiguracja Bazy Danych, dodanie operatora, uzupełnienie danych słownikowych.
8. **ERP Optima.** Moduł Kasa-Bank. Definiowanie rejestrów kasowo-bankowych, planowanie płatności, definiowanie zapisów kasowo-bankowych, rozliczanie zapisu z planowaną płatnością, kompensacja zapisów kasowych.
9. **ERP Optima.** Moduł Handel. Konfiguracja modułu, dodanie karty ewidencyjnej nowego towaru, wprowadzenie Bilansu Otwarcia, ustalanie warunków sprzedaży dla kontrahenta, wystawianie faktur sprzedaży dla kontrahentów o różnym statusie, sprzedaż ratalna.
10. **ERP Optima.** Realizacja procesu biznesowego związanego z przyjęciem i sprzedażą towaru. Sprawdzian.
11. **Przegląd wybranych informatycznych systemów zarządzania.** Prezentacje studenckie wybranego systemu informatycznego zarządzania z wybranego sektora lub branży.
12. **Przegląd wybranych informatycznych systemów zarządzania.** Prezentacje studenckie wybranego systemu informatycznego zarządzania z wybranego sektora lub branży.

Literatura podstawowa:

1. Andrzej Bytniewski, Anna Chojnacka-Komorowska, Marianna Kowalska, Marcin Hernes, Kamal Matouk. *Architektura zintegrowanego systemu zarządzania*. Wydawnictwo Uniwersytetu Ekonomicznego. Wrocław. 2015.
2. Jerzy Kisielnicki. *Systemy informatyczne zarządzania*. Agencja Wydawniczo-Poligraficzna Placet. Warszawa. 2013.

Literatura dodatkowa:

1. Włodzimierz Jerzy Jabłoński, Witold Bartkiewicz. *Systemy informatyczne zarządzania: klasyfikacja i charakterystyka systemów*. Wydawnictwo KPSW. Bydgoszcz. 2006.
2. Mirosława Kopertowska-Tomczak, Witold Sikorski. *MS Project : kurs podstawowy*. Wydawnictwo Naukowe PWN. 2007.

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Efekt realizowany na egzaminie formie pisemnej lub testu wiedzy. Przykładowe pytanie: Zdefiniuj pojęcie organizacji i scharakteryzuj rolę zasobów informacyjnych w procesie podejmowania decyzji.
W_02	Efekt realizowany na egzaminie w formie pisemnej lub testu wiedzy. Przykładowe pytanie: Scharakteryzuj system standardu ERP, wskaż obszary w których ma zastosowanie i podaj przykład takiego systemu.
W_03	Efekt realizowany na egzaminie w formie pisemnej lub testu wiedzy. Przykładowe pytanie: Co to jest harmonogram Gantta, do czego służy i z czego się składa.
W_04	Efekt realizowany na egzaminie w formie pisemnej lub testu wiedzy. Przykładowe pytanie: Scharakteryzuj zintegrowane systemy informatyczne zarządzania.

U_01	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji oraz realizacja mini projektu w formie sprawdzianu. Przykładowe zadanie: w MS Project zdefiniuj relacje typu Rozpoczęcie-Rozpoczęcie pomiędzy wskazanymi zadaniami. Opracuj harmonogram prac związany z napisaniem swojej pracy inżynierskiej, zidentyfikuj zadania, zasoby oraz czas pracy.
U_02	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie jakość i kompletność wykonanych zadań z instrukcji oraz sprawdzianu umiejętności wykorzystania SIŁ. Przykładowe zadanie: zrealizuj Bilans Otwarcia dla wskazanych grup asortymentowych towaru. Sprawdzian: w poznanym SIŁ zrealizuj proces przyjęcia towaru i jego sprzedaży kontrahentowi wystawiając odpowiednie dokumenty.
U_03	Efekt realizowany w trakcie zajęć laboratoryjnych. Oceniana będzie kompletność zagadnień zaprezentowanych na prezentacji. Przykładowe wymagania: Prezentacja wybranego systemu informatycznego zarządzania powinna zawierać: 1. Nazwę oprogramowania i producenta, 2. Informację mówiącą do kogo jest kierowane oprogramowanie, kto jest klientem, jaki sektor, itp., 3. Model licencji, cena, 4. Omówienie możliwości systemu 5. Zaprezentowanie na "screenach" ścieżki realizacji reprezentatywnego procesu biznesowego 6. Podsumowanie, np. dostrzeżone wady/zalety.
K_01	Efekt realizowany w trakcie zajęć laboratoryjnych podczas prezentacji prezentowanego SIŁ i dyskusji.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Ocena z przedmiotu składa się z dwóch ocen cząstkowych: • realizacja zadań z MS Project – 5 pkt., • sprawdzian z MS Project – 10 pkt • realizacja zadań z ERP Optima – 5 pkt., • sprawdzian z ERP Optima – 10 pkt • prezentacja wybranego systemu zarządzania – 10 pkt., • kolokwium/test wiedzy z treści wykładowych – 10 pkt. Na ocenę końcową składają się suma punktów wyszczególnionych powyżej oraz 50 punktów za egzamin. Sumarycznie można zdobyć 100 punktów. Ocena końcowa, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS): • 0 – 50 pkt: niedostateczna (F), • 51 – 60 pkt: dostateczna (E), • 61 – 70 pkt: dostateczna plus (D), • 71 – 80 pkt: dobra (C), • 81 – 90 pkt: dobra plus (B), • 91 – 100 pkt: bardzo dobra (A).	
Bilans punktów ECTS:	
Studia stacjonarne	
Aktywność	Obciążenie studenta

Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	22 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Indywidualne studiowanie i przygotowanie się do egzaminu	28 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Indywidualne studiowanie i przygotowanie się do egzaminu	35 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Rozpoznawanie obrazów
Nazwa w języku angielskim:	Image recognition	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	piąty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Andrzej Salamończyk
Imię i nazwisko prowadzących zajęcia:		dr Andrzej Salamończyk mgr inż. Monika Wereszczyńska
Założenia i cele przedmiotu:		Zakłada się, że student zna podstawy analizy matematycznej, algebry oraz podstaw programowania.. Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z algorytmami i metodami rozpoznawania obrazów oraz zdobycie praktycznych umiejętności w implementacji tych algorytmów i metod oraz korzystania z odpowiednich bibliotek.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	rolę etapów w zadaniu rozpoznawania obrazów: preprocesing, segmentacja, detekcja i ekstrakcja cech, klasyfikacja	K_W08
W_02	zagadnienia związane z metodami i algorytmami rozpoznawania obrazów	K_W08
	UMIEJĘTNOŚCI Student potrafi:	
U_01	pozyskiwać informacje dotyczące z literatury, baz danych, dokumentacji i innych źródeł, potrafi stosować pozyskane informacje do zadania rozpoznawania obrazów	K_U01
U_02	wykorzystać właściwie dobrane środowiska programistyczne i narzędzia komputerowe do projektowania i weryfikacji systemów informatycznych związanych z przetwarzaniem obrazów	K_U11
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu rozpoznawania obrazów	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		

Znajomość analizy matematycznej, algebry oraz podstaw programowania.

Treści modułu kształcenia:

Wykład

1. Wstęp. Cyfrowa reprezentacja obrazu. Etapy zadania rozpoznawania obrazów: preprocessing, segmentacja, ekstrakcja cech, klasyfikacja, interpretacja. Kryteria oceny i zastosowania systemów rozpoznających obrazy.
2. Formaty plików graficznych (obrazów). Modele barwowe. Podstawy języka Python.
3. Przydatne biblioteki w rozpoznawaniu obrazów: Python: NumPy OpenCV, Scikit-Learn, Tensorflow.
4. Preprocesing. Metody aproksymacyjne. Przetwarzanie obrazu z przykładami w języku Python.
5. Segmentacja. Metody punktowe: progowanie i klasteryzacja, metody krawędziowe, metody obszarowe: rozrost/podział/łączenie obszarów, metody hybrydowe i wododziałowe.
6. Detekcja i ekstrakcja cech. Deskryptory kształtu, HOG, SIFT, BRIEF, FAST. Kody łańcuchowe.
7. Segmentacja i klasyfikacja tekstur. Deskryptory tekstury. Metody k-NN.
8. Rozpoznawanie wzorców. Metoda PCA. Maszyna wektorów nośnych SVM. Metody statystyczne, klasyfikacja bayesowska.
9. Sieci neuronowe i głębokie sieci neuronowe. Wykorzystanie sieci neuronowych i konwolucyjnych sieci neuronowych w rozpoznawaniu obrazów. Wykorzystanie metod i narzędzi sztucznej inteligencji.
10. Analiza sekwencji obrazów. Detekcja ruchu i śledzenie obiektów.
11. Tendencje w rozpoznawaniu obrazów. Kolokwium.

