

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

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Systemy wbudowane
Nazwa w języku angielskim:	Embedded systems	
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:	drugi	
Semestr:	czwarty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski
Imię i nazwisko prowadzących zajęcia:		dr Marcin Stępnia, mgr Mateusz Przychodzki
Założenia i cele przedmiotu:		Zakłada się, że student zna języki assemblerów oraz C/C++ na poziomie podstawowym. Celem kursu jest uzyskanie wiedzy i nabycie umiejętności dotyczącej podstaw systemów wbudowanych. Realizacja praktycznych zadań będzie odbywała się na platformie Arduino Uno i ESP8266.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA	
	Student zna i rozumie:	
W_01	zagadnienia dotyczące struktur i funkcjonalności wbudowanych systemów mikroprocesorowych	K_W02, K_W05
W_02	zagadnienia z zakresu działania modułów interfejsowych wykorzystywanych we wbudowanych systemach mikroprocesorowych	K_W05
W_03	zagadnienia na temat zasad działania różnych protokołów służących do budowy sieci przewodowych i bezprzewodowych m.in. w obszarze automatyki budynkowej i motoryzacji	K_W07

UMIEJĘTNOŚCI		
Student potrafi:		
U_01	pracować z dokumentacją wybranego mikrokontrolera.	K_U01, K_U06
U_02	zaprojektować, zbudować i zaprogramować wbudowany system mikroprocesorowy w oparciu o platformę Arduino Uno.	K_U01, K_U18, K_U24
U_03	dobrać moduł interfejsowy oraz zrealizować na nim komunikację pomiędzy układami elektronicznymi.	K_U17, K_U21, K_U24
U_04	weryfikować poprawność napisanego programu, potrafi dobrać odpowiednie dane testowe.	K_U19, K_U22
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 informatyki.	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 języków assemblerów oraz C/C++ na poziomie podstawowym		
Treści modułu kształcenia:		
Wykłady: Wstęp do systemów wbudowanych. Ważne wydarzenia historyczne. Podstawowe definicje. Zastosowania systemów wbudowanych. Podstawowe elementy systemów mikroprocesorowych. Typy rejestrów, rodzaje pamięci i magistral, elementy CPU. Paradygmaty architektoniczne systemów komputerowych. Platformy sprzętowe umożliwiające tworzenie układów wbudowanych. Przegląd producentów układów mikroprocesorowych. Zasady projektowania systemów wbudowanych. Przegląd platform Arduino. Organizacja pamięci oraz cykl wykonania rozkazów w ATmega328. Sekcje pamięci programowej i operacyjnej oraz ich znaczenie. Budowa rozkazów w mikrokontrolerach ATmega328. Cykl wykonania rozkazu. Porty I/O oraz system przerwań w ATmega328. Funkcje portów oraz ich budowa. Rodzaje przerwań. Algorytm wykonywania przerwań. System przerwań w ATmega328. Układy czasowo-licznikowe w ATmega328. Budowa i zasada działania układów czasowo-licznikowych. Tryby pracy układów czasowo-licznikowych w mikrokontrolerze ATmega328. Moduły interfejsowe RS232 i USART. Budowa ramki RS232. Błędy związane z transmisją danych. Sposoby kontroli przepływu danych. Budowa modułu USART w mikrokontrolerze ATmega328. Tryby synchroniczny i asynchroniczny. Konfiguracja szybkości transmisji. Moduły interfejsowe SPI i I²C. Budowa modułów SPI i I²C. Komponenty sygnałowe I²C. Standardy szybkości transmisji. Sposoby wyboru urządzenia do komunikacji. Unikanie kolizji w I²C - procedura arbitrażu. Moduły interfejsowe stosowane w motoryzacji: CAN i LIN, Rlex Rey, MOST. Zastosowania, wersje protokołów i ich rozszerzenia. Budowa ramki CAN i LIN. Obsługa błędów.		

10. **Sieci bezprzewodowe w automatyce domowej i Internet of Things (IoT).** Charakterystyka protokołów Z-Wave, ZigBee, Bluetooth i innych. Rodzaje urządzeń w sieciach. Topologie sieciowe. Parametry sieci (częstotliwości, kanały, szybkość). Adresowanie urządzeń. Tryby pracy urządzeń. Energooszczędność.
11. **Raspberry Pi:** Możliwości i zastosowania platformy. Historia i specyfikacje Raspberry Pi. Systemy operacyjne. Przegląd interfejsów (wyposażenia). Symulatory i środowiska programistyczne.

Ćwiczenia:

1. **Wprowadzenie do Arduino.** Podstawowe informacje na temat platformy Arduino Uno oraz diody RGB ze wspólną katodą/anodą. Nabycie umiejętności podłączania układów według schematów, dobierania rezystorów, czytania dokumentacji i tworzenia prostych programów na platformę Arduino.
2. **Obsługa wejść, przyciski.** Zapoznanie się z konfiguracją wyprowadzeń jako **wejścia**. Odczytywanie stanu wyprowadzeń, w tym pozbycie się stanów nieustalonych i zakłóceń podczas odczytu tego stanu.
3. **Klawiatura matrycowa.** Budowa klawiatury membranowej oraz jej obsługa. Rozdanie tematów projektów indywidualnych.
4. **Wyświetlacz.** Biblioteki umożliwiające obsługę wyświetlacza JHD162A-YG na platformie Arduino. Zapoznanie się z protokołem komunikacyjnym wykorzystywany przez wyświetlacz.
5. **Pamięć nieulotna.** Zapoznanie się ze sposobami zapisu danych do pamięci nieulotnej EEPROM i programowej.
6. **Wejścia analogowe i układy czasowo-licznikowe.** Konfiguracja wejść analogowych i układów czasowo-licznikowych. Sterowanie urządzeniami za pomocą sygnału PWM.
7. **Czujnik ultradźwiękowy.** Zapoznanie się z czujnikiem HC-SR04. Konfiguracja oraz zbieranie i wykorzystanie pomiarów z tego czujnika.
8. **Komunikacja szeregową.** Standardowa komunikacja szeregową i biblioteki I²C umożliwiające komunikację za pomocą interfejsu 2-Wire. Realizacja komunikacji pomiędzy wieloma urządzeniami.
9. **Silniki krokowe, serwomechanizmy.** Zapoznanie się z zasadą działania silników krokowych 28BYJ-48 oraz serwomechanizmu Serwo TowerPro SG-92R micro. Konfiguracja i wykorzystanie tych elementów w aplikacji.
10. **ESP8266 – mikrokontroler z modulem WIFI.** Mikrokontroler ESP8266 z wbudowanym modulem wifi umożliwiającym komunikację bezprzewodową.
11. **ESP8266 – protokół MQTT.** Zapoznanie się z protokołem MQTT (MQ Telemetry Transport) i realizacja praktycznych zadań z wykorzystaniem mikrokontrolera ESP8266.
12. **Obrona projektów indywidualnych.** Testowanie i weryfikacja poprawności realizacji projektu indywidualnego. Możliwe niewielkie modyfikacje oprogramowania, w celu weryfikacji samodzielności jego wykonania.

