

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:	Wydział Nauk Ścisłych i Przyrodniczych	
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 Grzegorz Terlikowski	
Imię i nazwisko prowadzących zajęcia:	dr Grzegorz Terlikowski dr Marcin Stępnia, Mgr Mateusz Przychodzki	
Założenia i cele przedmiotu:	Celem kursu jest przedstawienie wiedzy dotyczącej systemów wbudowanych i jej praktyczne wykorzystanie w zbudowaniu wybranego systemu mikroprocesorowego na platformie Arduino Uno.	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia dotyczące struktur i funkcjonalności wbudowanych systemów mikroprocesorowych	K_W02, K_W05
W_02	Zna i rozumie zagadnienia z zakresu działania modułów interfejsowych wykorzystywanych we wbudowanych systemach mikroprocesorowych	K_W05
W_03	Zna i rozumie zagadnienia na temat zasad działania różnych protokołów służących do budowy sieci przewodowych i bezprzewodowych m.in. w obszarze automatyki budynkowej i motoryzacji	K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi pracować z dokumentacją wybranego mikrokontrolera	K_U01, K_U06
U_02	Potrafi zaprojektować, zbudować i zaprogramować wbudowany system mikroprocesorowy w oparciu o platformę Arduino Uno	K_U01, K_U18, K_U24

U_03	Potrafi dobrać moduł interfejsowy oraz zrealizować na nim komunikację pomiędzy układami elektronicznymi.	K_U17, K_U21, K_U24
U_04	Potrafi weryfikować poprawność napisanego programu, potrafi dobrać odpowiednie dane testowe	K_U19, K_U22
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 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 assemblera oraz C/C++ na poziomie podstawowym		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 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. 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. Raspberry Pi: Możliwości i zastosowania platformy. Historia i specyfikacje Raspbery Pi. Systemy operacyjne. Przegląd interfejsów (wyposażenia). Symulatory i środowiska programistyczne. 		
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. S. Monk. Elektronika z wykorzystaniem Arduino i Rapsberry 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:

Efekty W_01 – W_03 będą sprawdzane 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 i porównaj funkcje modułu I²C i SPI.

Efekt U_01 - U_04 będą systematycznie sprawdzane na zajęciach oraz podczas obrony projektu indywidualnego. 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. Zaprojektować układ oraz zaimplementować program, który steruje na przemian zapalającymi się diodami..
2. Zapoznać się z funkcjonalnością środowiska Arduino IDE.
3. Zaprojektować układ oraz napisać program, który wykorzystuje układ czasowo-licznikowy.

Efekty K_01 będzie 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 cząstkowe 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. 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:

- Jednorazowa poprawa każdego ćwiczenia 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.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	16 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	8 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	28 godz.
Udział w konsultacjach godz. z przedmiotu	1 godz.

Przygotowanie się do egzaminu i obecność na egzaminie	16 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:	Wydział Nauk Ścisłych i Przyrodniczych	
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 Stępiak	
Imię i nazwisko prowadzących zajęcia:	dr Marcin Stępiak	
Założenia i cele przedmiotu:	<p>Studenci przystępujący do tego przedmiotu powinni znać zasady programowania obiektowego, a także treści z przedmiotu „algorytmy i złożoność”.</p> <p>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 wykrywania, zakończenia i awarii. Na koniec studenci zostają zapoznani z podstawami technologii blockchain.</p>	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego

W_01	Student zna i rozumie podstawowe standardy, pojęcia i modele przetwarzania rozproszonego.	K_W06, K_W07, K_W09
W_02	Zna i rozumie modele systemów rozproszonych i techniki ich programowania.	K_W06, K_W07
W_03	Zna i rozumie zagadnienia z teorii grafów wykorzystywane w przetwarzaniu rozproszonym. Definicje podstawowych pojęć. Wybrane twierdzenia. Struktury grafów mające zastosowanie w programowaniu rozproszonym.	K_W01
W_04	Zna i rozumie zagadnienia komunikacji zachowującej uporządkowanie wiadomości, konstrukcji spójnego obrazu stanu globalnego, detekcji zakleszczenia rozproszonego oraz detekcji zakończenia przetwarzania rozproszonego.	K_W07
W_05	Zna i rozumie algorytmy przetwarzania w zawodnym środowisku rozproszonym, w tym zagadnienia: modelowania i klasyfikacji awarii, konstrukcji niezawodnych kanałów komunikacyjnych, realizacji detektorów awarii, niezawodnej komunikacji grupowej oraz konsensusu rozproszonego i jego zastosowań.	K_W07
W_06	Zna i rozumie technologię blockchain i jej zastosowania.	K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi zbudować aplikację działającą w środowisku rozproszonym.	K_U01, K_U10, K_U11
U_02	Umie wybrać i zaimplementować odpowiednie do problemu i zgodne ze standardami algorytmy rozproszone.	K_U01, K_U10, K_U11, K_U19, K_U20
U_03	Potrafi programować algorytmy i systemy rozproszone w środowisku Java.	K_U10, K_U17, K_U22
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do 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 w tych przedmiotach.