Laboratorium

1. Przetwarzanie obrazów w języku Python
2. Przetwarzanie obrazów, NumPy (preprocessing i segmentacja obrazów)
3. Przetwarzanie obrazów OpenCV (preprocessing i segmentacja obrazów)
4. Przetwarzanie obrazów wideo w języku Python
5. Deskryptory SIFT, szukanie odpowiadających sobie punktów na obrazach
6. Klasyfikacja z wykorzystaniem sztucznych sieci neuronowych
7. Keras, model uczenia maszynowego
8. Wykorzystanie konwolucyjnych sieci neuronowych do rozpoznawania obrazów
9. Biblioteka Tesseract, rozpoznawanie tekstu
10. Wykrywanie twarzy na obrazach
11. Wykrywanie linii (np. pasa ruchu na drodze)
12. Przegląd innych ciekawych bibliotek i narzędzi w rozpoznawaniu obrazów..

Literatura podstawowa:

1. R.Tadeusiewicz, M.Flasiński, Rozpoznawanie obrazów, WNT, 1991.
2. Yu-Jin Zhang. 2D Computer Vision: Principles, Algorithms and Applications. World Scientific Pub 2022
3. K. Stąpor, Automatyczna klasyfikacja obrazów, EXIT, 2005

Literatura dodatkowa:

1. Sandipan Dey. Image Processing and Computer Vision Masterclass with Python - 2nd Edition. BPB Publications 2025
2. Kaehler, G.Bradski, OpenCV 3: Komputerowe rozpoznawanie obrazów w C++ przy użyciu biblioteki OpenCV, Helion 2017.
3. M. Domański, Obraz cyfrowy, WKŁ 2009.

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Efekt weryfikowany na egzaminie. Przykładowe pytanie: Jakie są główne etapy rozpoznawania obrazów i jaka jest rola każdego z nich?
W_02	Efekt weryfikowany na egzaminie. Przykładowe pytanie: Które z poniższych stwierdzeń dotyczą algorytmu SIFT (Scale-Invariant Feature Transform) są prawdziwe? (Wybierz wszystkie poprawne odpowiedzi) A) SIFT wykrywa i opisuje punkty kluczowe, które są odporne na skalowanie i rotację obrazu. B) Algorytm SIFT opiera się na transformacji Fouriera obrazu. C) SIFT wykorzystuje do detekcji punktów kluczowych różnicę Gaussowską (DoG). D) Metoda SIFT jest wykorzystywana głównie do segmentacji obrazów. E) Dopasowanie cech w SIFT odbywa się na podstawie podobieństwa deskryptorów.
U_01	Efekt weryfikowany na laboratorium. Przykładowe zadanie: Na podstawie dokumentacji OpenCV i przykładu napisz program do śledzenia obiektów.
U_02	Efekt weryfikowany na laboratorium. Przykładowe zadanie: Zaprojektuj prosty system do detekcji i klasyfikacji obiektów na obrazach (np. wykrywanie twarzy lub klasyfikacja zwierząt). Zaimplementuj system. Przeprowadź testy działania systemu na zestawie testowym obrazów, oceniając jego skuteczność (np. dokładność klasyfikacji lub wskaźnik trafień detekcji)
K_01	Obserwacja postaw studenta w czasie zadań laboratoryjnych.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i egzaminu. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na laboratoriach (oprócz pierwszego i ostatniego).

- Oceniane laboratoria (10 zajęć po 5 pkt.)

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

$$P=L+E,$$

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

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

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

- 81-90 punktów – 4,5
- 91-100 punktów – 100

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	22 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	28 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	30 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	35 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Programowanie zaawansowane
Nazwa w języku angielskim:	Advanced programming	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski
Imię i nazwisko prowadzących zajęcia:		mgr Łukasz Marchel
Założenia i cele przedmiotu:		Zakłada się, że student zna podstawy programowania obiektowego oraz język C++ lub Java. Celem przedmiotu jest zapoznanie studentów z projektowaniem i implementacją aplikacji wykorzystujących zaawansowane konstrukcje i mechanizmy programistyczne dostępne w języku C# na platformie .NET. Studenci będą dokładnie rozumieć znaczenie i sposób działania takich konstrukcji i mechanizmów jak obsługa wyjątków, polimorfizm, budowanie bazy wspólnego kodu, delegaty, refleksja, serializacja, LINQ i innych, oraz będą potrafili prawidłowo je wykorzystywać w procesie tworzenia aplikacji różnego typu.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	zaawansowane mechanizmy programistyczne, w tym: strukturalną obsługę wyjątków, interfejsy; generyczność, refleksję i serializację.	W_S01
W_02	elementy składające się na platformę .NET, język programowania C# i jego zaawansowane konstrukcje w tym: właściwości automatyczne, metody rozszerzeniowe, niejawne typowanie, delegaty, inicjalizatory obiektów.	W_S01
W_03	różne sposoby utrwalania danych poprzez bazy danych, pliki XML, oraz sposoby korzystania z tych źródeł danych (ADO.NET i Entity Framework).	W_S01
	UMIEJĘTNOŚCI Student potrafi:	
U_01	korzystać z środowiska programistycznego Visual Studio i implementować w nim podstawowe rodzaje projektów (biblioteki klas, aplikacje konsolowe, desktopowe i sieciowe).	U_S01

U_02	korzystać z zaawansowanych mechanizmów i konstrukcji (obsługi wyjątków, interfejsów, generyczności, delegatów, LINQ, refleksji i serializacji).	U_S01
U_03	zaprojektować, zaimplementować oraz przetestować aplikacje sieciowe z dostępem do baz danych i wykorzystaniem wzorca MVC.	U_S06
KOMPETENCJE SPOŁECZNE Student jest gotów do:		
K_01	podejmowania decyzji i przeprowadzania analizy efektów tych decyzji w ramach rozwiązywania zadań programistycznych	K_S01
K_02	uznania znaczenia wiedzy w rozwiązywaniu problemów programistycznych oraz do konstruktywnej krytyki powstałych rozwiązań	K_S02
Forma i typy zajęć:		studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (45 godz.) studia niestacjonarne: wykłady (18 godz.), ćwiczenia laboratoryjne (24 godz.)
Wymagania wstępne i dodatkowe:		
Zakłada się, że student zna podstawy programowania obiektowego oraz język C++ lub Java.		
Treści modułu kształcenia:		
Wykłady: 1. Platforma .NET. Omówienie wspólnego silnika uruchomieniowego, wspólnego systemu typów, wspólnej specyfikacji języka, wielojęzyczności i niezależności od platformy. 2. Obiektowe programowanie w języku C#. Znaczenie filarów programowania obiektowego (obiekty, klasy, dziedziczenie, hermetyzacja, abstrakcja i polimorfizm) z naciskiem na elementy składni charakterystycznej dla języka C#. 3. Strukturalna obsługa wyjątków. Znaczenie mechanizmu wyjątków. Sposoby implementacji strukturalnej obsługi wyjątków w języku C#. 4. Interfejsy. Znaczenie interfejsu jako warstwy abstrakcji, definiowanie własnych interfejsów, implementowanie i korzystanie z bazy wspólnego kodu, różnice między interfejsami a klasami abstrakcyjnymi. 5. Generyczność. Znaczenie typów i metod generycznych, sposoby definiowania i wykorzystywania własnych szablonów, definiowanie ograniczeń dla parametrów generycznych. 6. Delegaty. Wykorzystanie delegatów do komunikacji między obiektami, silnie typowane sposoby przekazywania wskaźników do metod, metody anonimowe, wyrażenia lambda. 7. Zaawansowane konstrukcje języka C#. Niejawne typowanie zmiennych lokalnych, właściwości automatyczne, metody rozszerzeniowe, metody częściowe, inicjalizator obiektów i typy anonimowe. 8. Technologia LINQ. Język zapytań zintegrowany z językiem programowania jako ujednolicony sposób wyszukiwania danych niezależnie od ich źródła. Operatory zapytań i ich wewnętrzna reprezentacja. 9. ADO.NET – warstwa połączeniowa. Architektura dostawców danych, wspólne interfejsy i klasy bazowe, korzystanie z obiektów połączenia, polecenia i odczytywania danych. 10. ADO.NET – warstwa bezpołączeniowa. Rola warstwy bezpołączeniowej, typy obiektów wykorzystywane do lokalnego reprezentowania bazy danych, sposoby kontroli zmian w lokalnej wersji. 11. Entity Framework. Nowoczesne narzędzie ORM do mapowania obiektowo-relacyjnego, upraszczające dostęp do baz danych, umożliwiające pracę z danymi jako obiektami .NET. Podejście Code First, obiekty encji i związane z nimi atrybuty. 12. LINQ to ADO.NET i LINQ to Entity. Wykorzystanie technologii LINQ do wyszukiwania danych w bazie danych lub jej lokalnej reprezentacji. Automatyczne generowanie klas reprezentujących tabele i klas wspomagających. 13. Refleksja. Znaczenie metadanych, klasy i metody związane z mechanizmem refleksji; dynamiczne ładowanie pakietów, późne wiązanie, definiowanie własnych atrybutów. 14. Serializacja. Znaczenie serializacji, dostępne formaty (binarny, SOAP, XML), konfigurowanie procesu serializacji. 15. ASP.NET. Tworzenie nowoczesnych aplikacji webowych. Budowanie dynamicznych stron internetowych, API oraz aplikacji w architekturze MVC.		
Laboratoria:		