Literatura podstawowa:

1. M. Wolf, Embedded System Interfacing, Design for the Internet-of-Things and Cyber-Physical Systems, Morgan Kaufmann, 2019
2. M Kardaś. Mikrokontrolery AVR. Język C - podstawy programowania. Wyd. 2. Wyd. ATNEL 2013.
3. Martin Evans, Joshua Noble, Jordan Hochenbaum. Arduino w akcji. Helion 2014.

Literatura dodatkowa:

1. R. Faludi. Building Wireless Sensor Networks with ZigBee, XBee, Arduino, and Processing (ebook), Helion 2010,
2. Monk. Elektronika z wykorzystaniem Arduino i Raspberry Pi. Receptury. Helion 2018,

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. Zdefiniuj pojęcie "System wbudowany". Opisz uogólniony schemat funkcjonalny systemu. 2. Scharakteryzuj organizację pamięci mikrokontrolera ATmega328. 3. Scharakteryzuj system przerwań mikrokontrolera ATmega328.
W_02	Sprawdzone podczas egzaminu w formie pisemnej. Przykładowe pytania: 1. Scharakteryzuj i porównaj funkcje modułu I2C i SPI. 2. Omów budowę i działanie generatora zegara modułu USART w ATmega328.
W_03	Sprawdzone podczas egzaminu w formie pisemnej. Przykładowe pytania: 1. Scharakteryzuj ramkę protokołu I2C. 2. Na czym polega procedura arbitrażu stosowana w CAN?
U_01	Systematycznie sprawdzane na zajęciach. Zadania na następne laboratorium muszą być dostępne tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować. Przykładowe zadania: 1. Zapoznaj się z funkcjonalnością środowiska Arduino IDE. 2. Zapoznaj się z dokumentacją wyświetlacza HD4478, a następnie wyświetla na nim napis "Witaj twoje_imie".
U_02	Systematycznie sprawdzane na zajęciach. Przykładowe zadania: 1. W oparciu o płytkę Arduino Uno, zaprojektuj układ oraz zaimplementuj program, który steruje na przemian zapalającymi się diodami. 2. W oparciu o płytkę Arduino Uno, zaprojektuj układ oraz napisz program, który wykorzystuje układ czasowo-licznikowy.
U_03	Systematycznie sprawdzane na zajęciach. Przykładowe zadania: 1. Użyj interfejsu TWI/I2C, w celu wyświetlenia napisu na wyświetlaczu LCD. 2. Połącz ze sobą 2 układy Arduino Uno za pomocą modułów USART w trybie synchronicznym. Podłącz do układu DCE klawiaturę, a do DTE 5 diod. Naciśnięcie klawisza na klawiaturze powinno powodować zapalenie odpowiedniej diody.
U_04	Systematycznie sprawdzane na zajęciach. Przykładowe zadania: 1. Oblicz wartości oporników potrzebnych do podłączenia diod RGB, następnie dobierz odpowiedni opornik. Przetestuj układ z wykorzystaniem symulatora Tinkercad. 2. Podłącz do układu potencjometr, a następnie zaprogramuj go w Arduino IDE, w celu przetestowania sterowania silnikiem DC.

K_01	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.,
- Obrona zadania indywidualnego – 20 pkt.

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

Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 26 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	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Przygotowanie się do zajęć laboratoryjnych	10 godz.
Praca nad projektem indywidualnym	8 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Obecność na egzaminie	1 godz.
Przygotowanie się do egzaminu	10 godz.

Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Przygotowanie się do zajęć laboratoryjnych	18 godz.
Praca nad projektem indywidualnym	8 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.
Obecność na egzaminie	1 godz.
Przygotowanie się do egzaminu	17 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Podstawy przetwarzania rozproszonego
Nazwa w języku angielskim:	Fundamentals of distributed computing	
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):		pierwszego stopnia
Rok studiów:	drugi	
Semestr:	czwarty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Marcin Stępnia
Imię i nazwisko prowadzących zajęcia:		dr Marcin Stępnia
Założenia i cele przedmiotu:		Studenci przystępujący do tego przedmiotu powinni znać zasady programowania obiektowego, a także treści z przedmiotu „algorytmy i złożoność”. Celem przedmiotu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z przetwarzaniem rozproszonym oraz modelami wykorzystywanymi w projektowaniu i implementacji systemów rozproszonych. Studenci poznają podstawy teorii grafów, modele obliczeniowe i podstawowe algorytmy wykorzystywane w przetwarzaniu rozproszonym. W ramach tego przedmiotu studenci poznają także mechanizmy synchronizacji zegarów oraz algorytmy budowy drzewa rozpinającego, numerowania węzłów i rutowania. Dodatkowo poznają algorytmy wyboru przywódcy, mechanizmy wzajemnego wykluczania, a także sposoby wykrywania zakleszczenia, zakończenia i awarii. Na koniec studenci zostają zapoznani z podstawami technologii blockchain.
Symbol efektu	Efekty uczenia się	
	WIEDZA Student zna i rozumie:	
W_01	podstawowe standardy, pojęcia i modele przetwarzania rozproszonego.	
		K_W06, K_W07, K_W09

W_02	zagadnienia z teorii grafów wykorzystywane w przetwarzaniu rozproszonym, definicje podstawowych pojęć i wybrane twierdzenia.	K_W01
W_03	zagadnienia komunikacji zachowującej uporządkowanie wiadomości, konstrukcji spójnego obrazu stanu globalnego, detekcji zakleszczenia rozproszonego oraz detekcji zakończenia przetwarzania rozproszonego.	K_W07
W_04	algorytmy przetwarzania w zawodnym środowisku rozproszonym, w tym zagadnienia: modelowania i klasyfikacji awarii.	K_W07
W_05	technologię blockchain i jej zastosowania.	K_W07
Symbol efektu	UMIEJĘTNOŚCI Student potrafi:	Symbol efektu kierunkowego
U_01	zbudować aplikację działającą w środowisku rozproszonym.	K_U01, K_U10, K_U11
U_02	wybrać i zaimplementować odpowiednie do problemu i zgodne ze standardami algorytmy rozproszone.	K_U01, K_U10, K_U11, K_U19, K_U20
U_03	programować algorytmy i systemy rozproszone w środowisku Java.	K_U10, K_U17, K_U22
Symbol efektu	KOMPETENCJE SPOŁECZNE Student jest gotów do:	Symbol efektu kierunkowego
K_01	podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań przetwarzania rozproszonego.	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.); studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
Warunkiem uczestnictwa w zajęciach jest wcześniejsze uzyskanie zaliczenia z następujących przedmiotów: 1. programowanie obiektowe, 2. algorytmy i złożoność, lub znajomość literatury obowiązującej na tych przedmiotach.		
Treści modułu kształcenia:		
Wykłady: 1. Wprowadzenie do przetwarzania rozproszonego. Podstawowe pojęcia i definicje. Charakterystyka środowiska przetwarzania rozproszonego. Przykłady środowisk rozproszonych. 2. Podstawy teorii grafów w przetwarzaniu rozproszonym. Definicje podstawowych pojęć. Wybrane twierdzenia. Struktury grafów mające zastosowanie w programowaniu rozproszonym. 3. Modele obliczeniowe programów rozproszonych. Pojęcie modelu obliczeniowego. Model procesów współbieżnych. Cięcia, stan globalny, stan globalny spójny. Podnoszenie z awarii.		

