

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Technologie programistyczne		Kod 1010512321010510611
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność Inteligentne systemy wspomagania decyzji	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: - Projekty/seminaria: 45		Liczba punktów 5
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) inny		(ogólnouczelniany, z innego kierunku) ogólnouczelniany
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 5 100% 5 100%
Odpowiedzialny za przedmiot / wykładowca: dr hab. inż. Szymon Wilk email: Szymon.Wilk@cs.put.poznan.pl tel. 61 6652930 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań dr inż. Bartłomiej Prędkie email: Bartlomiej.Predki@cs.put.poznan.pl tel. 61 6652932 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu programowania oraz popularnych technologii programistycznych (np. Java) prezentowaną w ramach wcześniejszych przedmiotów.
2	Umiejętności:	Student powinien posiadać umiejętność projektowania i implementowania prostych systemów informatycznych oraz pozyskiwania informacji ze wskazanych źródeł. Student powinien rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu.
3	Kompetencje społeczne	Student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi, umiejętność pracy grupowej.
Cel przedmiotu:		
1. Przekazanie studentom podstawowej wiedzy z informatyki, w zakresie nowoczesnych technologii programistycznych służących do budowy zaawansowanych systemów informatycznych (np. mobilnych, wykorzystujących techniki uczenia maszynowego) oraz w zakresie projektowania i implementacji tego typu systemów. 2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z oceną i wyborem odpowiednich technologii programistycznych dostosowanych do specyfiki, z projektowaniem i implementacją zaawansowanych systemów informatycznych oraz z przygotowaniem prezentacji i sprawozdań opisujących wykonane prace. 3. Kształtowanie u studentów umiejętności pracy zespołowej związanej z realizacją projektów programistycznych w grupach wieloosobowych (2-3 osoby).		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma zaawansowaną i pogłębioną wiedzę z zakresu technologii, narzędzi i środowisk programistycznych wykorzystywanych do implementacji systemów informatycznych (w szczególności systemów mobilnych i inteligentnych) - [K2st_W1] 2. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki z zakresu technologii programistycznych - [K2st_W4]		
Umiejętności:		

1. potrafi pozyskiwać informacje z zakresu technologii programistycznych z różnych źródeł (publikacje, zasoby internetowe), właściwie je integrować i interpretować - [K2st_U1]
2. potrafi ocenić przydatność wybranych technologii programistycznych do rozwiązania zadania inżynierskiego, polegającego na budowie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych technologii - [K2st_U9]
3. potrafi ? zgodnie z zadaną specyfikacją ? zaprojektować i zaimplementować system informatyczny używając właściwych technologii i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]
4. potrafi przygotować i przedstawić opracowanie w języku polskim i angielskim, przedstawiające wyniki procesu implementacji systemu informatycznego ze szczególnym uwzględnieniem wykorzystanych technologii - [K2st_U13]
5. potrafi współdziałać w zespole, przyjmując w nim różne role - [K2st_U15]
Kompetencje społeczne:
1. rozumie, że w informatyce wiedza i umiejętności dotyczące technologii programistycznych bardzo szybko stają się przestarzałe - [K2st_K1]
2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu technologii programistycznych w rozwiązywaniu problemów praktycznych - [K2st_K2]

Sposoby sprawdzenia efektów kształcenia

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 projektów:

- na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane poprzez:

- ocenę wiedzy i umiejętności wykazanych na sprawdzianie pisemnym obejmującym zarówno pytania otwarte, jak i test wielokrotnego wyboru. Sprawdzian obejmuje 10 pytań, punktacja wyrażona jest procentowo, przy czym do uzyskania oceny 3.0 konieczne jest uzyskanie 51% punktów. Podczas sprawdzianu nie można korzystać z żadnych materiałów dydaktycznych,

- ocenę prezentacji przedstawionej na wykładzie. Prezentacje powinny dotyczyć nowatorskich i mniej typowych (eksperymentalnych, zorientowanych naukowo) technologii i narzędzi programistycznych. Powinny być one przygotowywane w grupach 1-2 osobowych. Sugerowany czas prezentacji to 30 minut w przypadku jednego autora i 60 minut w przypadku dwóch autorów. Prezentacje będą oceniane w skali od 2.0 do 5.0. Uzyskanie oceny pozytywnej będzie uprawniało do zwolnienia ze sprawdzianu pisemnego.