Treści modułu kształcenia:

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. Algorytm Chandy-Lamporta. Algorytm Lai Yanga. Podnoszenie z awarii. Model obiektów rozproszonych.
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 Tarriego. Problem minimalnego drzewa rozpinającego. Zagadnienie enumeracji węzłów. Metoda losowania. Metoda z przywódcą. Algorytm w topologii drzewa.
7. **Wybór przywódcy.** Algorytm Merlina-Segalla. Algorytm potopowy. Algorytm na pierścieniu. Algorytm na drzewie. Algorytmy w sieciach anonimowych.
8. **Rozproszone wykrywanie zakończenia i awarii.** Wykrywanie zakończenia. Algorytm Dijkstra'y-Scholten'a. 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 strategie postępowania z zakleszczeniami. Podejście scentralizowane i rozproszone. Algorytm Chandy-Misra-Haas.
11. **Technologia blockchain.** Omówienie podstawowych zastosowań. Kryptowaluty na przykładzie Bitcoin.

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. Tel G. Introduction to Distributed Algorithms, Cambridge University Press, 2000.
3. A.S. Tanenbaum, M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2001.
4. Lynch N. A. Distributed algorithms, Morgan Kaufmann Publishers, 1996.
5. Guerraoui R., Rodrigues L. Introduction to Reliable Distributed Programming, Springer-Verlag, 2006.
6. Drescher D., Blockchain. Podstawy technologii łańcucha bloków w 25 krokach, Helion, 2018.
7. Dhillon V., Hooper M., Metcalf D. Zastosowania technologii blockchain, Wydawnictwo Naukowe PWN, 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:

Efekty W_01 – W_06 będą sprawdzane na egzaminie pisemnym.

Przykładowe zadania:

- Omów metody synchronizacji zegarów fizycznych w środowiskach rozproszonych.
- W jaki sposób przeprowadzić detekcję zakończenia programu rozproszonego w sieci w topologii pierścienia?
- W jaki sposób można wykrywać zakleszczenia w systemach rozproszonych?
- Jaki jest cel algorytmów rozgłaszania? Podaj przykłady takich algorytmów.
- Znajdź minimalne drzewo rozpinające grafu.

Efekty U_01 – U_03 sprawdzane będą na bieżąco, na każdych zajęciach (poza pierwszymi) poprzez implementacje w środowisku Java algorytmów rozwiązujących zadania praktyczne.

Przykładowe zadanie:

- Napisz system rozproszony w języku Java, który realizuje model producenci-konsumenci, zapewniając poniższe założenia:
 - jeden producent,
 - dwóch konsumentów,
 - bufor ograniczony.

Efekt K_01 będzie 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.

Tematyka zajęć laboratoryjnych zostanie podana około tygodnia przed konkretnymi zajęciami.

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 trakcie konsultacji. 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	14 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	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.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	15 godz.
Udział w konsultacjach godz. z przedmiotu	3 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	27 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:	Wydział Nauk Ścisłych i Przyrodniczych	
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:	4	
Imię i nazwisko koordynatora przedmiotu:	dr Jarosław Skaruz	
Imię i nazwisko prowadzących zajęcia:	dr Jarosław Skaruz	
Założenia i cele przedmiotu:	Celem przedmiotu jest zapoznanie studentów z etapami cyklu rozwoju oprogramowania i szczegółami ich realizacji.	
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia z zakresu inżynierii oprogramowania.	K_W06
W_02	Zna i rozumie etapy w procesie produkcji oprogramowania oraz szczegóły ich realizacji.	K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi pracować indywidualnie i w zespole wytwarzającym oprogramowanie;	K_U05
U_02	Potrafi oszacować czas potrzebny na realizację zleconego zadania	K_U13
U_03	Potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego	K_U08
U_04	Potrafi posłużyć się właściwie dobranym oprogramowaniem wykorzystywanym do zaprojektowania oraz implementacji aplikacji	K_U11

U_05	Potrafi przetestować wykonany program komputerowy	K_U19
U_06	Potrafi zaprojektować i zaimplementować	K_U20, K_U22
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 z zakresu inżynierii oprogramowania	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:		
1 Umiejętność programowania		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> Wprowadzenie do inżynierii oprogramowania. Podstawowe zagadnienia inżynierii oprogramowania. Kryzys oprogramowania. Złożoność projektu oprogramowania. Szacowanie kosztu wytwarzania oprogramowania. Modelowanie biznesowe. Cele i efekty modelowania biznesowego. Rodzaje modelowania biznesowego. Modelowanie biznesowe II. Notacja BPMN. Modele cyklu życia oprogramowania Metodyki wytwarzania oprogramowanie. RUP, Scrum, Kanban. Wymagania na system. Cele wymagań na system. Wymagania funkcjonalne i нефункционалне. Pozyskiwanie i specyfikowanie wymagań. Projektowanie użytecznych interfejsów użytkownika. Cele usability. Wpływ optymalizacji stron WWW. Optymalizacja elementów WWW: nawigacja, wyszukiwarki, teksty, linki i komunikaty. Analiza i projektowanie I. Cele analizy i projektowania. Forward i backward engineering. Cele analizy i realizacja procesu analizy. Analiza i projektowanie II. Notacja diagramów klas. Ujęcie diagramów w różnych perspektywach. Prezentacja atrybutów, metod i związków pomiędzy klasami. Analiza i projektowanie III. Pozostałe wybrane diagramy notacji UML. Implementacja. Metody efektywnego kodowania: tworzenie i usuwanie obiektów, konstruktory prywatne, unikanie powielania obiektów, wyciek pamięci, interfejsy i klasy abstrakcyjne. Testowanie. Weryfikacja i walidacja. Rodzaje testów. Metody czarnej i białej skrzynki. Inspekcja a testowanie. Framework JUnit. Zarządzanie projektem. Metodyki zarządzania projektami, zarządzanie ryzykiem. Narzędzia inżynierii oprogramowania. Systemy kontroli wersji. Systemy ciągłej integracji i statycznej analizy kodu źródłowego. Zarządzanie ryzykiem w projektach informatycznych. Identyfikacja i klasyfikacja ryzyka, analiza ryzyka i metody ich minimalizacji, monitorowanie ryzyka 		
Literatura podstawowa:		
1. FC. Larman, UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji, Helion, 2011		