1. **Podstawy programowania .NET.** Implementacja aplikacji konsolowej wykorzystującej dziedziczenie i polimorfizm w języku C#.
2. **Obsługa sytuacji wyjątkowych.** Definiowanie i wyrzucanie własnych wyjątków. Wykorzystanie polimorficznego działania hierarchii klas z właściwościami automatycznymi.
3. **Interfejsy.** Definiowanie kontraktów (zestawy metod, właściwości, zdarzeń), które muszą zostać zaimplementowane przez klasy. Rozdzielanie tematów projektów indywidualnych.
4. **Generyczność.** Wykorzystanie istniejących typów generycznych (szablonów), tworzenie własnych typów i metod generycznych.
5. **Delegaty.** Tworzenie zmiennych, które przechowują metody do późniejszego wykonania, przekazywanie metod jako argumenty innych metod, i wykonywanie ich przy pomocy zdarzeń.
6. **Zaawansowane konstrukcje języka C#.** Właściwości automatyczne, klasy abstrakcyjne, porównywanie i sortowanie obiektów, inicjalizatory obiektów. Tworzenie aplikacji desktopowej w Window Forms.
7. **Zapytania LINQ.** Wykorzystanie zapytań zintegrowanych z językiem C#.
8. **Solucje.** Tworzenie rozwiązania zbudowanego z kilku projektów.
9. **Solucje c.d.** Wskazywanie zależności między projektami. Powstanie m.in. wspólna biblioteka (.dll), która będzie wykorzystana przez pozostałe projekty.
10. **Aplikacja webowa ASP.NET Core MVC.** Zapoznanie się ze sposobami tworzenia aplikacji webowych z wykorzystaniem języka C#.
11. **ADO.NET.** Wykorzystanie ADO.NET (warstwa połączeniowa) w aplikacjach webowych.
12. **Entity Framework.** Podejście Code First do tworzenia bazy danych. Wykorzystanie Linq to Entity.
13. **Refleksja i serializacja.** Tworzenie „punkt wejścia” dla wtyczek, z wykorzystaniem mechanizmu refleksji. Poprzez wtyczki (pakiety wrzucane do ustalonego katalogu) będzie można dodawać funkcjonalność serializacji do nowych formatów. Podstawowa aplikacja będzie obsługiwać zapisywanie danych w postaci dokumentu XML.
14. **Praca z dokumentami XML.** Wykorzystanie LINQ to XML do tworzenia i manipulowania dokumentami XML.
15. **Obrona projektów indywidualnych.** Testowanie i weryfikacja poprawności realizacji projektu indywidualnego. Możliwe niewielkie modyfikacje oprogramowania, w celu weryfikacji samodzielności jego wykonania.

Literatura podstawowa:

1. Mark Michaelis, C# 8.0. Kompletny przewodnik dla praktyków. Wydanie VII, Helion 2021
2. Adam Freeman, ASP.NET Core 3. Zaawansowane programowanie. Wydanie VIII, Helion 2021

Literatura dodatkowa:

1. Joseph Albahari, C# 9.0 w pigułce, Helion 2021

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia wspomagane sprzętem laboratoryjnym. Zamieszczanie na stronie internetowej zadań ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Sprawdzone podczas egzaminu w formie pisemnej Przykładowe pytania: 1. Czym jest interfejs? Podaj przykład jego użycia, 2. Na czym polega mechanizm refleksji? Podaj przykłady jego zastosowania.
W_02	Sprawdzone podczas egzaminu w formie pisemnej Przykładowe pytania:

	1. Czym są delegaty i do czego ich używamy? 2. Wymień 5 konstrukcji językowych dostępnych w C# a nie dostępnych w Java.
W_03	Sprawdzone podczas egzaminu w formie pisemnej Przykładowe pytania: 1. Scharakteryzuj wzorce O/RM używane do ładowania powiązanych danych w Entity Framework 2. Wymień i scharakteryzuj komponenty <u>ADO.NET</u> potrzebne do implementacji warstwy bezpołączeniowej.
U_01	Systematycznie sprawdzane na zajęciach. Wszystkie zagadnienia związane z danym laboratorium udostępnione są co najmniej tydzień wcześniej. Studenci powinni przygotować się do laboratoriów znając ich treści. Przykładowe zadanie: 1. W Visual Studio utwórz rozwiązanie, a następnie dodaj do niego 3 typy projektów (aplikacja sieciowa, aplikacja desktopowa, biblioteka dll). Bibliotek dll ma być współdzielona pomiędzy pozostałymi projektami i powinna zwracać przykładowy komunikat.
U_02	Systematycznie sprawdzane na zajęciach. Przykładowe zadania: 1. Dana jest lista osób o strukturze z poprzedniego zadania zapisana w liście generycznej. Napisz odpowiednie zapytania LINQ umożliwiające wyszukanie podanych niżej informacji. 2. Zaprojektuj aplikację, którą można będzie rozszerzać za pomocą wtyczek. Opracuj odpowiedni interfejs i mechanizm dynamicznego ładowania pakietów z wtyczkami.
U_03	Systematycznie sprawdzane na zajęciach oraz podczas obrony projektu indywidualnego. Przykładowe zadania: 1. Wykorzystując wzorzec MVC zaimplementuj CRUD dla encji Pojazd. Zaproponuj właściwości tej encji. Wykorzystaj EntityFramework do utrwalania danych w bazie.
K_01	Weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, podczas zaliczania zadania indywidualnego, a także będą sprawdzane na egzaminie.
K_02	Weryfikowany, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych, podczas zaliczania zadania indywidualnego, a także będą sprawdzane na egzaminie.
Forma i warunki zaliczenia:	

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

- Regularne zajęcia – 30 pkt.,
- Zadanie indywidualne – 20 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 16 pkt., obrona indywidualnego zadania – co najmniej 11 pkt. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt. Ponadto, każde laboratorium z osobna należy zaliczyć na co najmniej 50% punktów.

Egzamin jest egzaminem pisemnym lub ustnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 25 pkt.

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

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

Poprawy:

- Poprawa dwóch ćwiczeń w trakcie zajęć w semestrze.

Poprawa zadania indywidualnego w sesji egzaminacyjnej przed drugim terminem egzaminu.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	45 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych i wykonanie zadania indywidualnego	10 godz.
Praca nad projektem indywidualnym	8 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Przygotowanie się do egzaminu	5 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	25 godz.
Praca nad projektem indywidualnym	15 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	14 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Aplikacje internetowe i rozproszone
Nazwa w języku angielskim:		Web and distributed applications
Język wykładowy:	Polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	Trzeci	
Semestr:	Szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Dariusz Mikułowski
Imię i nazwisko prowadzących ćwiczenia:		dr Dariusz Mikułowski
Założenia i cele przedmiotu:		Założeniem kursu jest znajomość przez studentów podstawowych protokołów i technologii Internetowych oraz umiejętność programowania w językach Java, Javascript i znajomość technologii webowych. Celem kursu jest zapoznanie studentów z technologiami i narzędziami pozwalającymi na implementację aplikacji internetowych wykorzystujących różne protokoły architektury i standardy przesyłania informacji m.in. Działanie poczty internetowej, konstrukcja i działanie pająków internetowych i indeksatorów, operacje na dokumentach XML, działanie aplikacji P2P, działanie mobilnych agentów.
Symbol efektu	Efekty uczenia się: WIEDZA Student zna i rozumie:	Symbol efektu kierunkowego
W_01	Funkcjonowanie języków opisu dokumentu HTML i XML i ich zastosowania. zasady działania języków opisu struktury dokumentu DTD oraz XMLSchema	W_S03
W_02	Protokoły komunikacyjne używane w aplikacjach internetowych; zna zasady działania aplikacji rozproszonych takich jak poczta elektroniczna, wyszukiwarki internetowe, pająki i roboty.	W_S03
W_03	Zna i rozumie działanie języków zapytań do dokumentów XML: XPath oraz XQuery.	W_S03
W_04	koncepty architektur otwartych systemów internetowych: WebService, WS-REST, technologii semantycznych oraz P2P. oraz ich zastosowania.	W_S03
Symbol efektu	Efekty uczenia się: UMIEJĘTNOŚCI Student potrafi:	Symbol efektu kierunkowego
U_01	Pisać programy posługujące się pocztą internetową jako medium komunikacji	U_S02

