



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy i aplikacje bez granic (ubiquitous) [N1Inf1>UBI]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
3/6

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
niestacjonarne

Wymagalność
obieralny

Liczba godzin

Wykład
16

Laboratorium
16

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Bartłomiej Prędkie
bartlomiej.predki@put.poznan.pl

Wykładowcy

Wymagania wstępne

Presented outcomes are verified as follows: Forming degree: a) on lectures - based on the answers concerning material presented on previous lectures; b) on laboratories - based on the fulfillment of current tasks, Summary degree: - verification of skills used in laboratory exercises, - constant verification in classes - verification of knowledge and skill acquisition, - written test consisting of 10-15 questions; to pass the test student has to obtain at least 50% of correct answers. Additional point obtained in classes, especially: - demonstration of interesting extracurricular competences, - presentation of additional problem aspects, - doing a presentation on interesting subject concerning ubiquitous systems, - efficacy of obtained knowledge use while solving a problem, - ability to work in team, - useful remarks concerning teaching materials.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy na temat historii i funkcjonowania systemów powszechnych i mobilnych. 2. Przekazanie studentom wiedzy związanej z historią i rozwojem platform mobilnych, w szczególności iOS. 3. Zapoznanie studentów z praktykami wytwarzania aplikacji mobilnych, środowiskiem do ich tworzenia, zasadami działania SDK i dystrybucji aplikacji. 4. Kształtowanie u studentów umiejętności pracy zespołowej w trakcie realizacji projektu na zajęciach laboratoryjnych

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie tzw. systemów i aplikacji bez granic .
2. Ma wiedzę o istotnych kierunkach rozwoju i najważniejszych osiągnięciach informatyki w zakresie tzw. systemów i aplikacji bez granic.
3. Zna podstawowe techniki, metody oraz narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych w zakresie tzw. systemów i aplikacji bez granic.
4. Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych, systemów operacyjnych.

Umiejętności:

1. Pozyskiwać informacje z zakresu tzw. systemów i aplikacji bez granic z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie.
2. Potrafi odpowiednio posługiwać się technikami informacyjno-komunikacyjnymi, znajdującymi zastosowanie na różnych etapach realizacji przedsięwzięć informatycznych z zakresu tzw. systemów i aplikacji bez granic.
3. Potrafi, formułując i rozwiązując zadania informatyczne z zakresu tzw. systemów i aplikacji bez granic, zastosować odpowiednio dobrane metody, w tym eksperymentalne.
4. Potrafi zaprojektować system informatyczny z zakresu tzw. systemów i aplikacji bez granic, dobierając język programowania do danego zadania programistycznego z zakresu tzw. systemów i aplikacji bez granic oraz używając właściwych metod, technik i narzędzi.
5. Ma umiejętność formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi stosowanych w tzw. systemach i aplikacjach bez granic.
6. Potrafi planować i realizować proces własnego permanentnego uczenia się oraz zna możliwości dalszego doksztalcania się w zakresie tzw. systemów i aplikacji bez granic.

Kompetencje społeczne:

1. Rozumie, że w informatyce wiedza i umiejętności z zakresu tzw. systemów i aplikacji bez granic bardzo szybko stają się przestarzałe.
2. Ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich z zakresu tzw. systemów i aplikacji bez granic oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach;

b) w zakresie laboratoriów:

- na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- ocenianie ciągle, na każdych zajęciach (ocena wykonanych na zajęciach projektów) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
- ocenę wiedzy i umiejętności wykazanych na teście pisemnym w formie kilkunastu pytań, do zaliczenia testu potrzebne jest minimum 50% poprawnych odpowiedzi.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- wykazanie się ciekawymi umiejętnościami ponadprogramowymi,
- omówienia dodatkowych aspektów zagadnienia,
- przygotowanie opracowania na określony temat,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,

- uwagi związane z udoskonaleniem materiałów dydaktycznych

Treści programowe

Kurs przedstawia problematykę wytwarzania aplikacji mobilnych dla systemu iOS z praktycznymi przykładami ilustrującymi wybrane techniki.

Tematyka zajęć

W ramach wykładu przedstawiane są następujące zagadnienia:

- wprowadzenie teoretyczne do systemów i aplikacji bez granic,
- programowanie na platformie iOS i iPadOS, a w szczególności:
- podstawy języka programowania Swift i ogólne informacje o Objective-C,
- programowanie z wykorzystaniem różnorodnych bibliotek (UIKit, SwiftUI, SpriteKit) dla poszczególnych platform sprzętowych - iPhone, iPad, Apple Watch,
- programowanie z wykorzystaniem różnorodnych API, takich jak: CoreData, SwiftData, CoreLocation, itp.,
- protokoły wymiany danych, np. JSON, REST,

W ramach ćwiczeń laboratoryjnych studenci będą mogli w praktyce przećwiczyć materiały prezentowane na wykładach w formie mini-projektów obejmujących najczęściej jedno zajęcia laboratoryjne. Laboratoria są przygotowane w ten sposób, że nie wymaga się od studenta posiadania/dostępu do żadnych urządzeń z systemami iOS/macOS. Zajęcia odbywają się na komputerach mac mini, a całość programu realizowana jest wyłącznie w laboratorium - nie przewiduje się pracy domowej.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, dyskusja i analiza problemów.
2. Ćwiczenia laboratoryjne: wykonywanie eksperymentów, rozwiązywanie zadań, dyskusja, praca w zespole

Literatura

1. Podstawy języka Swift : programowanie aplikacji dla platformy iOS, Mark A. Lasso, Helion 2016
2. iOS 12 : wprowadzenie do programowania w Swiftie, Matt Neuburg, Helion 2019
3. Objective-C : praktyczny podręcznik tworzenia aplikacji na systemy iOS i Mac OS X!, Stephen G. Kochan, Helion 2012
4. Poznaj Swifta, tworząc aplikacje : profesjonalne projekty dla systemu iOS, Emil Atanasov, Helion 2019
5. Service design patterns: fundamental design solutions for SOAP/WSDL and RESTful Web services, Robert Daigneau, Addison-Wesley, 2012
6. Inteligentny dom: automatyzacja mieszkania za pomocą platformy Arduino, systemu Android i zwykłego komputera / Mike Riley, Helion 2013
7. Android : programowanie aplikacji / Dawn Griffiths, David Griffiths, Helion 2016

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	41	1,50