2. E. Hasted, Sprzedaj swój software, Helion, 2007	
Literatura dodatkowa:	
1. B. K. Jayaswal, P. C. Patton, Oprogramowanie godne zaufania. Metodologia, techniki i narzędzia projektowania, Helion, 2007	
2. D. Pilone, R. Miles, Head First Software Development, Helion, 2008	
Planowane formy/działania/metody dydaktyczne:	
Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia z wykorzystaniem narzędzi CASE. Zamieszczanie na stronach internetowych materiałów dydaktycznych	
Sposoby weryfikacji efektów uczenia się osiągniętych przez studenta:	
<p>Efekty W_01 i W_02 sprawdzane będą na egzaminie w sesji egzaminacyjnej.</p> <p>Efekty U_01-U_06 będą sprawdzane na każdym zajęciach laboratoryjnych oraz podczas oceny wykonanego przez studentów projektu na ostatnich zajęciach. Studenci realizują zadanie indywidualne polegające na utworzeniu oprogramowania na podstawie samodzielnie oraz w grupie opracowanego projektu technicznego oprogramowania. Studenci realizują zadania polegające na utworzeniu modeli z wykorzystaniem języka UML np. diagramów przypadków użycia, klas, pakietów, sekwencji, aktywności, wdrożenia, komponentów.</p> <p>Efekt K1 jest sprawdzany podczas zajęć laboratoryjnych. Studenci zobowiązani są do pracy w grupie, rozwiązując wspólnie zadanie dotyczące zaprojektowania małej części systemu informatycznego z wykorzystaniem języka UML.</p>	
Forma i warunki zaliczenia:	
<p>Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Zaliczenie laboratorium odbywa się na podstawie oceny uzyskanej za wykonanie projektu oraz implementacji małego systemu informatycznego według schematu:</p> <ul style="list-style-type: none"> • Obrona zadania indywidualnego – 50 pkt. <p>Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów.</p> <p>Egzamin jest egzaminem ustnym. Można na nim uzyskać do 30 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 50pkt.) jest następująca (w nawiasach ocena wg skali ECTS):</p> <ul style="list-style-type: none"> • 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.
Samodzielne przygotowanie się do ćwiczeń laboratoryjnych	22 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	12 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	36 godz.
Udział w konsultacjach godz. z przedmiotu	4 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	30 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

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:	Wydział Nauk Ścisłych i Przyrodniczych	
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:	10	
Imię i nazwisko koordynatora przedmiotu:		dr Jarosław Skaruz, Mgr Maciej Nazarczuk
Imię i nazwisko prowadzących zajęcia:		Osoba delegowana z firmy/institucji
Założenia i cele przedmiotu:		<p>Cele praktyki:</p> <ul style="list-style-type: none"> • zdobycie specjalistycznej wiedzy związanej z funkcjonowaniem firmy/institucji w zakresie stosowanych systemów informatycznych • pogłębienie wiedzy i umiejętności związanych z rozwojem istniejących systemów i wytwarzania nowych aplikacji • pogłębienie wiedzy i umiejętności związanych z wybraną specjalnością • pogłębienie i wykorzystanie w praktyce wiedzy i umiejętności nabytych podczas nauki oraz poprzedniego etapu praktyki zawodowej • nawiązanie kontaktów zawodowych ułatwiających poszukiwanie pracy, firmy/institucji do odbycia kolejnych etapów praktyk czy realizacji prac dyplomowych.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zdobywa specjalistyczną wiedzę dziedzinową związaną z działalnością firmy.	K_W13
W_02	Zna systemy i narzędzia informatyczne wykorzystywane w firmie.	K_W13
W_03	Posiada wiedzę umożliwiającą wytwarzanie oprogramowania oraz administrowanie sieciami i systemami informatycznymi.	K_W06, K_W07
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego

U_01	Potrafi wykorzystać wiedzę teoretyczną, praktyczną oraz narzędzia informatyczne do realizacji stosunkowo 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	Potrafi zarządzać infrastrukturą informatyczną. Umie projektować oraz implementować proste aplikacje i systemy informatyczne.	K_U23, K_U24
U_04	Potrafi w praktyce stosować zasady bezpieczeństwa i ergonomii pracy.	K_U14
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	jest gotów do krytycznej oceny posiadanej wiedzy, podejmuje starania, aby przekazać informacje i opinie w sposób zrozumiały	K_K01, K_K02
Forma i typy zajęć:	praktyka (320 godz.)	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> 1. Podstawowa wiedza z zakresu funkcjonowania systemów informatycznych. 2. Wiedza, umiejętności i kompetencje społeczne nabyte podczas poprzedniego etapu praktyki zawodowej. 		
Treści modułu kształcenia:		
<ol style="list-style-type: none"> 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, a zwłaszcza wspierającymi zarządzanie i produkcję. W szczególności należy: <ol style="list-style-type: none"> 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. Zapoznanie się z istniejącą infrastrukturą i wykorzystywanymi technologiami informatycznymi 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. W miarę możliwości współdziałanie w projektowaniu nowych i ulepszaniu istniejących systemów informatycznych, biorąc pod uwagę: <ol style="list-style-type: none"> a. Wymagania i cele stawiane przed systemem informatycznym, b. Politykę bezpieczeństwa oraz procedury organizacyjne dotyczące wykorzystania infrastruktury informatycznej, c. Napotykanne ograniczenia techniczne i biznesowe, d. Zagadnienia związane ze zwrotem kosztów inwestycji. 8. 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:	
Wyrywkowa hospitacja w miejscu praktyki, rozmowa ze studentem, ocena przedstawionej dokumentacji	
Forma i warunki zaliczenia:	
<p>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).</p> <p>Ocena końcowa z przedmiotu, w zależności od sumy uzyskanych punktów (maksymalnie 100 pkt) jest następująca (w nawiasach ocena wg skali ECTS):</p> <p>0 – 50 pkt: niedostateczna (F), 51 – 60 pkt: dostateczna (E), 61 – 70 pkt: dostateczna plus (D), 81 – 80 pkt: dobra (C), 81 – 90 pkt: dobra plus (B), 91 – 100 pkt: bardzo dobra (A).</p>	
Bilans punktów ECTS:	
Studia stacjonarne	
Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki	320 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	10 ECTS
Studia niestacjonarne	
Aktywność	Obciążenie studenta
Udział w zorganizowanej formie pracy na terenie zakładu pracy – miejscu odbywania praktyki	320 godz.
Sumaryczne obciążenie pracą studenta	320 godz.
Punkty ECTS za przedmiot	10 ECTS

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):		pierwszego stopnia
Rok studiów:	drugi	
Semestr:	czwarty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		dr Mirosław Szaban
Imię i nazwisko prowadzących zajęcia:		dr Mirosław Szaban, Mgr Krzysztof Gajc
Założenia i cele przedmiotu:		<p>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.</p>
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Student zna strukturę i działanie sieci mobilnych i bezprzewodowych, szczególności sieci komórkowych oraz ideę przetwarzania mobilnego.	K_W02, K_W03, K_W05, K_W07

W_02	Zna 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	Zna 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
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Posługuje 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	Potrafi 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	Student potrafi utworzyć podstawową aplikację mobilną na urządzenie z systemem Android.	K_U01, K_U06, K_U11, K_U12, K_U15, K_U24
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Potrafi formułować opinie 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:		
1. 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.

2. **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.
3. **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.
4. **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.
5. **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 strategię 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
4. Bieżąca prasa z branży IT
5. Android : wprowadzenie do programowania aplikacji / Joseph Anzuzzi Jr., Lauren Darcey, Shane Conder ; [tłumaczenie Piotr Rajca]. - Gliwice : Wydawnictwo Helion, cop. 2016. - ISBN 978-83-283-2612-5

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 2018
3. Biznes w świecie mobile : jak zaprojektować, wykonać i wypromować aplikację mobilną / Sylwia Żółkiewska, Małgorzata Rycharska, Noemi Gryczko. - Warszawa : Poltext, 2018. - ISBN 978-83-7561-866-2
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:

Efekty W_01 – W_03 sprawdzane będą na egzaminie pisemnym w sesji egzaminacyjnej, jako zagadnienia teoretyczne. Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań. Przykładowe pytania:

- Scharakteryzuj System Mobilny. Wymień jego elementy i krótko opisz.
- Wykonaj rysunek prawidłowego rozmieszczenia stacji bazowych i uzasadnij.
- Scharakteryzuj pojęcia: pasmo, kanały częstotliwościowe i wiązki kanałów.
- Jaka jest różnica pomiędzy szybkim i wolnym tłumieniem sygnału (fast and slow fading)?

Efekty U_01 – U_03 sprawdzane będą na bieżąco, na każdym zajęciach poza pierwszym i ostatnim w postaci zadań praktycznych. Tematyka następnego laboratorium będzie podana tydzień przed zajęciami. Student, na podstawie podanej literatury, musi się do nich samodzielnie lub korzystając z konsultacji przygotować. Przykładowe zadanie:

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 obsłużonych, 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.