U_02	Wykorzystać w komunikacji między aplikacjami języki i formaty takie jak JSON, XML oraz HTML i specyfikować język komunikacji w DTD i XMLSchema.	U_S02
U_03	Napisać aplikację wyszukiwarki Internetowej oraz prostego pająka Internetowego.	U_S02
U_04	Posługiwać się narzędziami do wyszukiwania informacji w dokumentach XML/HTML oraz do ich przetwarzania z wykorzystaniem parserów XML DOM i JAXP,	U_S02
Forma i typy zajęć:		studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (45 godz.) studia niestacjonarne: wykłady (18 godz.), ćwiczenia laboratoryjne (24 godz.)
Wymagania wstępne i dodatkowe:		
Warunkiem uczestnictwa w kursie jest posiadanie umiejętności programowania w języku Java i znajomość podstawowych technologii do tworzenia aplikacji webowych.		
Treści modułu kształcenia:		
Treści dotyczące wykładu: Architektury dla aplikacji internetowych: wybrane protokoły internetowe, poczta elektroniczna, WWW. Język XML (eXtensible Markup Language) jako język wymiany informacji między aplikacjami: XML jako szczególny przypadek SGML, budowa dokumentu, przestrzeń nazw, zastosowania XML Język XML (eXtensible Markup Language) - interfejsy programistyczne: model DOM, model SAX, realizacje w Javie Metainformacje dla XML - słaba kontrola składni: DTD, XSchema Język ścieżek XPATH oraz język transformacji dokumentów XML (XSLT): XPath, Koncepcja transformacji XSL (XML Stylesheet Language);, Budowa wzorców XSLT;; Środki programistyczne dostępne w XSLT. repozytoria XML i języki zapytań: repozytoria XML, XQUERY, inne języki zapytań dla XML Problemy technologiczne wyszukiwarek internetowych: Budowa wyszukiwarki, Technologie pająków, Technologie Indeksatorów, Metody wyszukiwania Otwarte architektury aplikacji: SOAP i Web-Serwisy: Oprogramowanie pośredniczące, WSDL, ebXML Technologie do modelowania, tworzenia i wykonywania procesów biznesowych: BPL4WS, BPMN, Systemy transakcyjne zgodne z paradygmatem ACID. Otwarte architektury aplikacji: REST: Wady koncepcji Webservisów, Naturalność architektury REST Otwarte architektury aplikacji: Modele "koleżeńskie": architektura P2P, wymagania technologiczne, JXTA, zastosowania w biznesie Technologie semantyczne i pojęcie sieci semantycznej (Semantic Web), pojęcie ontologii, języki do wyrażania ontologii (RDF, OWL), zastosowania technologii semantycznych do budowania aplikacji Internetowych, - Programowanie w chmurze: środki programistyczne, Koncepcja mikroservisów, kontrola zasobów, aspekty ekonomiczne. Mobilni agenci jako przykład aplikacji rozproszonej. Inne zastosowania XML (MathML, SVG, EPUB3, MusicML) i Elementy projektowania uniwersalnego w aplikacjach Internetowych. Treści dotyczące zajęć laboratoryjnych: - Implementacja aplikacji www i jej połączenie z bazą danych: Instalacja, ustawienia i deinstalacja serwera WWW: umieszczanie na serwerze pliku html, serwetu. Instalacja bazy danych, np. Derby. - Tworzenie aplikacji korzystającej z serwera pocztowego: Uruchomienie serwera pocztowego James, utworzenie mailtutwysyłającego i odbierającego wiadomości email. - Operacje na dokumentach XML: budowa plików XML, metody parsowania dokumentu, modyfikowanie i zapisywanie z poziomu języka Java. - Mapowanie modelu klas do schematów XSD: Opisywanie modelu klas w języku Java za pomocą adnotacji XML według specyfikacji JAXB. Generowanie pliku XSD z klas Javy. Zapisywanie obiektów Javy do pliku XML, Wczytywanie dokumentów XML do obiektów Javy. - Tworzenie ścieżek XPath i transformacja XSLT dla plików XML: Ustawianie stylów dla plików XML, Tworzenie ścieżek XPath, Tworzenie plików transformacji, implementacja transformacji z poziomu Java.		

6. Zapytania XQuery: Zastosowanie zapytań XQuery do odpytywania bazy danych w pliku XML, korzystanie z zapytań XQuery z poziomu kodu Java.
7. Wyszukiwarki – roboty internetowe: Tworzenie robota internetowego szukającego informacji w sieci w oparciu o zadane zapytanie.
8. Własna wyszukiwarka: Implementacja Indeksera i wyszukiwarki w oparciu o Apache Lucene.
9. Serwery proxy: Implementacja własnego serwera proxy z możliwością modyfikacji kodu html i obrazków.
10. Usługi sieciowe korzystające z WSDL i SOAP: Tworzenie usług sieciowych z wykorzystaniem standardu JAX-WS w Springu i utworzenie usługi i jej konsumenta.
11. Usługi webowe REST i rejestr usług Eureka: Tworzenie dwóch różnych usług Rest w technologiach Java i w Dotnet, utworzenie rejestru usług Spring Cloud Eureka i zarejestrowanie usług w rejestrze, utworzenie klientów dla usług.
12. Komunikacja według modelu peer-to-peer: Implementacja systemu wymiany plików, działającego w oparciu o model P2P z centralnym serwerem i komunikacją w XML.
13. Mobilne agenty: Implementacja agentów z wykorzystaniem biblioteki JADE.
14. Ontologie w języku OWL: Utworzenie własnej ontologii i zapisanie jej w formacie OWL, Manipulowanie strukturą ontologii z poziomu kodu w Java.

Literatura podstawowa:

1. Mikrouslugi w javie poradnik eksperta. Sourabh Sharma. Helion Gliwice 2018
2. REBECCA RIORDAN. HEAD FIRST AJAX EDYCJA POLSKA. HELION 2019
3. Priscilla Walmsley. Xquery: Search Across a Variety of XML Data. O'Reilly Media 2016

Literatura dodatkowa:

1. B. Eckel: Thinking in Java Prentice-Hall, December 2002
2. Dariusz Mikułowski, XML w programowaniu aplikacji internetowych. Akademia Podlaska 2009.
3. XML : księga eksperta / Elliott Rusty Harold ; [tł. Tomasz Żmijewski] Grupa Wydawnicza Helion. 2000

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektu uczenia się
W_01	Efekt będzie realizowany na egzaminie w formie testu online. Przykładowe pytanie: Podaj co najmniej 3 główne różnice pomiędzy dokumentami HTML a XML.
W_02	Efekt będzie realizowany na egzaminie w formie testu online. Przykładowe pytanie: Wymień kroki pracy pająka internetowego.
W_03	Efekt będzie realizowany na egzaminie w formie testu online. Przykładowe pytanie: Wymień co najmniej 3 najczęściej używane konstrukcje języka XQuery.
W_04	Efekt będzie realizowany na egzaminie w formie testu online. Przykładowe pytanie: Opisz 1 zdaniem podstawowe różnice pomiędzy usługami webowymi implementowanymi w architekturze SOAP+WSDL i REST.
U_01	Efekt będzie realizowany podczas zajęć laboratoryjnych poprzez wykonywanie zadań programistycznych przez studenta i ich obronę. Przykładowe zadanie: Napisz program w formie servletu który pozwoli na odebranie listy wiadomości z serwera email i zaprezentowanie ich użytkownikowi.

U_02	Efekt będzie realizowany podczas zajęć laboratoryjnych poprzez wykonywanie zadań programistycznych przez studenta i ich obronę. Przykładowe zadanie: Napisz program który wczyta dokument XML i zweryfikuje jego poprawność zgodnie z podanym schematem XSD.
U_03	Efekt będzie realizowany podczas zajęć laboratoryjnych poprzez wykonywanie zadań programistycznych przez studenta i ich obronę. Przykładowe zadanie: Napisz program robota internetowego który wyszuka w 5 podanych stronach wszystkie grafiki i wyświetli ich adresy oraz opisy alternatywne.
U_04	Efekt będzie realizowany podczas zajęć laboratoryjnych poprzez wykonywanie zadań programistycznych przez studenta i ich obronę. Przykładowe zadanie: Napisz program który za pomocą języka XQuery wyszuka w podanym dokumencie XML informacje o autorach i tytułach książek z dziedziny informatyka.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 140 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu co najmniej 71 pkt.

Egzamin jest egzaminem pisemnym w formie testu online z pytaniami zamkniętymi i otwartymi. Można na nim uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych),

- 0 - 100 pkt: niedostateczna (F)
- 101 – 120 pkt: dostateczna (E),
- 121 – 140 pkt: dostateczna plus (D),
- 141 – 160 pkt: dobra (C),
- 161 – 180 pkt: dobra plus (B),
- 181 – 200 pkt: bardzo dobra (A).