4. **Synchronizacja zegarów.** Czas fizyczny a logiczny. Algorytmy czasu logicznego.
5. **Elementarne algorytmy rozproszone.** Podstawowe własności algorytmów. Algorytm fali, przemierzania. Algorytm rozgłaszania.
6. **Budowa drzewa rozpinającego i numerowanie węzłów.** Pojęcie drzewa rozpinającego. Algorytm echo. Algorytm Tariego. Problem minimalnego drzewa rozpinającego. Zagadnienie enumeracji węzłów. Algorytm w topologii drzewa.
7. **Wybór przywódcy.** Algorytm Merlin-Segall. Algorytm potopowy. Algorytm na pierścieniu. Algorytm na drzewie. Algorytmy w sieciach anonimowych.
8. **Rozproszone wykrywanie zakończenia i awarii.** Wykrywanie zakończenia. Algorytm na podwójnym pierścieniu. Algorytm energetyczny. Wykrywanie awarii. Metoda ping-ack. Metoda heart-beat.
9. **Wzajemne wykluczanie.** Algorytm centralnego serwera. Algorytm Lamporta. Algorytm głoszący Maekawa. Algorytm Ricarta-Agrawali. Algorytm pierścienia z żetonem. Algorytm Raymonda.
10. **Wykrywane zakleszczenia.** Pojęcie i strategię postępowania z zakleszczeniami. Podejście scentralizowane i rozproszone. Algorytm Chandy-Misra-Haas.
11. **Technologia blockchain.** Omówienie podstawowych zastosowań. Kryptowaluty na przykładzie Bitcoin.

Ćwiczenia laboratoryjne:

1. Podstawy pracy zdalnej w architekturze klient-serwer
2. Podstawy pracy zdalnej w połączeniu z wieloma węzłami
3. Problem producentów - konsumentów
4. Implementacja wybranych rozwiązań problemu czytelników i pisarzy
5. Zegary logiczne i komunikacja z wykorzystaniem Java Message Service
6. Konstrukcja drzewa rozpinającego
7. Implementacja wybranych wersji algorytmów wyboru przywódcy w różnych topologiach
8. Rozproszone wykrywanie zakończenia
9. Implementacja wybranego algorytmu wzajemnego wykluczania
10. Blockchain – Implementacja rozproszonego łańcucha bloków
11. Strumieniowe przetwarzanie danych
12. Mechanizmy wykrywania zakleszczeń

Literatura podstawowa:

1. M. Ben-Ari: Podstawy programowania współbieżnego i rozproszonego, WNT, 2009.
2. Zbigniew Weiss, Tadeusz Gruzlewski, Programowanie współbieżne i rozproszone w przykładach i zadaniach, Wydawnictwa Naukowo-Techniczne, 1994.

Literatura dodatkowa:

1. Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing Principles, Algorithms, and Systems, Cambridge University Press, 2008.
2. A.S. Tanenbaum, M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2001.
3. Drescher D., Blockchain. Podstawy technologii łańcucha bloków w 25 krokach, Helion, 2018.

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	Weryfikowany podczas egzaminu. Przykładowe pytanie: „Omów procesy skończone i nieskończone”.
W_02	Weryfikowany podczas egzaminu. Przykładowe pytanie: „Omów wybrany algorytm znajdowania minimalnego drzewa rozpinającego grafu”.
W_03	Weryfikowany podczas egzaminu. Przykładowe pytanie: „W jaki sposób przeprowadzić detekcję zakończenia programu rozproszonego w sieci w topologii pierścienia?”.
W_04	Weryfikowany podczas egzaminu. Przykładowe pytanie: „Wymień i omów rodzaje awarii procesów i łączy komunikacyjnych”.
W_05	Weryfikowany podczas egzaminu. Przykładowe pytanie: „Omów mechanizmy konsensusu w sieci blockchain?”.
U_01	Weryfikowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Zaimplementuj program rozproszony i umieść każdy z procesów na innym komputerze”.
U_02	Weryfikowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Zaimplementuj wybrany mechanizm rozproszonego wyznaczania drzewa rozpinającego”.
U_03	Weryfikowany w trakcie zajęć laboratoryjnych. Przykładowe zadanie: „Napisz system rozproszony w języku Java, który realizuje model producenci-konsumenci”.
K_01	Weryfikowany w oparciu o posiadaną wiedzę i umiejętności w czasie wykonywania zajęć laboratoryjnych w grupach. Studenci będą mieli za zadanie wybrać odpowiedni mechanizm do rozwiązania problemu i ocenić jego skuteczność, a także porównać wyniki z innymi studentami z grupy.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane w trakcie zajęć laboratoryjnych, za które w sumie można uzyskać maksymalnie 110 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu, co najmniej 56 pkt. Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 90 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 46 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 200pkt. po zaokrągleniu do liczby całkowitej) jest następująca (w nawiasach ocena wg skali ECTS): • 0 – 101 pkt: niedostateczna (F), • 102 – 120 pkt: dostateczna (E), • 121 – 140 pkt: dostateczna plus (D), • 141 – 160 pkt: dobra (C), • 161 – 180 pkt: dobra plus (B),	

- 181 – 200 pkt: bardzo dobra (A).

Poprawy:

- Poprawa wybranych ćwiczeń laboratoryjnych. W jednym tygodniu można poprawić maksymalnie 2 ćwiczenia laboratoryjne.
- Uzyskanie poprawkowego zaliczenia laboratoriów możliwe jest w sesji egzaminacyjnej, przed terminem egzaminu pisemnego.

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	17 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Obecność na egzaminie	1 godzina
Przygotowanie się do egzaminu	9 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Obecność na egzaminie	1 godz.
Przygotowanie się do egzaminu	26 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Inżynieria oprogramowania
Nazwa w języku angielskim:		Software engineering
Język wykładowy:	polski	
Kierunek studiów, dla którego przedmiot jest oferowany:		Informatyka
Jednostka realizująca:	Instytut Informatyki	
Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):		obowiązkowy
Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia, jednolitych magisterskich):		pierwszego stopnia
Rok studiów:	Drugi	
Semestr:	Czwarty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Jarosław Skaruz
Imię i nazwisko prowadzących zajęcia:		dr Jarosław Skaruz
Założenia i cele przedmiotu:		Założono, że studenci znają podstawy programowania w dowolnym języku. Celem kursu jest zapoznanie studentów z procesem wytwarzania oprogramowania oraz podstawowymi zagadnieniami związanymi z wytwarzaniem systemów informatycznych.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	zagadnienia z zakresu inżynierii oprogramowania, w tym zna fazy rozwoju oprogramowania i szczegółowy zakres prac realizowanych w każdej fazie.	K_W06
W_02	zagadnienia związane z testowaniem oprogramowania, w tym rodzaje testów.	K_W06
W_03	modele wytwarzania oprogramowania.	K_W06
	UMIEJĘTNOŚCI Student potrafi:	
U_01	pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania	K_U05
U_02	opracować dokumentację dotyczącą realizacji zadania inżynierskiego	K_U08
U_03	posłużyć się właściwie dobranymi środowiskami programistycznymi oraz narzędziami komputerowo wspomaganego projektowania do projektowania i testowania systemów informatycznych	K_U11

