

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Zaawansowane projektowanie obiektowe		Kod 1010515311010500606
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 1
Ścieżka obieralności/specjalność Technologie wytwarzania oprogramowania	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 12 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: 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:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
2	Umiejętności:	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
3	Kompetencje społeczne	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl 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: 1. Przekazanie studentom wiedzy dotyczącej zaawansowanych metod programowania obiektowego. 2. Pokazanie i przećwiczenie nowoczesnego programowania obiektowego z wykorzystaniem technologii Microsoft .Net 3. Prezentacja nowoczesnych języków programowania obiektowego - Kotlin, Swift. 3. Kształtowanie u studentów organizacji zespołu projektowego oraz umiejętności pracy zespołowej.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza: 1. ma zaawansowaną i pogłębioną wiedzę z zakresu podstaw budowania systemów informatycznych oraz metod, narzędzi, języków i środowisk programistycznych - [K2st_W1] 2. ma zaawansowaną wiedzę szczegółową dotyczącą programowania obiektowego - [K2st_W3] 3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki - [K2st_W4] 4. ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów informatycznych - [K2st_W5] 5. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich i prowadzeniu prac badawczych w wybranym obszarze informatyki - [K2st_W6]		
Umiejętności:		

1. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne oraz eksperymentalne - [K2st_U4]
2. potrafi ? przy formułowaniu i rozwiązywaniu zadań inżynierskich ? integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]
3. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K2st_U6]
4. potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8]
5. potrafi zaprojektować złożony system informatyczny oraz zrealizować ten projekt ? co najmniej w części ? używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]
6. potrafi współdziałać w zespole, przyjmując w nim różne role - [K2st_U15]
Kompetencje społeczne:
1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]
2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [K2st_K2]

Sposoby sprawdzenia efektów kształcenia
Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób: Ocena formująca: - na podstawie odpowiedzi udzielanych odnośnie realizacji zadań w ramach laboratoriów; Ocena podsumowująca: - ocena końcowa z ćwiczeń laboratoryjnych wynika z ocen cząstkowych otrzymanych z trzech zadań domowych realizowanych przez studenta pomiędzy terminami laboratoriów. Student może też zdecydować się na zaliczenie poprzez zrealizowanie jednego dużego projektu przez cały semestr. - wykład kończy się egzaminem pisemnym w formie testu składającego się z kilkunastu pytań. Niektóre z nich mają formę testu wyboru, inne charakter opisowy. Do zaliczenia wymagane jest uzyskanie 50% punktów.
Treści programowe
Tematyka zajęć obejmuje zagadnienia związane z wytwarzaniem nowoczesnego oprogramowania z wykorzystaniem paradygmatu obiektowego. Szczegółowe zagadnienia omawiane w trakcie zajęć: - przypomnienie zagadnień związanych z programowaniem obiektowym, takich jak np. polimorfizm, - charakterystyka platformy Microsoft .Net, - techniki refleksji, - tworzenie grafiki dwuwymiarowej z wykorzystaniem biblioteki GDI+, - tworzenie własnych kontrolek, - dostęp do baz danych w technologii ADO.Net, - wytwarzanie nowoczesnych interfejsów użytkownika w technologii Windows Presentation Foundation, - tworzenie bogatych aplikacji internetowych w technologii Silverlight, - realizowanie warstwy komunikacji sieciowej z wykorzystaniem technologii Windows Communication Foundation - przykładowych nowoczesnych języków programowania obiektowego. Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta. Metody dydaktyczne: 1. wykład: prezentacja multimedialna, dyskusja. 2. laboratoria: .
Literatura podstawowa: 1. Wileczek, Rafał: Microsoft Visual C ++ 2008 : tworzenie aplikacji dla Windows, Helion 2009 2. Troelsen, Andrew W: Język C# 2010 i platforma .NET 4, Wydawnictwo Naukowe PWN, 2011 3. Podstawy języka Swift : programowanie aplikacji dla platformy iOS, Mark A. Lassoﬀ & Tom Stachowitz, Helion 2016 4. Swift 4 : koduj jak mistrz, Jon Hoffman, Helion 2018
Literatura uzupełniająca: 1. Juval Löwy : Programowanie usług WCF, Helion 2012 2. Cisek J.: Tworzenie nowoczesnych aplikacji graficznych w WPF, Helion 2012 3. Kotlin w akcji, Dmitry Jemerov, Svetlana Isakova, Helion 2018

Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach wykładowych	12	
2. udział w zajęciach laboratoryjnych	16	
3. przygotowanie do zajęć laboratoryjnych	8	
4. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	22	
5. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (~10 stron tekstu naukowego = 1 godz.), 200 stron.	20	
6. udział w konsultacjach (mogą być realizowane drogą elektroniczną) związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2	
7. przygotowanie do zaliczenia wykładów i udział w egzaminie (2h), omówienie wyników (1h)	20	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1
Zajęcia o charakterze praktycznym	46	2