Poprawy: Jednorazowa poprawa egzaminu w sesji. Jednorazowe poprawy do 2 ćwiczeń laboratoryjnych w trakcie semestru.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	45 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	6 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	37 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	17 godz.
Obecność na egzaminie	1 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Systemy baz danych
Nazwa w języku angielskim:	Database systems	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:		szósty
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Piotr Świtalski
Imię i nazwisko prowadzących zajęcia:		dr Piotr Świtalski, mgr Łukasz Marchel
Założenia i cele przedmiotu:		Założeniem tego przedmiotu jest znajomość przez słuchaczy zagadnień związanych z podstawami baz danych. Wskazana jest znajomość zagadnień z zakresu systemów operacyjnych oraz inżynierii oprogramowania. Celem przedmiotu jest zaznajomienie studentów z problematyką systemów zarządzania bazami danych, m.in.: • architekturą systemów zarządzania bazami danych, • metodami przechowywania danych przez bazy danych, • optymalizacją zapytań w bazach danych, • rodzajami i zastosowaniem różnych systemów baz danych.
Symbol efektu	Efekty uczenia się	
	WIEDZA Student zna i rozumie:	
W_01	w zaawansowanym stopniu zagadnienia z zakresu podstaw teoretycznych baz danych, w szczególności systemów zarządzania bazami danych	W_S02, W_S04
W_02	mechanizmy zarządzania danymi na trwałych nośnikach oraz w obszarze pamięci operacyjnej	W_S02, W_S04
W_03	typowe mechanizmy zarządzania danymi w rozproszonych systemach baz danych	W_S02, W_S04
	UMIEJĘTNOŚCI Student potrafi:	
U_01	zoptymalizować zapytanie wysłane do bazy danych	U_S01, U_S03
U_02	dobrać właściwy model bazy danych dla prezentowanego problemu i zaproponować działającą aplikację w oparciu o ten model	U_S01, U_S03
U_03	zaprojektować system informatyczny wykorzystujący w zaawansowanym stopniu wskazaną bazę danych	U_S01, U_S03

Forma i typy zajęć:	studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:	
Warunkiem uczestnictwa w zajęciach jest znajomość zagadnień z przedmiotów „Bazy danych”, „Systemy operacyjne” i „Inżynieria oprogramowania” lub znajomość literatury obowiązującej w tych przedmiotach.	
Treści modułu kształcenia:	
Treści dotyczące wykładu:	
Wprowadzenie do systemów zarządzania bazami danych (SZBD). Rola i zalety SZBD. Typy baz danych. Ewolucja systemu plików. Problemy z przetwarzaniem danych w systemie plików. Funkcje SZBD. Języki programowania w bazach danych. Oracle PL/SQL oraz Microsoft T-SQL. Typy danych. Struktury składowane. Wyrażenia, operatory, funkcje i wyjątki. Struktury sterujące. Przegląd funkcji wbudowanych. Model relacyjny bazy danych. Logika predykatów i teoria zbiorów. Logiczny widok danych. Klucze. Reguły integralności. Algebra relacyjna. Katalogi danych. Relacje. Indeksy. Reguły Codd’a. Architektura systemu zarządzania bazą danych. Architektura relacyjnych systemów zarządzania bazą danych. Kolumnowe i wierszowe systemy SZBD. Magazyny z szerokimi kolumnami. Pliki danych i pliki indeksowe. Podstawy B-drzew. Drzewa wyszukiwania binarnego. Równoważenie drzewa. Drzewa dla pamięci masowych opartych na dyskach. Struktury oparte na dyskach. B-drzewa. Klucze oddzielające. Algorytm przeszukiwania B-drzewa. Podział węzłów B-drzewa. Scalanie węzłów B-drzewa. Formaty plików. Organizacja B-drzew na dysku. Kodowanie binarne. Struktura stron. Strony podzielone na obszary. Układ komórek. Zarządzanie danymi o zmiennym rozmiarze. Wersjonowanie. Sumy kontrolne. Implementowanie B-drzew. Nagłówek strony. Wyszukiwanie binarne. Propagowanie podziałów i scaleń. Przywracanie równowagi. Dołączanie tylko z prawej strony. Kompresja. Odkurzanie i konserwacja. Przetwarzanie transakcji i przywracanie poprzedniego stanu. Pojęcie transakcji. Transakcje w systemie bazy danych. Zarządzanie buforami. Semantyka buforowania. Zwalnianie pamięci podręcznej. Blokowanie stron w pamięci podręcznej. Zastępowanie stron. Przywracanie poprzedniego stanu. Kontrola współbieżności. Serializowalność. Izolacja transakcji. Anomalie odczytu i zapisu. Odmiany B-drzewa. Kopiowanie przy zapisie. Abstrakcja aktualizacji węzłów. Leniwe B-drzewa. Drzewa FD. Drzewa Bw. B-drzewa nieświadome pamięci podręcznej. Przetwarzanie i optymalizacja zapytań. Strategie przetwarzania zapytań. Translacja zapytań języka SQL. Algorytmy sortowania zewnętrznego. Algorytmy operacji selekcji. Implementacja operacji JOIN. Implementacja operacji agregujących. Optymalizacja zapytań. Statystyki bazy danych. Indeksy w procesie optymalizacji. Wybór optymalizatora. Dobór indeksów. Dobór wyrażeń warunkowych. Dostrajanie wydajności SZBD. Rozproszone systemy zarządzania bazami danych. Ewolucja rozproszonych baz danych. Przetwarzanie rozproszone i rozproszone bazy danych. Charakterystyka rozproszonych systemów zarządzania bazami danych. Cechy rozproszenia danych. Przejrzystość rozproszenia. Przejrzystość transakcji. Projektowanie rozproszonych baz danych. Twierdzenie CAP.	

12. **Hurtownie danych.** Analityka biznesowa. Ewolucja i trendy w rozwoju analityki biznesowej. Systemy wspomagania decyzji. Hurtownie danych. Schemat gwiazdy. Systemy OLAP. Relacyjne systemy OLAP. Rozszerzenia SQL dla OLAP. Widoki zmaterializowane.
13. **Big Data i bazy danych NoSQL.** Hadoop. HDFS. MapReduce. Bazy danych NoSQL. Bazy danych typu klucz-wartość. Pamięciowe bazy danych. Bazy dokumentowe. Bazy kolumnowe. Bazy grafowe.
14. **Architektura bazy danych ORACLE.** Architektura systemu Oracle. Architektura klastrowa Oracle. Shardy w Oracle. Struktura pamięci SGA. Pliki systemu Oracle.
15. **Architektura bazy danych Microsoft SQL Server.** Architektura systemu Microsoft SQL Server. Przechowywanie danych. Przetwarzanie zapytań. Zarządzanie transakcjami. Buforowanie i zarządzanie pamięcią. Rozszerzenia i narzędzia w Microsoft SQL Server.

Treści dotyczące laboratorium:

1. **Instalacja i konfiguracja środowiska bazy danych Oracle.** Metody łączenia się z bazą danych. Wbudowane narzędzia. Kluczowe cechy PL/SQL. Architektura PL/SQL.
2. **Język PL/SQL cz. 1.** Struktura bloku w PL/SQL. Podstawy PL/SQL. Funkcja zmiennej podstawiania. Komponenty języka PL/SQL. Zmienne PL/SQL.
3. **Język PL/SQL cz. 2.** Typy danych w PL/SQL. Podprogramy w PL/SQL. Zagnieżdżone bloki. Etykietowanie bloków. Procedury zagnieżdżone. Procedury autonomiczne. Funkcje zagnieżdżone. Funkcje autonomiczne. SQL w bloku PL/SQL. Używanie instrukcji DML w bloku PL/SQL. Pakiety w PL/SQL.
4. **Język PL/SQL cz. 3.** Instrukcje sterujące: IF, CASE, LOOP, WHILE, FOR, CONTINUE, Zagnieżdżanie pętli. Obsługa błędów i wbudowane wyjątki.
5. **Język PL/SQL cz. 4.** Wyjątki zdefiniowane przez użytkownika. Zaawansowane użycie wyjątków. Pragma EXCEPTION_INIT. Obsługa wyjątku OTHERS. Kursory jawne i niejawne. Rekordy oparte na tabelach i kursorach. Pętle kursora FOR. Wyzwalacze bazy danych.
6. **Język T-SQL cz. 1.** Instalacja i konfiguracja środowiska bazy danych Microsoft SQL Server. Wprowadzenie do T-SQL. Typy danych. Struktura języka T-SQL. Instrukcja BEGIN...END. Instrukcje warunkowe i instrukcje pętli.
7. **Język T-SQL cz. 2.** Instrukcje związane ze zmiennymi. Procedury składowane. Funkcje i procedury wbudowane.
8. **Język T-SQL cz. 3.** Wyrażenia tablicowe.
9. **Język T-SQL cz. 4.** Transakcje i obsługa błędów w T-SQL.
10. **Optymalizacja zapytań.** Plany wykonania. Statystyki bazy danych. Tworzenie i analiza optymalnych zapytań SQL.
11. **Projekt aplikacji bazodanowej cz. 1.** Analiza bazy danych OLTP AdventureWorks. Utworzenie projektu. Projekt interfejsu użytkownika. Model danych.
12. **Projekt aplikacji bazodanowej cz. 2.** Metody łączenia się z bazą danych. Modelowanie danych.
13. **Projekt aplikacji bazodanowej cz. 3.** Implementacja interfejsu użytkownika.
14. **Projekt aplikacji bazodanowej cz. 4.** Obsługa błędów i logowanie. Zabezpieczenie bazy danych.
15. **Prezentacja i ocena aplikacji bazodanowej.**