U_04	zaprojektować, zaimplementować oraz przetestować prosty system informatyczny	K_U16, K_U20, K_U22
KOMPETENCJE SPOŁECZNE Student jest gotów do:		
K_01	krytycznej oceny posiadanej wiedzy oraz do uznawania znaczenia wiedzy w rozwiązywaniu problemów poznawczych i praktycznych z zakresu informatyki	K_K01
Forma i typy zajęć:		Wykłady (30 godz), ćwiczenia laboratoryjne (30 godz) – studia stacjonarne Wykłady (15 godz), ćwiczenia laboratoryjne (15 godz) – studia niestacjonarne
Wymagania wstępne i dodatkowe:		
• Znajomość podstaw programowania • Umiejętność pracy w wybranym języku programowania i środowisku IDE.		
Treści modułu kształcenia:		
Wykład		
1. Wprowadzenie do inżynierii oprogramowania. 2. Modelowanie biznesowe. Cele i efekty modelowania biznesowego. 3. Modelowanie biznesowe II. Notacja BPMN. 4. Wymagania na system. 5. Analiza i projektowanie I. Cele analizy i projektowania. Forward i backward engineering. 6. Analiza i projektowanie II. Notacja diagramów klas. Ujęcie diagramów w różnych perspektywach. 7. Analiza i projektowanie III. Diagramy czynności, sekwencji i pakietów. 8. Implementacja. Metody efektywnego kodowania: tworzenie i usuwanie obiektów, konstruktory prywatne, unikanie powielania obiektów, wyciek pamięci, interfejsy i klasy abstrakcyjne. 9. Testowanie. Weryfikacja i walidacja. Rodzaje testów. Metody czarnej i białej skrzynki. Inspekcja a testowanie. 10. Projektowanie użytecznych interfejsów użytkownika. 11. Narzędzia inżynierii oprogramowania. Środowiska programowania., Systemy kontroli wersji. Systemy ciągłej integracji. Systemy raportowania błędów 12. Metodyki zarządzania projektami. 13. Wzorce projektowe I. Wprowadzenie do wzorców projektowych. Elementy składowe wzorców. Podział wzorców. Wzorzec Strategia 14. Wzorce projektowe II. Wzorzec Singleton i Dekorator. 15. Zarządzanie ryzykiem.		
Laboratorium		
1. Wprowadzenie 2. Opis procesów biznesowych 3. Modelowanie procesów biznesowych 4. Specyfikacja wymagań 5. Modelowanie za pomoc diagramów przypadków użycia 6. Analiza I 7. Analiza II 8. Analiza III 9. Modelowanie za pomocą diagramu klas 10. Modelowanie za pomocą diagramów sekwencji 11. Schemat relacyjnej bazy danych 12. Implementacja 13. Testowanie oprogramowania 14. Dokumentacja – generowanie raportów z EA 15. Obrona projektów		

Literatura podstawowa:

1. Sommerville I. *Inżynieria oprogramowania*, Wydanie X, 2020, WNT
2. Michał Śmiałek, Kamil Rybiński, *Inżynieria oprogramowania w praktyce. Od wymagań do kodu z językiem UML*, 2023, Helion
3. David Farley, *Nowoczesna inżynieria oprogramowania. Stosowanie skutecznych technik szybszego rozwoju oprogramowania wyższej jakości*, 2023, Helion

Literatura dodatkowa:

1. Git documentation: <https://git-scm.com/doc>
2. Scrum Guide: <https://scrumguides.org>
3. Atlassian tutorials (Jira, Bitbucket)

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01	będą sprawdzane na egzaminie ustnym. Przykładowe pytania: 1. Wymień etapy procesu wytwarzania oprogramowania 2. Wymień artefakty etapu modelowanie biznesowe 3. Wymień artefakty etapu wymagania na system 4. Wyjaśnij na czym polega etap analizy 5. Wyjaśnij na czym polega etap projektowania
W_02	będą sprawdzane na egzaminie ustnym. Przykładowe pytania: 1. Wymień rodzaje testów 2. Opisz sposób realizacji testów jednostkowych 3. Opisz sposób realizacji testów integracyjnych
W_03	będą sprawdzane na egzaminie ustnym. Przykładowe pytania: 1. Opisz metodykę scrum 2. Opisz metodykę kanban 3. Opisz metodykę wodospadku
U_01	będzie sprawdzany poprzez zadanie praktyczne realizowane podczas laboratorium indywidualnie oraz w zespołach
U_02	będzie sprawdzany poprzez realizację zadania polegającego na przygotowaniu szablonu dokumentacji w EA i generowania dokumentacji na podstawie tego szablonu
U_03	będzie sprawdzany poprzez ocenę realizacji zadań z wykorzystaniem Enterprise Architect (EA)
U_04	będzie sprawdzany poprzez zadanie praktyczne realizowane podczas laboratorium indywidualnie oraz w zespołach
K_01	Efekt sprawdzany jest podczas oceny zadań realizowanych w trakcie zajęć laboratoryjnych.

Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składa się ocena za realizację projektu oraz implementacji

prostej aplikacji realizowanej w trakcie zajęć laboratoryjnych. Student może otrzymać maksymalnie 50 pkt za laboratorium. Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów.

Egzamin jest egzaminem ustnym. Można na nim uzyskać do 50 pkt. Egzamin będzie zaliczony w przypadku uzyskania powyżej 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),
- 91 – 100 pkt: bardzo dobra (A).

