



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Zaawansowane programowanie terminali mobilnych [S2EiT1-TMiB>ZPTM]

Przedmiot

Kierunek studiów

Elektronika i telekomunikacja

Rok/Semestr

1/2

Studia w zakresie (specjalność)

Technologie mobilne i bezprzewodowe

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

15

Laboratorium

30

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

15

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Paweł Sroka

pawel.sroka@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać pogłębioną wiedzę w zakresie programowania, w szczególności zorientowanego obiektowo, z wykorzystaniem języków takich jak: C++, C# czy Java. Dodatkowo student powinien mieć podstawową wiedzę na temat funkcjonowania systemów operacyjnych i baz danych, a także znać podstawy architektury i sposobu działania mikroprocesorowych układów programowalnych. Wymagana jest też wiedza na temat działania nowoczesnych systemów telekomunikacyjnych (przewodowych i bezprzewodowych). Dodatkowo student powinien umieć zaimplementować zaawansowane algorytmy z wykorzystaniem wybranych języków programowania, posiadać umiejętność wyszukiwania rozwiązań problemów programistycznych korzystając z różnych źródeł, a także być gotowym do podjęcia współpracy w ramach zespołu. Student powinien być świadomy swoich umiejętności, ograniczeń, a także konieczności ciągłego kształcenia się. Powinien też rozumieć znaczenie profesjonalnego podejścia do realizowanego zadania i odpowiedzialności za opracowane rozwiązania.

Cel przedmiotu

Celem przedmiotu jest przekazanie studentom wiedzy na temat zasad i sposobów programowania urządzeń mobilnych z systemem iOS, a także rozwinięcie ich umiejętności programistycznych dla potrzeb tworzenia aplikacji dla systemu iOS. Po ukończeniu przedmiotu student będzie potrafił zaprojektować i zaimplementować aplikację o określonej funkcjonalności dla systemu iOS korzystając z dostępnych zasobów i narzędzi, przeprowadzić testy aplikacji, a także przygotować ją do publikacji w sklepie internetowym.

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma ugruntowaną wiedzę w zakresie programowania urządzeń mobilnych.
2. Posiada wiedzę na temat sposobów użycia różnych układów i zasobów dostępnych w urządzeniach mobilnych.
3. Wie jak wykorzystać różne narzędzia oraz biblioteki programistyczne w celu poszerzenia funkcjonalności aplikacji.

Umiejętności:

1. Umie korzystać z różnych zasobów i narzędzi dostępnych w Internecie.
2. Potrafi zaprojektować i zaimplementować w pełni funkcjonalną aplikację oraz przygotować opisującą ją dokumentację.
3. Potrafi sprawdzić poprawność implementacji aplikacji korzystając z mechanizmów testowania.

Kompetencje społeczne:

1. Jest świadomy swojej wiedzy i umiejętności, a także związanych z tym ograniczeń. Rozumie konieczność dalszego kształcenia się związaną z szybkim starzeniem się wiedzy i umiejętności z zakresu tworzenia aplikacji dla iOS.
2. Jest świadomy konieczności profesjonalnego podejścia do realizowanych zadań.
3. Jest świadomy odpowiedzialności za stworzone rozwiązania i aplikacje.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w trakcie wykładów jest weryfikowana podczas egzaminu, który może mieć formę pisemną lub ustną. Egzamin pisemny składa się 6-10 pytań (testowych i/lub otwartych), które mogą być różnie punktowane. Próg zaliczeniowy dla egzaminu pisemnego to 45% możliwych do zdobycia punktów. Egzamin ustny składa się z 3 pytań związanych z tematyką wykładów, które oceniane są w skali 2-5 uwzględniając rozumienie zagadnienia przez studenta, a także szczegółowość odpowiedzi. Próg zaliczeniowy dla egzaminu ustnego to 50% pozytywnie ocenionych odpowiedzi na pytania.