Literatura podstawowa:

1. Petrov A.: Baza danych od środka. Analiza działania rozproszonych systemów danych, Helion, 2023
2. Coronel C., Morris S.: Database Systems: Design, Implementation, and Management, 12th Edition, Cengage Learning, 2017 (pozycja dostępna online)

Literatura dodatkowa:

1. Ramez E., Shamkant B. N.: Wprowadzenie do systemów baz danych. Wydanie VII Helion, 2019

2. Dokumentacja online bazy danych Oracle: <https://docs.oracle.com/en/database/oracle/oracle-database/>
3. Dokumentacja online bazy danych Microsoft SQL Server: <https://learn.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver16>

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Czym jest brudny odczyt? Podaj przykład takiego typu odczytu.”.
W_02	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Jakie działania wykona SZBD w przypadku wstawienia nowego klucza do węzła B-drzewa, który jest w całości zajęty?”.
W_03	Efekt realizowany na egzaminie w formie pisemnej. Przykładowe pytanie: „Czym jest sharding w bazie danych?”.
U_01	Efekt realizowany w trakcie zajęć laboratoryjnych lub na egzaminie w formie pisemnej. Przykładowe zadanie: „Przedstawiono następujące zapytanie SQL. Jakie indeksy należy utworzyć? Napisz wymagane polecenia SQL. Przedstaw dwa alternatywne plany wykonania (podaj rodzaj operacji i koszty operacji WE/WY) dla przytoczonego zapytania.”.
U_02	Efekt realizowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Zbuduj szkielet aplikacji w języku C#. Utwórz model bazy danych i przeprowadź konfigurację połączenia z bazą danych.”.
U_03	Efekt realizowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Przeprowadź konfigurację użytkowników i ich ról w systemie bazy danych. Wprowadź wskazane uprawnienia dla obiektów bazy danych.”.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena z przedmiotu składa się z dwóch ocen częściowych:

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

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

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

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

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

- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Przygotowanie się i udział w egzaminie	20 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	18 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Przygotowanie się i udział w egzaminie	30 godz.
Przygotowanie się do ćwiczeń laboratoryjnych	38 godz.
Udział w konsultacjach z przedmiotu	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Programowanie gier komputerowych
Nazwa w języku angielskim:		Computer game programming
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Wydział Nauk Ścisłych i Przyrodniczych	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:		4
Imię i nazwisko koordynatora przedmiotu:		Mirosław Barański
Imię i nazwisko prowadzących zajęcia:		Mirosław Barański Jacek Krzysztof Gajc
Opis przedmiotu i cele przedłożenia:		Założono, że studenci poznają zagadnienia związane z programowaniem gier komputerowych, algorytmy w tym zakresie, środowiska programistyczne wspomagające programowanie gier komputerowych oraz wybrane zagadnienia matematyczne i fizyczne wykorzystywane w tworzeniu gier i dedykowane systemy informatyczne (głównie silniki). Celem kursu jest zapoznanie studentów zasadami tworzenia gier komputerowych, ze środowiskami do ich programowania oraz z wybranymi zagadnieniami fizyki związanymi z grami komputerowymi, w tym z zasadami tworzenia gier na urządzenia mobilne.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie etapy tworzenia gier komputerowych	K_W06, K_W08
W_02	Zna i rozumie zasadę funkcjonowania środowisk i systemów informatycznych wspomagających tworzenie gier komputerowych	K_W06, K_W08

W_03	Zna i rozumie działania algorytmów wykorzystywanych w projektowaniu i implementacji gier komputerowych	K_W06, K_W01, K_W02, K_W10, K_W07
W_04	Zna i rozumie zaawansowane zagadnienia matematyki i fizyki i informatyki związane z projektowaniem i implementacją gier komputerowych	K_W06, K_W01, K_W02, K_W10, K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi dobrać odpowiednie metody i narzędzia modelowania i programowania niezbędne do implementacji systemów gier komputerowych, potrafi je zaimplementować	K_U09, K_U10, K_U11, K_U22
U_02	Potrafi dobrać odpowiednie środowiska, techniki i technologie do projektowania systemów gier komputerowych i wykorzystać je w implementacji	K_U09, K_U10, K_U11, K_U22
U_03	Potrafi porównać środowiska lub systemy grafiki komputerowej wybrane do projektowania systemów gier komputerowych ze względu na zadane kryteria użytkowe i ekonomiczne	K_U15, K_U16
U_04	Potrafi ocenić przydatność rutynowych metod i narzędzi do rozwiązywania zadań związanych z projektowanie gier komputerowych dotyczących akwizycji, projektowania i implementacji tych systemów	K_U17, K_U19
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatycznych zagadnień związanych z grami komputerowymi	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (30 godz.), laboratoria (45 godz.), Studia niestacjonarne: wykłady (18 godz.), laboratoria (24 godz.),	
Wymagania wstępne i dodatkowe:		
1. Znajomość podstawowych pojęć i algorytmów grafiki komputerowej lub cyfrowego przetwarzania obrazu i dźwięku		
Treści modułu kształcenia:		

1. Architektura silnika gry komputerowej
2. Organizacja danych w grach komputerowych. Graf sceny. Podstawowe mechanizmy silników graficznych.
3. Podsystem wyświetlania grafiki, eliminacja niewidocznych obiektów, renderowanie,
4. Podsystem interakcji, wykrywanie i reakcja na kolizje,
5. Programowanie ruchu wirtualnej postaci,
6. Podsystem dźwięku,
7. Zagadnienia sieciowe w grach komputerowych,
8. Zaawansowane techniki graficzne, oświetlenie na wierzchołek i piksel, mapy nierówności.
9. Dynamiczna kontrola poziomu szczegółowości sceny.
10. Wykorzystanie Biblioteki XNA w grach komputerowych,
11. Wykorzystanie silnika Unity w grach komputerowych,
12. Wykorzystanie silnika Unreal Engine w grach komputerowych,
13. Języki skryptowe w grach komputerowych,
14. Programowanie gier komputerowych na urządzeniach mobilnych,
15. Wydajność i optymalizacja gier komputerowych,

Literatura podstawowa:

1. Karl Bunyan, HTML5. Tworzenie gier z wykorzystaniem CSS i JavaScript (ebook), Helion 2016.
2. Aram Cookson i inni, Unreal Engine w 24 godziny. Nauka tworzenia gier (ebook) Helion 2017.
3. Jon Manning, Paris Buttfield-Addison, Unity. Tworzenie gier mobilnych (ebook), Helion 2018,

Literatura dodatkowa:

1. M. de Loura, Perleki programowania gier (tomy 1-3). Helion 2002.
2. Rafał Nowocień, Tworzenie gier komputerowych. Kompendium producenta, Helion 2019

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	będzie sprawdzany na egzaminie ustnym oraz podczas realizacji zadania indywidualnego. Na egzaminie będzie opisana modelowa gra komputerowa i jedną z zadań będzie wyszczególnienie oraz ogólna charakterystyka etapów projektowania gier komputerowych. Natomiast podczas realizacji zadania indywidualnego ten efekt będzie sprawdzany podczas analizy sprawozdania
W_02	będzie sprawdzany podczas egzaminu na egzaminie ustnym. Przykładowe pytanie to: Scharakteryzuj środowisko UNITY 3D.
W_03	będzie sprawdzany na egzaminie ustnym. Przykładowe pytanie to: Omów problem wykrywania kolizji. Jak można go rozwiązać w implementacji gier komputerowych.
W_04	będzie sprawdzany na egzaminie pisemnym i ustnym. Przykładowe pytanie to: . Omów jakie zjawiska i jakie problemy występują podczas ruchu dwu kul. . Omów proces tworzenia gry „Zgaduj-Zgadula”.