Efekt K_01 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	25 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	25 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	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	35 godz.
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):		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:		Celem przedmiotu jest przedstawienie studentom zagadnień związanych z metodami i technikami programowania równoległego	
Symbol efektu	Efekty kształcenia		Symbol efektu kierunkowego
	WIEDZA		
W_01	Ma uporządkowaną wiedzę w zakresie metodyki i technik programowania równoległego, najważniejszych paradygmatów programowania na systemy z pamięcią współdzieloną i rozproszoną.		K_W05, K_W07
W_02	Orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych z zakresu programowania równoległego, zna najnowsze biblioteki i techniki.		K_W07, K_W09
UMIEJĘTNOŚCI			
U_01	Potrafi napisać program z uwzględnieniem technik programowania równoległego na komputery z pamięcią współdzieloną i rozproszoną z uwzględnieniem współczesnych dedykowanych bibliotek.		K_U22
U_02	Posiada umiejętność samodzielnej dekompozycji programu sekwencyjnego na zadania możliwe go wykonania równoległe oraz sprawdzić jego poprawność		K_U12, K_U22
U_03	Potrafi porównać efektywność oraz przyspieszenie programów równoległych i wybrać optymalną konfigurację (szybkość działania, koszt itp.)		K_U17, K_U19
U_04	Potrafi 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++	
Treści modułu kształcenia:	
<ol style="list-style-type: none"> Wstęp do programowania równoległego. Przetwarzanie sekwencyjne a równoległe vs rozproszone. Zasoby obliczeń równoległych. Obszar zastosowań programowania równoległego. Prawo Moore'a Podstawowe modele obliczeń równoległych: PRAM, model sieciowy. Złożoność czasowa alg. Równoległych. Przyspieszenie. Efektywność. Prawo Amdahla. Podstawy programowania równoległego. Klasyfikacja Flynna. O problemach dekompozycji w obliczeniach równoległych. Prawo Amdahla. Tryby komunikacji. Ogólna charakterystyka komunikacji kolektywnej. Podstawy interfejsu MPI. Nazwie i typy danych w MPI. Inicjalizacja MPI. Komunikaty i komunikacja blokująca. Procedury i funkcje komunikacji nie blokującej. Algorytm Gra w życie. Metody komunikacji kolektywnej. Bariery, rozgłaszanie danych (Broadcasting), rozproszenie (Scatter), zgromadzenie (Gather) danych i redukcja. Wyróżnione zagadnienia matematyczne. Intra i inter komunikatory w MPI. Tworzenie nowych komunikatorów. Stworzenie nowej grupy procesów_Modyfikacja grupy procesów. Wyróżnione zagadnienia równoległe. Sortowanie szybkie i przez scalanie. Problemy całkowania zagadnień wielo wymiarowych Architektura klastrów i komputery dużej mocy. Systemy kolejkowania. Uruchamianie i Monitorowanie zadań na klastrze obliczeniowym. 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. Zrównoleglenie kodu obliczania liczby Pi za pomocą trzech różnych algorytmów. Mechanizmy Synchronizacji w OpenMP. Dyrektywy synchronizacja wątków: barrier, critical, master, atomic. Obszary chronione i dyrektywy blokowania zmiennych (set_lock, unset_lock). Przykład rozwiązywanie zagadnienia przetwarzania potokowego w OpenMP. Projektowanie algorytmów współbieżnych i równoległych w języku C/C++. Zrównoleglenia algorytmów sortowania (sortowanie szybkie i przez scalanie). 	
Literatura podstawowa:	
<ol style="list-style-type: none"> Czech Zbigniew J. Wprowadzenie do obliczeń równoległych, PWN, Warszawa, 2, 2013 Strona domowa MPI http://www.mcs.anl.gov/research/projects/mpi Strona domowa OPENMP http://www.openmp.org/. 	
Literatura dodatkowa:	
<ol style="list-style-type: none"> Quinn M.J., Parallel Programming in C with MPI and OpenMP, Mc Graw Hill, 2004 Foster, Designing and Building Parallel Programs, Addison-Wesley Publ.Comp., 1995 (wersja online: http://www.mcs.anl.gov/~itf/dbpp/text/book.html) 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 kształcenia osiągniętych przez studenta:

Efekty W_01 – W_02 będą weryfikowane poprzez kolokwium pisemne na ostatnim wykładzie.

Na kolokwium pisemnym testowym zadania będą dotyczyły wybranych problemów algorytmicznych programowania równoległego, 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

Efekty U_01 -U_04 sprawdzane będą na zajęciach laboratoryjnych. Zadania na następne laboratorium muszą być dostępne tydzień przed zajęciami. przykładowe zadanie:

- Procesy programowania równoległego w MPI. Napisz kod algorytmu obliczania trójkąta Paskala,
- Opracowanie algorytmów równoległych na klaster Agenda: Sortowanie przez scalanie, bąbelkowe..

Forma i warunki zaliczenia:

Moduł kończy się zaliczeniem z oceną. Ocena końcowa jest wystawiana na podstawie zajęć laboratoryjnych i jednego kolokwium pisemnego przeprowadzonego na ostatnim wykładzie. Na zaliczenie laboratorium składają się oceny częściowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 40pkt. Zaliczenie zajęć laboratoryjnych następuje w przypadku uzyskania, co najmniej 21pkt.

Za pisemny kolokwium można na nim uzyskać do 60 pkt. Zaliczenie jest możliwe po uzyskaniu, co najmniej 31 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

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

Poprawy:

Uzyskanie poprawkowego zaliczenia laboratoriów możliwe jest w sesji egzaminacyjnej, odpowiednio przed drugim 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	10 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się i udział w egzaminie	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 egzaminu	22 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Modelowanie i wizualizacja grafiki 3D
Nazwa w języku angielskim:		Modeling and Visualisation 3D Graphics
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:	Drugi	
Semestr:	czwarty	
Liczba punktów ECTS:	3	
Imię i nazwisko koordynatora przedmiotu:		Mirosław Barański
Imię i nazwisko prowadzących zajęcia:		Mirosław Barański
Założenia i cele przedmiotu:		Założono, że studenci po skończonym kursie będą znali ważniejsze (i wybrane) aspekty związane z modelowaniem grafiki 3D, wykorzystywane w praktyce środowiska, ich możliwości. Ma to być wstęp do dalszych samodzielnych studiów w tym zakresie. Celem kursu jest zapoznanie studentów z podstawowymi zagadnieniami związanymi z metodami i narzędziami modelowania i wizualizacji grafiki 3D oraz zdobycie praktycznych umiejętności w implementacji tych algorytmów i metod oraz poznanie wybranych narzędzi informatycznych wspomagających modelowanie i wizualizację.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie zagadnienia z zakresu modelowania obiektów podstawowych obiektów graficznych: krzywe, powierzchnie, figury geometryczne i ich przetwarzania	K_W01, K_W06