Prezentacje przeznaczone są dla ograniczonej liczby chętnych. Studenci planujący przygotowanie prezentacji powinni zgłosić ten zamiar w ciągu trzech pierwszych tygodni semestru.

b) w zakresie projektów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę prezentacji multimedialnych oraz sprawozdań pisemnych przedstawiających postępy w realizacji projektów. Każda grupa projektowa zobowiązana jest przygotować dwie prezentacje i sprawozdania w ciągu semestru,

- ocenę i "obronę" przez grupę projektową końcowej prezentacji multimedialnej przedstawiającej zakończony projekt oraz demonstracji projektu,

- ocenę końcowego sprawozdania pisemnego przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,

- ocenę systemu informatycznego zrealizowanego przez grupę projektową z czytelności i udokumentowania kodu źródłowego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia (ze względu na szybki dziedziny rozwój wskazane poniżej technologie i narzędzia ze charakter przykładowy i mogą zostać zaktualizowane lub zmienione na nowsze odpowiedniki):

1. historia rozwoju platform mobilnych,
2. narzędzia, technologie oraz usługi do rozwoju aplikacji mobilnych, w szczególności aplikacji dla platformy iOS,
3. narzędzia i biblioteki służące do analizy danych i budowy modeli decyzyjnych przy użyciu technik uczenia maszynowego, np. Google TensorFlow, Microsoft Cognitive Services,
4. narzędzia i biblioteki służące do optymalizacji i rozwiązywania problemów przeszukiwania z ograniczeniami (ang. constraint satisfaction problems), np. MiniZinc, ECLiPSe,
5. narzędzia i biblioteki służące do analizy, przetwarzania i przeszukiwania dokumentów tekstowych, np. Apache Lucene, Elasticsearch,
6. narzędzia i biblioteki służące do budowy zaawansowanych środowisk dla przetwarzania danych, w tym danych nieustrukturalizowanych, np. Apache UIMA.

Zajęcia projektowe prowadzone są w formie piętnastu 3-godzinnych zajęć seminaryjnych odbywających się w laboratorium, podczas których studenci w grupach 2-4 osobowych realizują wybrane przez siebie projekty. Celem projektów jest zaprojektowanie i implementacja zaawansowanego systemu informatycznego przy wykorzystaniu technologii prezentowanych na wykładach. Tematy projektów związane są z charakterem specjalizacji i dotyczą szeroko pojętego podejmowania decyzji i przetwarzania danych. Program laboratorium obejmuje:

1. zapoznanie się z propozycjami projektów i wybór projektu przez każdą z grup,
2. zapoznanie się z tematem projektu wraz z niezbędną wiedzą dziedzinową,
3. opracowanie specyfikacji wymagań dla projektu, wybór technologii oraz przygotowanie harmonogramu prac,
4. przygotowanie formalnego projektu realizowanego systemu, jego implementację oraz testy,
5. przygotowanie prezentacji multimedialnych oraz sprawozdań pisemnych ? częściowych i końcowych (łącznie 3 prezentacje i sprawozdania),
6. przedstawienie prezentacji łącznie z ?obroną? prezentacji końcowej oraz demonstracją projektu.

Literatura podstawowa:

1. D. Conway, J. Myles White: Uczenie maszynowe dla programistów, Helion, 2015.
2. M.A. Lasoff: Podstawy języka Swift : programowanie aplikacji dla platformy iOS, Helion, 2016.
3. M. Galloway: Effective Objective-C 2.0 : 52 specific ways to improve your iOS and OS X programs, Addison-Wesley, 2013.

Literatura uzupełniająca:

1. A. Niederliński: Programowanie w logice z ograniczeniami. Łagodne wprowadzenie dla platformy ECLiPSe.
<http://www.pwizo.pl/>
2. dokumentacja użytkowa i techniczna wybranych narzędzi i środowisk

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach projektowych	45
2. przygotowanie prezentacji i sprawozdań na zajęcia projektowe (czas poza zajęciami)	15
3. dokończenie realizowanych projektów (czas poza zajęciami)	15
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności projektu (częściowo mogą być realizowane drogą elektroniczną)	3
5. udział w wykładach	30
6. zapoznanie się ze wskazaną literaturą / dokumentacją / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 50 stron	5
7. przygotowanie do prezentacji, zaliczenia wykładów, udział w sprawdzianie zaliczeniowym i omówienie jego wyników	12

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	78	3
Zajęcia o charakterze praktycznym	75	3