	Opisz metodę postępowania przy programowaniu tej gry na urządzenie mobilne w systemie Android.
U_01-U_04	będą sprawdzane na zajęciach laboratoryjnych i poprzez odpowiednio sformułowane zadanie indywidualne. Zajęcia laboratoryjne będą oceniane po każdym zajęciach. Zadania tam realizowane będą polegać na projektowaniu i implementacji gier komputerowych (prosty i złożonych w ramach zadania indywidualnego).
K_01	będzie weryfikowany w czasie realizacji zadania indywidualnego. Zadania indywidualne będą realizowane w grupach 3-4 osobowych. Na wybranych zajęciach i konsultacjach studenci będą przedstawiali swoje rozwiązania. Na regularnych zajęciach będą one krytycznie oceniane przez całą grupę.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego zadania indywidualnego według schematu: Regularne zajęcia – 26 pkt., Obrona zadania indywidualnego – 14 pkt. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 13 pkt., obrona indywidualnego zadania – co najmniej 7 pkt. Na tej formie zajęć student może maksymalnie uzyskać 40 pkt. Egzamin jest egzaminem ustnym. Można na nim uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. Ocena końcowa z modułu, w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS): · 0 – 50 pkt: niedostateczna (F), · 51 – 60 pkt: dostateczna (E), · 61 – 70 pkt: dostateczna plus (D), · 71 – 80 pkt: dobra (C), · 81 – 90 pkt: dobra plus (B), · 91 – 100 pkt: bardzo dobra (A). Poprawy: Jednorazowa poprawa każdego kolokwium w trakcie zajęć w semestrze. Dwie poprawy obu kolokwium w sesji egzaminacyjnej, odpowiednio przed drugim i trzecim terminem egzaminu pisemnego. Uwaga: Istnieje możliwość zwolnienia z egzaminu pisemnego lub ustnego studentów wyróżniających się na zajęciach laboratoryjnych. Warunkiem koniecznym zwolnienia z egzaminu jest uzyskanie 90% punktów możliwych do zdobycia w trakcie regularnych zajęć laboratoryjnych. Decyzję o ewentualnym zwolnieniu podejmuje osoba przeprowadzająca egzamin po zasięgnięciu opinii (poprzez rozmowę) osób prowadzących zajęcia. Decyzję o zwolnieniu prowadzący wykład przekazuje studentom nie później niż 2 tygodnie przed końcem semestru.	
Bilans punktów ECTS:	

Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	45 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	8 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	38 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	18 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Systemy animacji komputerowej
Nazwa w języku angielskim:		Systems of Computer Animation
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Mirosław Szaban
Imię i nazwisko prowadzących zajęcia:		Mirosław Barański Jacek Krzysztof Gajc
Założenia i cele przedmiotu:		Celem przedmiotu jest nabycie przez studenta wiedzy na temat technik, standardów oraz dostępnych narzędzi dedykowanych do tworzenia ruchomych obiektów i filmów zarówno w wersji dwuwymiarowej jak i trójwymiarowej. Efektem pozyskanej przez studenta wiedzy jest umiejętność samodzielnego wykonania animacji, filmu ukazującego poruszający się realistyczny obiekt.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Ma uporządkowaną wiedzę na temat zagadnień związanych z tworzeniem i wykorzystaniem animacji komputerowej.	K_W01, K_W02, K_W06, K_W09, K_W12
W_02	Zna zasady, metody i standardy tworzenia animacji cyfrowych.	K_W01, K_W02, K_W06, K_W09, K_W12
W_03	Zna wybrane narzędzia i aplikacje służące do tworzenia i obróbki animacji cyfrowych.	K_W01, K_W02, K_W06, K_W09, K_W12

Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Rozwiązując problemy związane z grafiką komputerową i przetwarzaniem obrazów potrafi dobrać i wykorzystać odpowiednie metody i narzędzia modelowania i programowania, odpowiednie środowiska programistyczne, techniki i technologie programistyczne niezbędne do implementacji animacji komputerowej, oraz potrafi ocenić ich przydatność.	K_U01, K_U02, K_U05, K_U08, K_U10, K_U11, K_U12, K_U13, K_U18, K_U19, K_U24
U_02	Potrafi wykonać animację w środowisku 3 ds. MAX.	K_U01, K_U02, K_U05, K_U08, K_U10, K_U11, K_U12, K_U13, K_U18, K_U19, K_U24
U_03	Potrafi wykonać animację poruszającego się człowieka z uwzględnieniem animacji układu kostnego.	K_U01, K_U02, K_U05, K_U08, K_U10, K_U11, K_U12, K_U13, K_U18, K_U19, K_U24
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych dotyczących animacji komputerowej.	K_K01
K_02	Potrafi wykorzystać metody i techniki charakterystyczne dla systemów grafiki komputerowej do realizacji przedsięwzięć biznesowych.	K_K03
Forma i typy zajęć:		studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (45 godz.) studia niestacjonarne: wykłady (18 godz.), ćwiczenia laboratoryjne (24 godz.)
Wymagania wstępne i dodatkowe:		
1. Umiejętność podstaw programowania.		
2. Znajomość zagadnień z zakresu grafiki komputerowej i cyfrowego przetwarzania obrazu i dźwięku.		
Treści modułu kształcenia:		
1. Animacja poklatkowa. Korzystanie z linii czasu, Tworzenie klatek kluczowych, Zaznaczanie klatek, Manipulowanie klatkami jednej warstwy, Usuwanie klatek, Tworzenie prostej animacji poklatkowej, Test Movie, Tryb Onion Skin. 2. Animacje z automatycznym uzupełnianiem klatek. Płynna animacja ruchu obiektu, Animacja obiektu zmieniającego kolor, Animacja z obracaniem obiektu, Animowanie grafik zmieniających rozmiar, Animacja z metamorfozą, Animacja obiektu po ścieżce, Animacja grafiki z maską. 3. Skomplikowane zadania animacyjne. Animacja z obiektem Movie Clip. Action Script. Panel Actions. Dołączanie akcji do klatki (Funkcje Movie Control, Goto and stop, Goto and play, Stop, Play), Przypisywanie etykiet do klatki.		

4. **Dołączanie dźwięku i wideo.** Importowanie dźwięków, Dołączanie dźwięków do klatek, Import materiału wideo, Dołączanie materiału wideo do klatek. **Interaktywność z obiektami.** Tworzenie prostych przycisków, Tworzenie przycisków zmieniających kształt, Dołączanie akcji do przycisku. **Publikacja filmów w Internecie.** Publikowanie filmów, Eksportowanie filmów Flash do innych formatów.
5. **Animacje zaawansowane.** Animacja w przestrzeni trójwymiarowej (obróć, przemieszczanie obiektów, Global Transform), Narzędzie Deco Tool, Kinematyka odwrotna i kości, Sztuczki i kruczki w animacji Flash. Zwiększanie realizmu animacji: cień i perspektywa.
6. **Importowanie i eksportowanie grafiki oraz multimediiów.** Formaty plików dla obsługiwanych przez technologię Flash dla importu i eksportu, Import i eksport plików Phostoshop, Fireworks, Illustrator, Praca z plikami dźwiękowymi, Praca z plikami wideo.
7. **Zaawansowana animacja ruchu.** precyzyjna kontrola i edycja sekwencji animacji przy użyciu nowego narzędzia Motion Editor. Posługiwanie się zestawami ustawień sekwencji animacji. Zagnieżdżanie sekwencji animacji.
8. **Zaawansowane techniki animacji.** Zaawansowane techniki synchronizowania obrazu z dźwiękiem. Wzbogacanie i uatrakcyjnianie animacji elementami i efektami grafiki 3D.
9. **System kości.** Teoria i praktyka animowania obiektów i postaci w sposób interaktywny. Animacja ruchu i kształtu obiektów przy użyciu systemu kości.
10. **Animacja chodu postaci.** technicznie i artystycznie poprawne projektowanie sekwencji animacji kończyn i ciała postaci. Publikowanie animacji w formacie zgodnym ze starszymi wersjami programu Adobe Flash.
11. **Podstawy teoretyczne pracy z grafiką 3D.** Podstawy pracy w pakiecie 3D na przykładzie programu 3ds max, Interfejs programu 3ds MAX, Koncepcja projektu i organizacja sceny 3D, Podstawowe operacje na obiektach.
12. **Modelowanie scen w 3ds MAX I.** Modelowanie w oparciu o wielokąty i powierzchnie podpodziału (subdivision surfaces), Modelowanie w oparciu o powierzchnie NURBS, Modelowanie i kształtowanie obiektów przy użyciu modyfikatorów;
13. **Modelowanie scen w 3ds MAX II.** Tworzenie i edycja map współrzędnych UV, Tworzenie i edycja map tekstur, Oświetlenie i rendering: reguły oświetlania sceny 3D za pomocą różnych źródeł światła (w tym IBL oraz Global Illumination); kontrolowanie cieni; renderowanie obrazów w module wewnętrznym oraz za pomocą modułów Mental Ray i Vray;
14. **Tworzenie animacji w 3ds MAX.** Podstawy tworzenia i animowania postaci w module Character Studio: konstruowanie szkieletu, mięśni i realistycznie odkształcającej się powłoki, Podstawy dynamiki obiektów w scenach statycznych i animacjach: interakcja pomiędzy ciałami sztywnymi, giętkimi i sprężystymi; dynamika tkanin; dynamika włosów, futra i sierści;
15. **Podstawy symulacji zjawisk fizycznych.** Emisja cząsteczek i interakcja cząsteczek z obiektami w scenie, Kompletowanie projektu i prezentowanie zwizualizowanych scen statycznych oraz animacji.

Literatura podstawowa:

1. Parent Rick, Animacja komputerowa, Algorytmy i techniki, PWN 2012
2. Tony Parisi, Aplikacje 3D : przewodnik po HTML5, WebGL i CSS3, Helion 2015.
3. Joanna Pasek. 3ds max 2010. Animacja 3D od podstaw. Szkoła efektu. Helion 2010.