Umiejętności nabyte podczas laboratorium są weryfikowane podczas realizacji 4-7 ćwiczeń praktycznych polegających na implementacji aplikacji o określonej funkcjonalności, zgodnie z przygotowaną instrukcją do zadania. Każde z ćwiczeń jest oceniane w postaci punktów na podstawie przygotowanego sprawozdania, przy czym liczba punktów odpowiadająca zadaniu jest zależna od jego stopnia skomplikowania. Dodatkowo wpływ na ocenę końcową może też mieć ocena pracy studenta na zajęciach. Ocena końcowa zależna jest od liczby zdobytych punktów, przy czym próg zaliczeniowy wynosi 50% maksymalnej liczby punktów za wszystkie ćwiczenia.

Umiejętności i kompetencje nabyte podczas realizacji zajęć projektowych są oceniane na podstawie realizacji obszernego zadania - projektu aplikacji o tematyce uzgodnionej ze studentem. Zadania mogą być realizowane indywidualnie lub w grupach maksymalnie 3-osobowych. Ostateczna ocena z projektu, w skali 2-5, zależna jest od stopnia skomplikowania zadania, zrealizowanych etapów/funkcjonalności, a także zaangażowania studenta w realizację projektu.

Treści programowe

Na wykładach omawiane są następujące zagadnienia:

- Wprowadzenie do programowania urządzeń iOS i języka programowania Swift.
- Rodzaje, rola i działanie kontrolerów widoków w aplikacji iOS.
- Tworzenie interfejsu użytkownika aplikacji iOS. Dynamiczna adaptacja interfejsu aplikacji oraz obsługa gestów.
- Trwały zapis danych w urządzeniach iOS.

- Powiadomienia i ich rola w aplikacji iOS.
- Wielowątkowość w aplikacji iOS.
- Dodatkowe funkcje aplikacji iOS: pobieranie informacji o lokalizacji, użycie map, publikowanie aplikacji w sklepie internetowym.

Tematyka laboratorium przedstawia się następująco:

- Tworzenie prostej aplikacji z podstawowym interfejsem użytkownika.
- Wprowadzenie do implementacji elementów graficznych.
- Implementacja aplikacji wielostronicowej.
- Trwały zapis danych.
- Implementacja adaptacyjnego interfejsu użytkownika oraz obsługa gestów.
- Testowanie aplikacji iOS: testy jednostkowe i integracji.
- Wykorzystanie map oraz informacji o lokalizacji w aplikacji.
- Tworzenie i obsługa powiadomień w iOS.

Tematyka projektu obejmuje stworzenie wielostronicowej aplikacji z adaptacyjnym interfejsem użytkownika, wykorzystującej zaawansowane funkcje dostępne w bibliotekach dla iOS (np. współdziałanie z innymi aplikacjami, trwały zapis danych, wykorzystanie układów elektronicznych dostępnych w urządzeniu) oraz sprawdzenie jej działania przy użyciu symulatora bądź urządzenia fizycznego z systemem iOS.

Metody dydaktyczne

Wykłady: prezentacja multimedialna.

Laboratorium: ćwiczenia laboratoryjne - implementacja aplikacji o określonej funkcjonalności w sposób opisany w instrukcji z wykorzystaniem komputerów Mac zawierających odpowiednie oprogramowanie i symulator urządzeń iOS.

Projekt: Realizacja zadania projektowego - aplikacji o określonej funkcjonalności z wykorzystaniem komputerów Mac zawierających odpowiednie oprogramowanie i symulator urządzeń iOS, a także ewentualnych dodatkowych narzędzi dostępnych w Internecie. Możliwa jest grupowa realizacja zadania.

Literatura

Podstawowa

Mark A. Lasso, Tom Stachowitz (tł. Robert Górczyński), "Podstawy języka Swift : programowanie aplikacji dla platformy iOS", Helion, 2016.

Matt Neuburg (tł. Robert Górczyński), "iOS 12 : wprowadzenie do programowania w Swifcie", Helion, 2019.

Uzupełniająca

N.Smyth, "iOS 10 App Development Essentials", CreateSpace Independent Publishing Platform, 2016

<https://developer.apple.com>

<https://www.appcoda.com/>

<https://www.raywenderlich.com/>

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	70	3,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	30	1,00