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	30 godzin
Udział w ćwiczeniach laboratoryjnych	30 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godzin
Udział w konsultacjach godz. z przedmiotu	2 godzina
Przygotowanie się do egzaminu	16 godzin
Obecność na egzaminie	2 godziny
Sumaryczne obciążenie pracą studenta	100 godzin
Punkty ECTS za przedmiot	4

Studia niestacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	15 godzin
Udział w ćwiczeniach laboratoryjnych	15 godzin
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	42 godzin
Udział w konsultacjach godz. z przedmiotu	2 godziny
Przygotowanie się do egzaminu	24 godzin
Obecność na egzaminie	2 godziny
Sumaryczne obciążenie pracą studenta	100 godzin
Punkty ECTS za przedmiot	4

Sylabus przedmiotu / modułu kształcenia			
Nazwa przedmiotu/modułu kształcenia:		Praktyka zawodowa II	
Nazwa w języku angielskim:		Apprenticeship II	
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:	drugi		
Semestr:		czwarty	
Liczba punktów ECTS:	10		
Imię i nazwisko koordynatora przedmiotu:		mgr inż. Maciej Nazarczuk	
Imię i nazwisko prowadzących zajęcia:		Osoba delegowana z firmy/instytucji	
Założenia i cele przedmiotu:		Zakłada się, że student zna podstawowe zasady BHP w firmach i jest zaznajomiony z podstawowymi zasadami organizacji i funkcjonowania firmy. Zna dodatkowo specyfikę rynku pracy i jest zaznajomiony z infrastrukturą technologiczną firmy lub instytucji. Cele praktyki zawodowej II realizują cele praktyki zawodowej I a ponadto: • pogłębienie umiejętności i kompetencji nabytych podczas stażu I • uzupełnienie wiedzy teoretycznej i praktycznej w zakresie użytkowania lub wytwarzania systemów informatycznych; • zapoznanie się ze sposobem funkcjonowania i organizacją wybranej instytucji; • zapoznanie się z techniką prowadzenia dokumentacji na poszczególnych stanowiskach pracy; • poznanie atmosfery pracy oraz zdobycie umiejętności adaptowania się w różnych zespołach ludzkich; • zdobycie specjalistycznej wiedzy związanej z funkcjonowaniem firmy/instytucji w zakresie stosowanych systemów informatycznych • pogłębienie i wykorzystanie w praktyce wiedzy i umiejętności nabytych podczas nauki oraz poprzedniego etapu praktyki zawodowej, • nawiązanie kontaktów zawodowych ułatwiających poszukiwanie pracy,	

		firmy/instytucji do odbycia kolejnych etapów praktyk czy realizacji prac dyplomowych.
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	specjalistyczną wiedzę dziedzinową związaną z działalnością firmy.	K_W13
W_02	systemy i narzędzia informatyczne wykorzystywane w firmie.	K_W13
	UMIEJĘTNOŚCI Student potrafi:	
U_01	wykorzystać wiedzę teoretyczną, praktyczną oraz narzędzia informatyczne do realizacji prostych zadań.	K_U09, K_U10
U_02	na podstawie posiadanej wiedzy, zainteresowań i znajomości rynku pracy potrafi zidentyfikować kierunki dalszego uzupełniania wiedzy i pozyskiwania kolejnych umiejętności oraz doświadczeń zawodowych.	K_U01
U_03	w praktyce stosować zasady bezpieczeństwa i ergonomii pracy.	K_U14
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	krytycznej oceny posiadanej wiedzy, podejmuje starania, aby przekazać informacje i opinie w sposób zrozumiały	K_K01, K_K02
Forma i typy zajęć:		Studia stacjonarne i niestacjonarne: praktyka (320 godz.)
Wymagania wstępne i dodatkowe:		
1. Podstawowa wiedza z zakresu funkcjonowania systemów informatycznych. 2. Wiedza, umiejętności i kompetencje społeczne nabyte podczas poprzedniego etapu praktyki zawodowej.		
Treści modułu kształcenia:		
Szczegółowe treści w zależności od miejsca wykonywania praktyk zawodowych. Podczas drugich praktyk mogą one obejmować między innymi: 1. Zasady bezpieczeństwa i higieny pracy na danym stanowisku oraz uwarunkowania prawne i etyczne stosownie do wykonywanych obowiązków. 2. Specyfika 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, 5. Zapoznanie się z istniejącą infrastrukturą i wykorzystywanymi technologiami informatycznymi w przedsiębiorstwie 6. W miarę możliwości współudział w projektowaniu nowych i ulepszaniu istniejących systemów informatycznych. 7. Prowadzenie dokumentacji przebiegu praktyk.		

Literatura podstawowa:	
Według zalecenia w miejscu odbywania praktyki.	
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_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.
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	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 i weryfikowana przez opiekuna praktyk 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, 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 stażu	270 godz.
Bezpośrednia komunikacja z przełożonymi i członkami zespołu na terenie zakładu pracy – miejscu odbywania stażu	50 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	10
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania stażu	270 godz.
Bezpośrednia komunikacja z przełożonymi i członkami zespołu na terenie zakładu pracy – miejscu odbywania stażu	50 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	10

Sylabus przedmiotu / modułu kształcenia			
Nazwa przedmiotu/modułu kształcenia:		Wprowadzenie do systemów mobilnych	
Nazwa w języku angielskim:		Introduction to Mobile 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:	drugi		
Semestr:		czwarty	
Liczba punktów ECTS:	4		
Imię i nazwisko koordynatora przedmiotu:		dr Grzegorz Terlikowski	
Imię i nazwisko prowadzących zajęcia:		dr Grzegorz Terlikowski, mgr Krzysztof Jacek Gajc	
Założenia i cele przedmiotu:		Zakłada się, że student potrafi programować w C++/Java lub ma zaliczone przedmioty: Podstawy programowania, Programowanie obiektowe, Algorytmy i złożoność, Bazy danych. Celem wykładu jest zapoznanie studentów z problemami przetwarzania mobilnego. W ramach wykładu omówione zostaną najnowsze technologie mobilne i bezprzewodowe, w szczególności sieć telefonii komórkowej - GSM. Wskazana zostanie potrzeba stosowania systemów ruchomych, złożoność problemów w nich występujących jak i sposoby rozwiązania tychże problemów w oparciu o zaadoptowane metody stosowane w innych gałęziach informatyki. W trakcie laboratoriów studenci nabędą umiejętności samodzielnego rozwiązywania praktycznych problemów z zakresu telefonii komórkowej, m.in. z użyciem symulatorów, oraz projektowania i programowania aplikacji mobilnych w systemie Android, z wykorzystaniem środowiska Android Studio.	
Symbol efektu	Efekty uczenia się		Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:		
W_01	strukturę i działanie sieci mobilnych i bezprzewodowych, szczególności sieci komórkowych oraz ideę przetwarzania mobilnego.		K_W02, K_W03, K_W05, K_W07
W_02	mobilne systemy komórkowe oraz architekturę i działanie systemu telefonii komórkowej GSM. Zna także techniczne pojęcia związane z komunikacją mobilną: fali radiowej, efektu tłumienia fal, interferencji oraz efekt Dopplera.		K_W02, K_W03, K_W05, K_W07, K_W14