Sylabus przedmiotu / modułu kształcenia		
W_02	Zna i rozumie zagadnienia związane z określaniem własności obiektów graficznych	K_W01, K_W02, K_W06
W_03	Zna obecny stan wiedzy oraz orientuje się w najnowszych trendach rozwojowych z zakresu modelowania i wizualizacji grafiki 3D	K_W01, K_W02, K_W06
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł na temat zagadnień związanych z modelowaniem i wizualizacją; a następnie integrować uzyskane informacje, dokonywać ich interpretacji i wyciągać wnioski oraz formułować i uzasadniać opinie	K_U01
U_02	posiada umiejętność samokształcenia się, m.in. w celu podnoszenia kompetencji zawodowych	K_U06
U_03	potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązywania zadań związanych z modelowaniem i wizualizacją oraz wybierać i stosować właściwe metody i narzędzia	K_U01, K_U02, K_U10, K_U16, K_U19
Symbol efektu	Efekt uczenia się: KOMPETENCJE SPOŁECZNE	Symbol efektu kierunkowego
K_01	Jest gotów do krytycznej oceny posiadanej wiedzy z metod, technik i narzędzi z zakresu modelowania i wizualizacji grafiki	K_K01
Forma i typy zajęć:	Studia stacjonarne: wykłady (21 godz.), ćwiczenia (24 godz), Studia niestacjonarne: wykłady (15 godz.), ćwiczenia (15 godz),	
Wymagania wstępne i dodatkowe:		
<ol style="list-style-type: none"> Umiejętność programowania, Umiejętność posługiwania się pojęciami geometrii obliczeniowej w podstawowym zakresie. Znajomość podstaw fizyki, zwłaszcza optyki. 		
Treści modułu kształcenia:		

Sylabus przedmiotu / modułu kształcenia

1. Modelowanie krzywych i powierzchni w modelowaniu obiektów grafiki 3D, metody reprezentacji. Własności, rodzaje, zastosowania. Wyznaczanie powierzchni widocznych.
2. Bryły w modelowaniu obiektów grafiki 3D i metody ich reprezentacji.
3. Wybrane zagadnienia dotyczące światła i barw. Fizyka światła i jej wpływ na modele kolorów. Podstawowe zagadnienia związane z percepcją postrzegania barw przez człowieka.
4. Modele kolorów: Model RGB i CMY, Model HSV, Obliczenia w przestrzeni barw.
5. Modelowanie scen. Podstawowe elementy sceny: obiekty, obserwator i oświetlenie. Proste modele scen. Modele scen uwzględniające wybrane efekty oświetlenia: własne oświetlenie, odbicie, odbicie zwierciadlane, załamanie.
6. Modelowanie scen - metoda ray-tracingu. Podstawy matematyczne metody ray-tracingu, uwzględnianie wybranych efektów geometrycznych w metodzie ray-tracingu. Podstawy metody energetycznej.
7. Tekstura - podstawowe zagadnienia. Pojęcie tekstury, metody generowania tekstury.
8. Modelowanie i wizualizacja grafiki 3D w HTML5.
9. Systemy grafiki komputerowej implementujące wybrane metody modelowania scen - aplikacja 3dsMax, Efekty specjalne dostępne w 3ds max. Elementy programowanie scen w aplikacji 3ds max - ogólna charakterystyka języka programowania.
10. Przegląd wybranych systemów wspomagających modelowanie scen. Modelowanie scen za pomocą: Blender, 3dsMax, Unity.

Literatura podstawowa:

1. Zabrodzki J. i inni : Grafika komputerowa, metody i narzędzia. WNT 1994.
2. Michał Jankowski: Elementy grafiki komputerowej. WNT 2006.
3. James D. Foley: Wprowadzenie do grafiki komputerowej. WNT 2001.
4. Tony Parisi, Aplikacje 3D. Przewodnik po HTML5, WebGL i CSS3, Helion 2014.

Literatura dodatkowa:

1. Kelly L. Murdock, 3ds Max 8. Biblia, Helion 2007,
2. Bogdan Bociek, Blender. Podstawy modelowania, Helion 2014,
3. Mike Geig, Unity. Przewodnik projektanta gier. Wydanie III, Helion 2019,
4. Aleksandra Tomaszewska, Google SketchUp. Ćwiczenia praktyczne, Helion 2009.

Planowane formy/działania/metody dydaktyczne:

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

Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:

Efekt W_01, W_02, W_3 i K_01 będą weryfikowane na kolokwium ustnym w postaci indywidualnych zadań do wykonania. Przykładowe zadania dotyczyć będą problemów poruszanych na wykładzie, np.:

- Wymień i scharakteryzuj elementy sceny występujące w modelowaniu grafiki 3D.
- Klasyczne modele kolorów występujące grafice komputerowej, wymień i scharakteryzuj je.

Efekty U_01, U_02 i U_03 będą weryfikowane na zajęciach laboratoryjnych w postaci zadań. Zadania będą udostępniane co najmniej tydzień wcześniej na stronie WWW.

Forma i warunki zaliczenia:

Sylabus przedmiotu / modułu kształcenia

Moduł kończy się zaliczeniem z oceną. Na zaliczenie składają się: kolokwium ustne i zajęcia laboratoryjne. Na ocenę laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim, za które można uzyskać maksymalnie 60 pkt. Zaliczenie zajęć laboratoryjnych jest możliwe po uzyskaniu co najmniej 30 pkt.

