



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Projektowanie aplikacji mobilnych z elementami SI [S2SI1E>PAM]

Przedmiot

Kierunek studiów

Sztuczna inteligencja/Artificial Intelligence

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

15

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

15

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Bartłomiej Prędko

bartlomiej.predki@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu funkcjonowania komputera i programowania imperatywnego (zdobytą na zajęciach z przedmiotów Wprowadzenie do informatyki i Podstawy programowania) oraz wybranych elementów sieci komputerowych i sztucznej inteligencji, w szczególności uczenia maszynowego. Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu implementacji i oceny kosztu działania prostych algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien posiadać podstawowe umiejętności z tematyki projektowania interfejsu użytkownika. Powinien również rozumieć konieczność poszerzania swoich kompetencji i wykazywać gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu

Celem przedmiotu jest: 1. Przekazanie studentom podstawowej wiedzy na temat historii i rozwoju systemów mobilnych. 2. Student posiada wiedzę dotyczącą aktualnego rozwoju platform mobilnych i trendów na przyszłość. 3. Pozyskanie przez studentów umiejętności projektowania i programowania aplikacji mobilnych. 4. Pozyskanie przez studentów podstawowych umiejętności związanych z integracją modeli uczenia maszynowego w aplikacjach mobilnych. 5. Kształtowanie u studentów wiedzy dotyczącej architektury systemów komputerowych i aplikacji.

Przedmiotowe efekty uczenia się

Wiedza 1. Ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie systemów mobilnych.
2. Ma wiedzę o istotnych kierunkach rozwoju i najważniejszych osiągnięciach informatyki w zakresie systemów mobilnych.

3. Zna podstawowe techniki, metody oraz narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych w zakresie projektowania i implementacji aplikacji mobilnych .

4. Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych, systemów operacyjnych i aplikacji mobilnych.

Umiejętności 1. Pozyskiwać informacje z zakresu aplikacji mobilnych z literatury, baz danych oraz innych źródeł (w języku 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 aplikacji mobilnych potrafi dobrać odpowiednie narzędzia i metody.

4. Potrafi zaprojektować aplikację mobilną, dobierając język programowania do danego zadania programistycznego oraz używając właściwych metod, technik i narzędzi.

5. Ma umiejętność formułowania algorytmów i ich implementacji z wykorzystaniem dominujących języków programowania (Swift i Kotlin).

6. Potrafi poprawnie użyć wybraną metodę szacowania pracochłonności wytwarzania oprogramowania, ze szczególnym naciskiem na metody sztucznej inteligencji

7. 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 (odpowiedzi ustne) - 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 oraz zadanie projektowe,
- uwagi związane z udoskonaleniem materiałów dydaktycznych

Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

- wprowadzenie do języka Swift,
- programowanie dla platformy iOS i iPadOS z wykorzystaniem środowiska Xcode,
- podstawy języka Kotlin,
- programowanie na platformie Android z wykorzystaniem środowiska Android Studio,
- programowanie z wykorzystaniem różnorodnych API,
- korzystanie z usług chmurowych, np. Dropbox, Google Drive, iCloud, OneDrive z poziomu API aplikacji,
- protokoły wymiany danych, np. JSON, REST,

W ramach ćwiczeń laboratoryjnych studenci będą mogli w praktyce przećwiczyć materiały prezentowane na wykładach w ramach zajęć projektowych studenci będą realizować zamknięty projekt programistyczny dla wybranej platformy.

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
3. Projekt - studenci wykonują indywidualnie zadanie programistyczne

Literatura

1. iOS 5: programowanie: receptury / Vandad Nahavandipoor ; [tł.: Robert Górczyński], Helion 2013.
2. Tworzenie aplikacji na platformę iOS 5 : z wykorzystaniem Xcode, Interface Builder, Instruments, GDB oraz innych kluczowych narzędzi, Brandon Alexander, J. Bradford Dillon, Kevin Y. Kim, Helion, 2012
3. Objective-C : praktyczny podręcznik tworzenia aplikacji na systemy iOS i Mac OS X!, Stephen G. Kochan, Helion 2012
4. Podstawy języka Swift: programowanie aplikacji dla platformy iOS / Mark A. Lassoﬀ & Tom Stachowitz, Helion 2016
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	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	40	2,00