W_03	współczesne, mobilne systemy dostępu do danych i ich przetwarzania oraz technologie i środowiska wytwarzania aplikacji mobilnych.	K_W02, K_W03, K_W05, K_W07, K_W09, K_W14
	UMIEJĘTNOŚCI Student potrafi:	
U_01	Posługiwać się wybranymi popularnymi aplikacjami, symulatorami odzwierciedlającymi procesy zachodzące w sieciach komórkowych, oraz potrafi korzystać z wybranych środowisk programistycznych i ich bibliotek pod kątem ich wykorzystania w projektowaniu symulatorów sieci komórkowych.	K_U01, K_U02, K_U06, K_U07, K_U08, K_U11, K_U12, K_U15, K_U19, K_U24
U_02	wykonać obliczenia wartości parametrów sieci bezprzewodowych i mobilnych, tj. odległość wykorzystania, zysk i moc anten odbiornika i nadajnika, wpływ efektu Dopplera w zależności od kierunku ruchu stacji mobilnej.	K_U01, K_U02, K_U06, K_U11, K_U15, K_U20,
U_03	utworzyć podstawową aplikację mobilną na urządzenie z systemem Android.	K_U01, K_U06, K_U11, K_U12, K_U15, K_U24
	KOMPETENCJE SPOŁECZNE Student jest gotów do:	
K_01	formułowania opinii na temat zagadnień z zakresu systemów oraz sieci mobilnych i bezprzewodowych, między innymi nt. jakości rozwiązań stosowanych w ich koncepcji i budowie.	K_K01
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
Umiejętność programowania w C++/Java. Zaliczone przedmioty:		
1. Podstawy programowania 2. Programowanie obiektowe 3. Algorytmy i złożoność 4. Bazy danych		
Treści modułu kształcenia:		
Wykład: Wprowadzenie – znaczenie systemów mobilnych. Zapoznanie z ideą przetwarzania mobilnego. Historia rozwoju systemów mobilnych i bezprzewodowych. Podstawy systemów mobilnych. Infrastruktura systemów komórkowych. Systemy satelitarne i sieci ad- hoc. Rozkwit dziedziny oraz czynników wpływających na jej rozwój. Przedstawienie mnogości zastosowań, olbrzymich korzyści dla klienta końcowego oraz złożonych i nietrywialnych problemów, jakie stoją przed projektantami nowoczesnych systemów mobilnych. Pojęcia i definicje. Podstawowe architektury, klasyfikacje terminali. Rozróżnienie między systemami mobilnymi i bezprzewodowymi. Wykazanie cech i elementów złożonego, sieciocentrycznego systemu mobilnego. Propagacja fal radiowych a systemy mobilne. Fale radiowe i ich charakterystyki. Tłumienie fal. Interferencja międzykanałowa. Systemy komórkowe. Idea i potrzeba stosowania systemów komórkowych. Podstawowe pojęcia i definicje. Koncepcja komórki. Parametry komórki. Powtórne wykorzystanie częstotliwości. Formowanie klastrów. Zwalczanie interferencji. Zwiększanie pojemności systemów komórkowych. Omówienie zjawisk typu „roaming” i „handover”. Wady i zalety rozwiązań komórkowych. Architektura i działanie systemu GSM (cz. 1). Podstawowe komponenty systemu GSM, budowa i rodzaje terminali komórkowych, zespoły stacji bazowych, część centralowa. Utrzymywanie informacji o położeniu terminala, zestawianie połączeń. Bezpieczeństwo w systemie GSM, technologie transmisji danych w telefonii komórkowej. Architektura i działanie systemu GSM (cz. 2). Przydział kanałów. Przydział statyczny i dynamiczny. Hybrydowe metody przydziału. Przydział kanałów w systemach specjalizowanych.		

Mobilne systemy komunikacyjne. Infrastruktura systemów komórkowych. Rejestracja i przeniesienie połączenia. Roaming.

6. **Architektura i działanie systemu GSM (cz. 3). Bezpieczeństwo systemów mobilnych.** Techniki szyfrowania. Uwierzytelnienie. Ściany ogniowe i systemy bezpieczeństwa.
7. **Pozycjonowanie i nawigacja użytkowników mobilnych.** Podstawowe pojęcia nawigacyjne, określenie jednostek miar. Sposoby wyznaczania pozycji zliczonej i obserwowanej, urządzenia i systemy pozycjonujące. Nawigacja w budynkach i zintegrowane systemy nawigacyjne. Lokalny charakter informacji pozycyjnej oraz strategie jej uaktualniania. **Systemy nawigacji satelitarnej** typu GNSS: GPS, GLONASS, Galileo, oraz nie będące typu GNSS: Beidou, DORIS i QZSS. Historia powstania, architektura i zasada działania satelitarnych systemów nawigacyjnych.
8. **Systemy chmurowe** (Cloud Computing). Budowa i działanie, rodzaje usług, Klasyfikacja wg budowy i oraz wg. rodzajów podejść do serwowanych usług, bieżące standardy. **Internet rzeczy (IoT).** Koncepcja, budowa, zastosowanie.
9. **Sieci ad hoc i systemy sensorowe.** Charakterystyki systemów ad hoc. Algorytmy i protokoły routingu. Sieci sensorowe. Zastosowania. **Systemy RFID**, zasada działania, rodzaje realizacji transakcji, zastosowania.
10. Przegląd narzędzi do tworzenia biznesowych aplikacji na smartfony.
11. Najnowsza wiedza z zakresu przedmiotu. Współczesne technologie mobilne i bezprzewodowe.

Laboratorium:

1. **Wprowadzenie do laboratorium.** Przedstawienie treści poruszanych w trakcie laboratoriów. Warunki zaliczenia. **Teoretyczne podstawy generowania zdarzeń (ruchu).** Zdarzenia dyskretne. Rozkłady zdarzeń dyskretnych. **Algorytmy i generowanie zdarzeń (ruchu).** Generatory o rozkładach dyskretnych oparte na rozkładzie równomiernym. Generatory o rozkładach ciągłych.
2. **Masowa obsługa zdarzeń.** Modele i charakterystyki. Systemy kolejkowe z pojedynczym kanałem obsługi. Systemy kolejkowe z wielokrotnym kanałem obsługi.
3. **System masowej obsługi (SMO) M/M/S/S.** Analiza procesów zachodzących w SMO. Charakterystyki SMO. Modelowanie systemu.
4. **Koncepcja komórki.** Parametry komórki. Powtórne wykorzystanie częstotliwości. Formowanie klastrów. Zwalczanie interferencji.
5. **Stacja bazowa w sieci komórkowej.** Zadania stacji bazowej. Procesy zachodzące w sieci komórkowej w zasięgu stacji bazowej. Symulator – analiza. **Modelowanie topologii sieci komórkowej.** Powtórne wykorzystanie częstotliwości. Zwalczanie interferencji. Modelowanie systemu sieci komórkowej.
6. **Propagacja fal radiowych w systemach mobilnych.** Tłumienie fal. Efekt Dopplera. Interferencje.
7. **Wprowadzenie do środowiska Android Studio.** Instalacja środowiska i oprogramowania. Obsługa podstawowych narzędzi.
8. **Podstawy tworzenia aplikacji mobilnych w Android Studio.** Zapoznanie z strukturą projektu. Adaptery. Cykl życia aktywności. Utworzenie adaptera. Obsługa podstawowych metod związanych z cyklem życia aplikacji. Przejścia między aktywnościami. Sensory. Obsługa dostępnych sensorów. Pozyskiwanie i przetwarzanie danych z sensorów. Podstawowe obiekty grafiki 2D i zaawansowane mechanizmy do obsługi.
9. **Adaptery. Cykl życia aktywności.** Utworzenie adaptera. Obsługa podstawowych metod związanych z cyklem życia aplikacji. Przejścia między aktywnościami.
10. **Wykorzystanie klasy Intent** do współpracy między aplikacjami. Wyjaśnienie roli manifestu oraz uprawnień.
11. **Sensory.** Obsługa dostępnych sensorów. Pozyskiwanie i przetwarzanie danych z sensorów.
12. Przegląd narzędzi do tworzenia biznesowych aplikacji na smartfony. Najnowsza wiedza z zakresu przedmiotu. Współczesne technologie mobilne i bezprzewodowe.