Na kolokwium ustnym można uzyskać do 40 pkt. Kolokwium będzie zaliczone w przypadku uzyskania co najmniej 20 pkt. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS):

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

Poprawy:

Jednorazowa poprawa każdego zajęcia laboratoryjnego. W przypadku obecności usprawiedliwionej można na poprawie uzyskać maksymalnie 100% punktów, a w przypadku obecności nieusprawiedliwionej maksymalnie 80% punktów.

Uwaga: Istnieje możliwość zwolnienia z kolokwium ustnego studentów wyróżniających się na zajęciach laboratoryjnych. Warunkiem koniecznym zwolnienia z kolokwium ustnego jest uzyskanie 90% punktów możliwych do zdobycia w trakcie regularnych zajęć laboratoryjnych. Decyzję o ewentualnym zwolnieniu podejmuje osoba przeprowadzająca egzamin po zasięgnięciu opinii (np. 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	21 godz.
Udział w ćwiczeniach	24 godz.
Samodzielne przygotowanie się do ćwiczeń	15 godz.
Udział w konsultacjach godz. z przedmiotu	5 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	10 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Studia niestacjonarne

Sylabus przedmiotu / modułu kształcenia	
Aktywność	Obciążenie studenta
Udział w wykładach	15 godz.
Udział w ćwiczeniach	15 godz.
Samodzielne przygotowanie się do ćwiczeń	28 godz.
Udział w konsultacjach godz. z przedmiotu	2 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	15 godz.
Sumaryczne obciążenie pracą studenta	75 godz.
Punkty ECTS za przedmiot	3 ECTS

Sylabus przedmiotu / modułu kształcenia		
Nazwa przedmiotu/modułu kształcenia:		Technologie programistyczne aplikacji webowych
Nazwa w języku angielskim:		Web Application Programing Technologies
Język wykładowy:	Angielski	
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:	Drugi	
Semestr:	Czwarty	
Liczba punktów ECTS:	4	
Imię i nazwisko koordynatora przedmiotu:		Dr Dariusz Mikułowski
Imię i nazwisko prowadzących zajęcia:		Dr Dariusz Mikułowski
Założenia i cele przedmiotu:		<p>Do realizacji przedmiotu niezbędne jest aby studenci:</p> <ul style="list-style-type: none"> • posiadali podstawową wiedzę na temat działania sieci Internet, podstawowych języków, technologii i protokołów jakie są w niej wykorzystywane. • Studenci powinni znać model sieci, zasadę działania protokołów Internetowych takich jak FTP czy HTTP oraz podstawową znajomość języków HTML i JavaScript. • Studenci powinni znać wzorzec projektowy MVC i 3-warstwową architekturę aplikacji webowej. • Studenci powinni znać język angielski w zakresie pozwalającym na czytanie, pisanie i wypowiedzanie się na poziomie B2. • Do zajęć niezbędne są stanowiska komputerowe z zainstalowanym na nim J2EE oraz środowiskiem programistycznym STS, IntelliJ Idea. • Celem kursu jest zapoznanie studentów z zagadnieniami związanymi z tworzeniem aplikacji webowych zwłaszcza technologiami służącymi do implementowania frontendowej części takich aplikacji.
Symbol efektu	Efekt uczenia się: WIEDZA	Symbol efektu kierunkowego
W_01	Zna i rozumie podstawowe i zaawansowane konstrukcje programistyczne języków Java, JavaScript, zna standard dostępu do dokumentów webowych	K_W06

	DOM, oraz najpopularniejsze biblioteki i frameworki służące do programowania aplikacji webowych.	
W_02	Zna technologie i frameworki do tworzenia front-endowej i back-endowej części aplikacji webowej takie jak: ThymeLeaf, Angular, Bootstrap, Node.js, React, VUE, zna uniwersalny format do przesyłania obiektów pomiędzy komponentami aplikacji JSON.	K_W12, K_W09
W_03	Zna i rozumie technologie takie jak standard JavaServer Faces oraz Spring Web Flow.	K_W06, K_W12
W_04	Zna i rozumie technologie i frameworki wspomagające tworzenie aplikacji webowych takie jak Spring Boot, JHipster czy Cuba platform.	K_W06, K_W12
Symbol efektu	Efekt uczenia się: UMIEJĘTNOŚCI	Symbol efektu kierunkowego
U_01	Potrafi zaprojektować i zaimplementować z poziomu programu dokumenty HTML i XML wykorzystując model dokumentu DOM i biblioteki korzystające z języków JavaScript i TypeScript.	K_U24, K_U11,
U_02	Potrafi napisać, skompilować i uruchomić na serwerze aplikacje wykorzystujące technologie Spring Web Flow, Spring Boot.	K_U02, K_U11, K_U24,
U_03	Potrafi zaprojektować i zaimplementować aplikację webową z użyciem technologii i frameworków do tworzenia Front-endu takich jak Bootstrap, Angular, Thymeleaf, Reactu VUE.	K_U24, K_U11
U_04	Potrafi zaprojektować i utworzyć aplikację webową przy użyciu frameworka Node.js z dostępem do bazy danych zrealizowanym np. przy pomocy bazy MongoDB i użyciem standardowego formatu przesyłania obiektów JSON.	K_U24, K_U11
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 informatyki	K_K01
K_02	Ma świadomość odpowiedzialności za pracę własną oraz gotowość ponoszenia odpowiedzialności za wspólnie realizowane zadania.	K_K02
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:		
<ol style="list-style-type: none"> Umiejętność programowania w języku Java, projektowania obiektowego i paradygmatu MVC. Znajomość HTML5 i podstaw języka JavaScript. 		
Treści zajęć		

<ol style="list-style-type: none"> 1. Skryptowy język programowania po stronie klienta JavaScript i biblioteka jQuery: manipulowanie drzewem dokumentu DOM, zaawansowana obsługa zdarzeń, tworzenie animacji, obsługa asynchronicznych zapytań AJAX, 2. obsługa asynchronicznych zapytań AJAX w języku JavaScript, 3. Technologie do tworzenia front-endowej części aplikacji webowej Angular, Bootstrap 4. Frameworki do tworzenia aplikacji webowych – Node.js i baza danych MongoDB. 5. Technologie i frameworki do tworzenia aplikacji webowych: React, VUE, Amber. 6. Przepływ informacji w aplikacji webowej: zapoznanie z technologią Spring Web Flow 7. Koncepty umiędzynarodawiania aplikacji: umiędzynarodowienie i inkulturacja w języku Java, JSP, Javascript 8. Przegląd alternatywnych rozwiązań do tworzenia systemów internetowych, SpringBoot, JHipster, czy Cuba Platform. 9. Aplikacje wykorzystujące usługi webowe w architekturze REST oparte o koncepcję mikroserwisów.
Literatura podstawowa:
<ol style="list-style-type: none"> 1. Yakov Fain, Anton Moiseev, Angular. Programowanie z użyciem języka TypeScript. Wydanie II (ebook), wyd. polskie Helion Gliwice 2019. 2. Jeff Dickey, Nowoczesne aplikacje internetowe. MongoDB, Express, AngularJS, Node.js, Helion Gliwice 2016.
Literatura dodatkowa:
<ol style="list-style-type: none"> 1. Podręcznik Node.js : smashing magazine / Guillermo Rauch ; [tł. Krzysztof Wołowski] 2. AngularJS : tworzenie aplikacji webowych : receptury / Matt Frisbie ; [tłumaczenie Radosław Meryk], Gliwice Helion 2016.
Planowane formy/działania/metody dydaktyczne:
<p>Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia programistyczne laboratoryjne, Zamieszczanie na stronach internetowych problemów i zadań ćwiczeniowych, prezentowanie zagadnień przez studentów.</p>
Sposoby weryfikacji efektów kształcenia osiągniętych przez studenta:
<p>Podczas ćwiczeń laboratoryjnych sprawdzane będą efekty U_01, U_02, U_03 i U_04 oraz K_01 i K_02. Podczas kolokwium zaliczeniowego sprawdzane będą efekty W_01, W_02, W_03, W_04.</p> <p>Przykładowe pytanie z kolokwium: (test wyboru):</p> <p>Funkcjami dostępnymi w bibliotece jQuery są:</p> <p>odp 1: click(), text(), html(), val().</p> <p>odp 2: toggle(), append(), attr(), removeClass().</p> <p>odp 3: draw(), addElement(), prepend(), hide().</p> <p>odp 4: resizeWindow(), insert(), before(), after().</p> <p>Przykładowe zadanie z laboratorium:</p> <p>Utwórz stronę internetową składającą się z 3 widoków o nazwach: kontakt, więcej o mnie i moja praca wykorzystując mechanizm routingu dostępny we frameworku Angular.</p>
Forma i warunki zaliczenia:

Moduł kończy się egzaminem. Ocena końcowa jest wystawiana na podstawie ocen cząstkowych uzyskanych na każdym zajęciach laboratoryjnych, i egzaminu pisemnego (który może mieć formę testu w wersji elektronicznej). Za zajęcia laboratoryjne można uzyskać maksymalnie 120pkt, a za egzamin - maksymalnie 30 pkt. Daje to łącznie 150pkt. Zaliczenie zajęć następuje w przypadku uzyskania co najmniej 61 pkt za zadania wykonywane podczas laboratorium oraz co najmniej 16 punktów za egzamin.

Ocena końcowa z modułu (po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 150pkt.) może być następująca (w nawiasach ocena wg skali ECTS):

- 0-75 pkt. ndst (F)
- 76-91 pkt. dst (E)
- 92-107 pkt. dst+ (D)
- 108-121 pkt. db (C)
- 122-137 pkt. db+ (B)
- 138-150 pkt. Bdb (A)

Warunek uzyskania zaliczenia przedmiotu: co najwyżej dwie nieusprawiedliwione nieobecności na ćwiczeniach i spełnienie każdego z niżej opisanych warunków

1. uzyskanie co najmniej 61 punktów z ćwiczeń
2. uzyskanie co najmniej 16 punktów z kolokwium(testu)

Sposób uzyskania punktów:

1. ćwiczenia laboratoryjne: 120 pkt. (za każde z 12 zajęć do 4,5 pkt.)
2. Egzamin / test elektroniczny: 30 pkt

Poprawy:

- Jednorazowa poprawa egzaminu (e-testu) w sesji.
- Obrony do 3 nie wykonanych ćwiczeń laboratoryjnych w trakcie semestru (za obronione zadanie można otrzymać maksymalnie 50% punktów.

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	34 godz.
Udział w konsultacjach godz. z przedmiotu	6 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	15 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	45 godz.
Udział w konsultacjach godz. z przedmiotu	4 godz.
Przygotowanie się do egzaminu i obecność na egzaminie	21 godz.
Sumaryczne obciążenie pracą studenta	100 godz.
Punkty ECTS za przedmiot	4 ECTS