Literatura dodatkowa:

1. Dev Ramtal, Adrian Dobre. Wprowadzenie do fizyki w grach, animacjach i symulacjach Flash, Helion 2013.

2. Parent R. Computer Animation, Algorithms and Techniques, Morgan Kaufmann, 2002

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

W_01-W_03	prawdzone będą na egzaminie pisemnym w sesji egzaminacyjnej. . Przykładowe pytania: Omów rolę animacji układu kostnego w animacji postaci. • Wymień i krótko scharakteryzuj podstawowe zasady, metody i standardy tworzenia animacji cyfrowych. • Opisz wybrane narzędzie lub aplikację służące do tworzenia i obróbki animacji cyfrowych.
U_01-U_03	sprawdzone będą na bieżąco, na każdych zajęciach poza pierwszym i ostatnim. Przykładowe zadania: Ponadto sprawdzane będą poprzez wykonanie zadanie indywidualnego (projektu). • Wykonaj prostą animację układu kostnego postaci, wybierz właściwe narzędzie. • Wykonaj prostą animację przy pomocy aplikacji Flash, dołącz plik dźwiękowy w tle animacji. • Korzystając z narzędzi 3ds Max wykonaj animację biegnącego człowieka. • Wykonaj projekt, dobierz właściwe narzędzie i wykonaj animację pojazdu przejeżdżającego przez skrzyżowanie, widok z pierwszej osoby kierowcy pojazdu (kierowca nieruchomy patrzy przed siebie). Kierowca widzi inne pojazdy, ludzi poruszających się chodnikiem oraz przekraczających skrzyżowanie na pasach. Utwórz dokumentację.
K_01-K_02	prawdzone będą poprzez wykonanie zadanie indywidualnego (projektu). Przykładowe zadanie: • Wykonaj projekt, dobierz właściwe narzędzie i wykonaj animację pojazdu przejeżdżającego przez skrzyżowanie, widok z pierwszej osoby kierowcy pojazdu (kierowca nieruchomy patrzy przed siebie). Kierowca widzi inne pojazdy, ludzi poruszających się chodnikiem oraz przekraczających skrzyżowanie na pasach. Utwórz dokumentację.

Forma i warunki zaliczenia:

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

Regularne zajęcia – 45 pkt.,

Obrona zadania indywidualnego – 15 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 23 pkt., obrona indywidualnego zadania – co najmniej 8 pkt. Na tej formie zajęć student może maksymalnie uzyskać 60 pkt.

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 40 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 20 pkt.

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

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

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	45 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	12 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	10 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	18 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	28 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	27 godz.

Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Akwizycja i przetwarzanie informacji wizualnej
Nazwa w języku angielskim:		
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		fakultatywny
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	trzeci	
Semestr:	szósty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		Dr Mirosław Szaban
Imię i nazwisko prowadzących zajęcia:		Dr Andrzej Salamończyk, mgr inż. Monika Wereszyńska
Założenia i cele przedmiotu:		Celem modułu jest zapoznanie z metodami pozyskiwania i przetwarzania informacji wizualnej.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	Zna i rozumie funkcjonowanie urządzeń i metod akwizycji wizualnej.	K_W08
W_02	Zna i rozumie funkcjonowanie elementów składowych systemu automatycznego widzenia.	K_W08
W_03	Zna i rozumie zasady korzystania z przykładów deskryptorów kształtu	K_W08
	UMIEJĘTNOŚCI Student potrafi:	
U_01	przetworzyć wstępnie obraz rastrowy w celu wykorzystania go w systemach rozpoznających.	K_U11, K_U09, K_U15, K_U16, K_U17, K_U19
U_02	Wykorzystywać biblioteki i algorymy do przetwarzania informacji wizualnej	K_U11, K_U09, K_U15, K_U16, K_U17, K_U19
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy oraz jest gotów do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu akwizycji i przetwarzania informacji wizualnej	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
Znajomość podstawowych pojęć i algorytmów grafiki komputerowej		
Treści modułu kształcenia:		

1. Wprowadzenie do systemów wizyjnych Struktura i etapy przetwarzania informacji wizualnej. Elementy systemu wizyjnego. 2. Akwizycja obrazu Zasada działania kamer cyfrowych (matryce CCD i CMOS). 3. Parametry kamer: rozdzielczość, czułość, szумы, zakres dynamiczny. Układy optyczne – ogniskowa, głębia ostrości, zniekształcenia geometryczne. 4. Cyfrowe formaty obrazów, przestrzenie barw (RGB, HSV, YCbCr, LAB). Kalibracja kamery (wewnętrzna i zewnętrzna). 5. Reprezentacja i modelowanie obrazu 6. Wstępne przetwarzanie obrazu. Przegląd bibliotek i narzędzi do przetwarzania informacji wizualnej 7. Segmentacja i ekstrakcja cech 8. Opis kształtu: momenty geometryczne, kontury, deskryptory Fouriera, cechy tekstury. Detekcja cech lokalnych: Harris, SIFT, SURF, ORB. 9. Analiza i interpretacja informacji wizualnej Klasyfikacja obiektów w obrazie. 10. Zastosowanie metod uczenia maszynowego w analizie obrazów 11. Zastosowanie metod uczenia głębokiego w analizie obrazów 12. Modele YOLO 13. Kalibracja i rekonstrukcja przestrzenna Stereo-wizja i rekonstrukcja 3D. 14. Wideo i analiza sekwencji czasowych Detekcja ruchu Przepływ optyczn 15. Zastosowania systemów przetwarzania informacji wizualnej	
Literatura podstawowa:	
1. R Szeliski. Computer Vision: Algorithms and Applications. Springer 2022 2. K. Stąpor, Automatyczna klasyfikacja obrazów, EXIT, 2005	
Literatura dodatkowa:	
1. D. Forsyth, J. Ponce. Computer Vision: A Modern Approach, 2nd Edition. Pearson 2012 2. A. Kaehler, G. Bradski, OpenCV 3: Komputerowe rozpoznawanie obrazów w C++ przy użyciu biblioteki OpenCV, Helion 2017. 3. E. R. Davies. Computer Vision Principles, Algorithms, Applications, Learning. Academic Press 2017	
Planowane formy/działania/metody dydaktyczne:	
Wykład tradycyjny wspomagany technikami multimedialnymi. Zamieszczanie na stronach internetowych zadań i materiałów do laboratoriów.	
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
Symbol efektu	Metody weryfikacji efektów uczenia się

W_01, W_02, W_03	Efekty sprawdzane na egzaminie pisemnym Przykładowe pytania: Wymień i omów wybrane elementy składowe systemu automatycznego widzenia.. Podaj przykłady i omów wybrane deskryptory kształtu.
U_01, U_02	Efekty sprawdzane na zajęciach laboratoryjnych
K_01	Obserwacja postawy studenta
Forma i warunki zaliczenia:	
Wszystkie zajęcia laboratoryjne (za wyjątkiem pierwszego) są oceniane. W przypadku nieobecności studenta na laboratorium sposób ich odpracowania określa osoba odpowiedzialna za kurs i jest on podany na pierwszych zajęciach. Na każdych ćwiczeniach prowadzący podaje zakres zadań dla studentów do przygotowania na następne zajęcia (o charakterze praktycznym lub teoretycznym) i wytyczne do ich realizacji. Przygotowanie do zajęć oraz realizacja zadań każdego ćwiczenia są oceniane w skali od 0 do 10 pkt. Łącznie student za 14 zajęć może uzyskać od 0 do 140 pkt.. Warunek uzyskania zaliczenia laboratorium: co najwyżej dwie nieusprawiedliwione nieobecności na laboratoriach i uzyskanie łącznie co najmniej 71 punktów z zajęć (na 140 możliwych) Zaliczenie laboratorium jest warunkiem koniecznym przystąpienia do egzaminu. Egzamin odbywa się w formie pisemnej, za egzamin można uzyskać maksymalnie 100pkt. Ocena końcowa z zajęć zależy od wyniku laboratorium (w 60%) i egzaminu (w 40%), a końcowy wynik punktowy oblicza się w następujący sposób: $P=60(L/140)+40(E/100),$ gdzie P-końcowy wynik punktowy(maksymalnie 100pkt.) , L-punkty uzyskane z części laboratoryjnej, E- punktowy wynik egzaminu. Ocena z zajęć zależy od końcowego wyniku punktowego i wyznacza się w następujący sposób. • 0-50 punktów – 2 • 51-60 punktów – 3 • 61-70 punktów - 3,5 • 71-80 punktów – 4 • 81-90 punktów – 4,5 • 91-100 punktów – 100 Sposób uzyskania punktów: Laboratorium Ocena udziału w laboratoriach oraz przygotowania się do tych zajęć: 140 pkt. (14 zajęć po 10 pkt.). Wykład Egzamin pisemny: 100 pkt.	
Bilans punktów ECTS:	
Studia stacjonarne	

Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	18 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu	18 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	30 godz.
Udział w ćwiczeniach laboratoryjnych	30 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	18 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu	18 godz.
Obecność na egzaminie	2 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4