Literatura podstawowa:

1. D. P. Agrawal, Q. A. Zeng: Introduction to Wireless and Mobile Systems, United States : Cengage Learning, 2016
2. Dmitry Jemerov, Svetlana Isakova: Mastering Kotlin : learn advanced Kotlin programming techniques to build apps for Android, iOS, and the web / Nate Ebel, Birmingham ; Mumbai : Packt 2019
3. K. Smith: Mobile Communications and Networks, Larsen and Keller Education, 2017

Literatura dodatkowa:

1. Android Studio : wygodne i efektywne tworzenie aplikacji / Adam Gerber, Clifton Craig ; [tłumaczenie Rafał Jońca]. - Gliwice : Helion, cop. 2016. - ISBN 978-83-283-2009-3
2. Dmitry Jemerov, Svetlana Isakova: Kotlin w akcji, Helion 2018e Piotr Rajca]. - Gliwice : Wydawnictwo Helion, cop. 2016. - ISBN 978-83-283-2612-5,
3. Android : wprowadzenie do programowania aplikacji / Joseph Annuzzi Jr., Lauren Darcey, Shane Conder ; [tłumaczenie Piotr Rajca]. - Gliwice : Wydawnictwo Helion, cop. 2016. - ISBN 978-83-283-2612-5,
4. J. Januszewski. Systemy satelitarne GPS, Galileo i inne. PWN, 2010

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

Symbol efektu	Metody weryfikacji efektów uczenia się
W_01 – W_03	Egzamin pisemny. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania: • Scharakteryzuj System Mobilny. Wymień jego elementy i krótko opisz. (W_01) • Wykonaj rysunek prawidłowego rozmieszczenia stacji bazowych i uzasadnij. (W_02) • Wymień najważniejsze modele Chmur Obliczeniowych (Cloud Computing). Opisz jeden z nich. (W_03)
U_01- U_03	Na bieżąco, na każdych zajęciach poza pierwszym i ostatnim w postaci zadań praktycznych. Tematyka następnego laboratorium będzie podana tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich przygotować samodzielnie lub korzystając z konsultacji. Przykładowe zadanie (sprawdza efekt: U_01): Na podstawie teorii modelowania systemu (M/M/S/S), dokonaj implementacji symulatora stacji bazowej. Parametry: • Liczba kanałów, • λ - parametr natężenia ruchu w rozkładzie Poissona, • N – średnia długość rozmowy, wartość oczekiwana w rozkładzie Gaussa, • σ - odchylenie standardowe w rozkładzie Gaussa, • Min i Maks- minimalna i maksymalna długość rozmowy, • Długość kolejki, • Czas symulacji.

	Kontrolki: • Graficzne przedstawienie: połączeń w kanałach, liczby obsługowanych, czasu obsługi połączenia, • Czas symulacji. Wyniki: • Wykresy: ρ - Intensywność ruchu, Q - średnia długość kolejki, W – średni czas oczekiwania. • Plik: Parametry symulacji, ρ, Q, W poniżej w kolumnach. Przykładowe zadanie (sprawdza efekt: U_02): Bezprzewodowy odbiornik o średnicy 250cm odbiera sygnał z częstotliwością 20GHz, z nadajnika o mocy 30mW i zysku anteny transmitującej 30dB. Policzyć: zysk anteny odbiornika oraz moc odbiornika, jeśli jest on oddalony od nadajnika o 5km. Przykładowe zadanie (sprawdza efekt: U_03): 1. Utwórz nowy projekt z pustą aktywnością. 2. Prześledź cykl życia aktywności poprzez nadpisanie wszystkich metod z nim związanych. W każdej z metod wypisz informację o jej wywołaniu, spróbuj wywołać wszystkie z nich. Przeanalizuj momenty ich wywołania w celu ewentualnej persystencji danych aplikacji w przyszłości. 3. Dodaj do aktywności 2 przyciski (Add oraz Remove) oraz element typu ListView. 4. Obsłuż przycisk Add. Po jego kliknięciu aplikacja powinna dodawać nowy wiersz do obiektu ListView (item1, item2, item3). 5. Dodaj możliwość usuwania elementów z listy po ich kliknięciu. 6. Zmodyfikuj ListView w głównej aktywności by wyświetlić bardziej złożone informacje na liście. Dodaj wyświetlanie numeru telefonu. 7. Obsłuż rotację urządzenia w celu tymczasowego przechowania danych. Wykorzystaj funkcję onSaveInstanceState(). Zwróć uwagę, kiedy się wykonuje.
K_01	Efekt będzie sprawdzany poprzez wykonanie i wygłoszenie prezentacji dotyczącej najnowszych technologii z zakresu systemów mobilnych.
Forma i warunki zaliczenia:	
Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 50 pkt. Zaliczenie zajęć laboratoryjnych i dopuszczenie do egzaminu jest możliwe po uzyskaniu co najmniej 25 pkt. Egzamin jest egzaminem pisemnym. Można na nim uzyskać do 40 pkt. Osoby, które zaliczyły laboratorium w terminie, na co najmniej 90% możliwych punktów, mogą skorzystać z egzaminu zerowego ustnego. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 20 pkt. Ponadto, zaliczenie wykładu obejmuje przygotowanie i wygłoszenie prezentacji na zadane tematy dotyczące najnowszych technologii z zakresu technologii i systemów mobilnych, za które można uzyskać do 10 pkt. Wykład będzie zaliczony w przypadku uzyskania co najmniej 5 pkt. z prezentacji. Na tej formie zajęć student może maksymalnie uzyskać 50 pkt. Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS): • 0 – 50 pkt: niedostateczna (F),	

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

Bilans punktów ECTS:

Studia stacjonarne

Aktywność	Obciążenie studenta
Udział w wykładach	21 godz.
Udział w ćwiczeniach laboratoryjnych	24 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	27 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Obecność na egzaminie	2 godziny
Przygotowanie się do egzaminu	23 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	32 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu	33 godz.
Obecność na egzaminie	2 godziny
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Programowanie równoległe
Nazwa w języku angielskim:	Parallel 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:	drugi	
Semestr:	czwarty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		dr Anna Wawrzyńczak-Szaban
Imię i nazwisko prowadzących zajęcia:		dr Anna Wawrzyńczak-Szaban mgr Michał Seredyński
Założenia i cele przedmiotu:		Zakłada się, że student posiada umiejętność programowania w C/C++ oraz znajomość systemu Unix/Linux. Celem przedmiotu jest przedstawienie studentom zagadnień związanych z metodami i technikami programowania równoległego
Symbol efektu	Efekty uczenia się	Symbol efektu kierunkowego
	WIEDZA Student zna i rozumie:	
W_01	metody i techniki programowania równoległego, najważniejsze paradygmaty programowania na systemy z pamięcią współdzieloną i rozproszoną.	K_W05, K_W07
W_02	obecny stan oraz najnowsze trendy z zakresu programowania równoległego, zna najnowsze biblioteki i techniki.	K_W07, K_W09
	UMIEJĘTNOŚCI Student potrafi:	
U_01	napisać program z uwzględnieniem technik programowania równoległego na komputery z pamięcią współdzieloną i rozproszoną z uwzględnieniem współczesnych dedykowanych bibliotek.	K_U22
U_02	samodzielnie dokonać dekompozycji programu sekwencyjnego na zadania możliwe do wykonania równoległe oraz sprawdzić jego poprawność	K_U12, K_U22
U_03	porównać efektywność oraz przyspieszenie programów równoległych i wybrać optymalną konfigurację (szybkość działania, koszt itp.)	K_U17, K_U19
U_04	uruchomić program równoległy na superkomputerze z wykorzystaniem dostępnego systemu operacyjnego i systemu kolejkowego.	K_U12, K_U26
Forma i typy zajęć:		studia stacjonarne: wykłady (21 godz.), ćwiczenia laboratoryjne (24 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (15 godz.)
Wymagania wstępne i dodatkowe:		
Umiejętność programowania w C/C++, znajomość systemu Unix/Linux		
Treści modułu kształcenia:		

1. **Wstęp do programowania równoległego.** Przetwarzanie sekwencyjne a równoległe vs rozproszone. Zasoby obliczeń równoległych. Obszar zastosowań programowania równoległego. Prawo Moore'a
2. **Podstawowe modele obliczeń równoległych:** PRAM, model sieciowy. Złożoność czasowa alg. Równoległych. Przyspieszenie. Efektywność. Prawo Amdahla.
3. **Podstawy programowania równoległego.** Klasyfikacja Flynna. O problemach dekompozycji w obliczeniach równoległych. Prawo Amdahla. Tryby komunikacji. Ogólna charakterystyka komunikacji kolektywnej.
4. **Podstawy interfejsu MPI.** Nazwie i typy danych w MPI. Inicjalizacja MPI. Komunikaty i komunikacja blokująca. Procedury i funkcji komunikacji nie blokującej. Algorytm Gra w życie.
5. **Metody komunikacji kolektywnej.** Bariery, rozgłaszanie (Broadcasting), rozproszenie (Scatter), gromadzenie (Gather) danych i redukcja. Wyróżnione zagadnienia matematyczne.
6. **Komunikatory w MPI.** Tworzenie nowych komunikatorów. Stworzenie nowej grupy procesów. Modyfikacja grupy procesów. Wyróżnione zagadnienia równoległe. Sortowanie szybkie i przez scalanie. Problemy całkowania zagadnień wielowymiarowych.
7. **Architektura klastrów i komputery dużej mocy.** Systemy kolejkowania. Uruchamianie i Monitorowanie zadań na klastrze obliczeniowym.
8. **Superkomputery architektury pamięcią dzieloną i interfejs OpenMP (Open Multi-Processing).** Model programowania OMP. Zrównoleglenie kodu za pomocą dyrektywy OpenMP (parallel, for, Section, schedule). Biblioteka bieżącego przetwarzania. Zmienne środowiskowe.
9. **Mechanizmy Synchronizacji w OpenMP.** Dyrektywy synchronizacja wątków: barrier, critical, master, atomic. Obszary chronione i dyrektywy blokowania zmiennych (set_lock, unset_lock).
10. **Style programowania hybrydowego MPI + OpenMP.** Potencjałe wady i zalety łączenia MPI i OpenMP. Techniki programowania hybrydowego.

Literatura podstawowa:

1. Czech Zbigniew J. Wprowadzenie do obliczeń równoległych, PWN, Warszawa, 2, 2013
2. Strona domowa MPI <http://www.mcs.anl.gov/research/projects/mpi>
3. Strona domowa OPENMP <http://www.openmp.org/>.

Literatura dodatkowa:

1. Quinn M.J., Parallel Programming in C with MPI and OpenMP, Mc Graw Hill, 2004
2. Foster, Designing and Building Parallel Programs, Addison-Wesley Publ.Comp., 1995 (wersja online: <http://www.mcs.anl.gov/~itf/dbpp/text/book.html>)
3. Grama, A. Gupta, G. Karypis, V. Kumar, Introduction to Parallel Computing, Pearson Education Limited 2003

Planowane formy/działania/metody dydaktyczne:

Wykład tradycyjny wspomagany technikami multimedialnymi, laboratorium komputerowe wykorzystujące środowisko obliczeń równoległych z wykorzystaniem MPI i OpenMP. Zamieszczanie na stronach internetowych problemów, zadań oraz materiałów ćwiczeniowych.

Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:

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

W_01 – W_02	Efekty będą weryfikowane poprzez kolokwium pisemne w formie testu wielokrotnego wyboru przeprowadzonego na ostatnim wykładzie Przykładowe pytanie testowe: • Współczynnik przyspieszenia bezwzględnego algorytmu równoległego to: a) Iloraz czasu wykonania optymalnej implementacji algorytmu sekwencyjnego dla zadania o wielkości n do czasu wykonania algorytmu równoległego uzyskanego dla n procesorów b) Iloraz czasu wykonania optymalnej implementacji algorytmu równoległego dla zadania o wielkości n na jednym procesorze do analogicznego czasu uzyskanego dla n procesorów c) Iloczyn efektywności programu równoległego dla zadania o wielkości n wykonanego z użyciem n procesorów oraz liczby procesorów d) Iloraz czasu wykonania dowolnej implementacji algorytmu równoległego dla zadania o wielkości n na jednym procesorze do analogicznego czasu uzyskanego dla n procesorów
U_01 -U_04	Efekty sprawdzane będą na zajęciach laboratoryjnych poprzez realizację punktowanych zadań Przykładowe zadanie: • Procesy programowania równoległego w MPI. Napisz kod algorytmu obliczania trójkąta Pascala, uruchom go i sprawdź przyspieszenie,

Forma i warunki zaliczenia:

Przedmiot kończy się zaliczeniem z oceną. Ocena końcowa ustalana jest na podstawie wyników uzyskanych podczas zajęć laboratoryjnych oraz pisemnego kolokwium końcowego z wykładu.

Warunkiem zaliczenia każdej z formy zajęć jest uzyskanie co najmniej 51% możliwych do zdobycia punktów.

Ocena końcowa wyliczana jest jako średnia ważona, w której:

- 60% stanowi suma punktów uzyskanych w trakcie zajęć laboratoryjnych,
- 40% stanowią punkty uzyskane z kolokwium pisemnego z wykładu.

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	13 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.

Przygotowanie się do kolokwium	14 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach laboratoryjnych	15 godz.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	20 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do kolokwium	22 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS