

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Modelowanie i analiza systemów informatycznych		Kod 1010512311010513996
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) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 5
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		Podział ECTS (liczba i %) 5 100%
Odpowiedzialny za przedmiot / wykładowca:		
<p>dr inż. Bartosz Walter email: Bartosz.Walter@cs.put.poznan.pl tel. 61 6652980 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	<p>Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_W1-2, K_W4, K_W6-15, K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, K_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl</p> <p>Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu inżynierii oprogramowania oraz programowania obiektowego.</p>
2	Umiejętności:	<p>Powinien posiadać umiejętność modelowania w języku UML i implementacji prostych systemów informatycznych oraz znajomość podstawowych problemów związanych z analizą wymagań i tworzeniem specyfikacji wymagań dla oprogramowania. Powinien potrafić pozyskiwać informację ze wskazanych źródeł.</p>
3	Kompetencje społeczne	<p>Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu.</p> <p>Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.</p>
Cel przedmiotu:		
<ol style="list-style-type: none"> Przekazanie studentom wiedzy z zakresu projektowania obiektowego, w zakresie rozumienia czym są obiekty, jak definiować ich odpowiedzialność i kształtować relacje między nimi. Przedstawienia studentom metod oceny jakości projektu obiektowego z wykorzystaniem metryk obiektowych i tzw. Code smells Kształtowanie u studentów umiejętności pracy zespołowej w kontekście projektowania systemu informatycznego Przedstawienia studentom testowania jednostkowego jako metody weryfikacji realizacji odpowiedzialności przez obiekty Przedstawienie studentom wzorców obiektowych oraz kształcenie u nich umiejętności ich stosowania w tworzeniu systemów informatycznych. Przedstawienie studentom metod refaktoryzacji oprogramowania. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów informatycznych, projektowania oprogramowania, testowania oprogramowania, weryfikacji oprogramowania, inżynierii oprogramowania - [K_W4]</p> <p>2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: tworzenie architektury systemów, dokumentowanie architektury systemów, ocena architektury systemów, modelowanie oprogramowania, projektowanie oprogramowania, testowanie oprogramowania, weryfikacja oprogramowania - [K_W5]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, - [K_W6]</p> <p>4. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych, - [K_W7]</p> <p>5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki, - [K_W8]</p>
<p>Umiejętności:</p> <p>1. potrafi pozyskiwać informacje 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, - [K_U1]</p> <p>2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, - [K_U5]</p> <p>3. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K_U9]</p> <p>4. 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 - [K_U10]</p> <p>5. potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12]</p> <p>6. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]</p> <p>7. potrafi stworzyć model obiektowy prostego systemu (np. w języku UML) - [K_U17]</p> <p>8. potrafi - zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne - zaprojektować złożone urządzenie, system informatyczny lub proces 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 - [K_U27]</p>
<p>Kompetencje społeczne:</p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, - [K_K1]</p> <p>2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K_K4]</p> <p>3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]</p>

Sposoby sprawdzenia efektów kształcenia

Efektów kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:
- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach,
- b) w zakresie laboratoriów:
- na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze testu wielokrotnego wyboru i pytań problemowych, z testu można uzyskać 10 punktów, aby zaliczyć test na ocenę 3.0 należy zdobyć ponad 5.0 punktów
 - omówienie wyników egzaminu,
- b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez:
- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian wejściowy) oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
 - ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
 - ocenę i obronę przez studenta sprawozdania z realizacji projektów laboratoryjnych; ocena ta w niektórych przypadkach obejmuje także umiejętność pracy w zespole.
- Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:
- omówienia dodatkowych aspektów zagadnienia,
 - 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,
 - opracowanie wyróżniającego rozwiązania zadania - do wykorzystania w postaci studium przypadku
 - wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe		
<p>Program wykładu obejmuje następujące zagadnienia:</p> <p>Obiekty, ich rola i odpowiedzialność w modelowaniu. Mechanizmy obiektowości. Kryteria jakości projektu obiektowego. Metryki obiektowe. Testowanie jednostkowe jako metoda weryfikacji obiektów. Obiekty zastępcze. Wzorce obiektowe - koncepcja, zapis, systematyka. Katalog wzorców projektowych GoF: cel, struktura, uczestnicy i konsekwencje każdego z nich. Zjawisko erozji kodu - przyczyny, objawy, skutki. Wysokopoziomowa ocena jakości oprogramowania za pomocą tzw. code smells. Metody identyfikacji code smells. Przegląd wybranych przekształceń refaktoryzacyjnych. Metody weryfikacji poprawności przekształceń refaktoryzacyjnych. Koncepcja odwrócenia sterowania i wstrzykiwania zależności. Przykłady modelowania systemów - case study.</p> <p>Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium. Ćwiczenia realizowane są indywidualnie lub przez zespoły 2-4 studentów.</p> <p>Program laboratorium obejmuje następujące zagadnienia:</p> <p>Modelowanie odpowiedzialności obiektów z wykorzystaniem kart CRC. Pomiary, analiza i interpretacja metryk obiektowych. Tworzenie testów jednostkowych. Zastosowanie obiektów zastępczych w testach jednostkowych. Dobór i stosowanie właściwych wzorców projektowych w problemach projektowych. Identyfikacja code smells w kodzie programów. Narzędzia wykrywania anomalii projektowych. Stosowanie przekształceń refaktoryzacyjnych w kodzie programu (manualne oraz z użyciem narzędzi). Zaprojektowanie oraz implementacja prostego programu wykorzystującego koncepcję odwróconego sterowania i wstrzykiwania zależności. Zaprojektowanie oraz implementacja prostego programu wykorzystującego technologię JEE.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> 1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, pokaz multimedialny, demonstracja, dyskusja, studium przypadku 2. ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia praktyczne, dyskusja, praca w zespole, pokaz multimedialny, warsztaty, studium przypadku, demonstracja 		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. E. Gamma i in.: Wzorce projektowe. Elementy projektowania obiektowego wielokrotnego użytku. Helion, 2010. 2. M. Fowler: Refaktoryzacja. Ulepszanie struktury istniejącego kodu. Helion, 2011. 3. J. Kerievsky: Refaktoryzacja do wzorców projektowych. Helion, 2005. 4. R. C. Martin: Czysty kod. Podręcznik dobrego programisty. Helion, 2010 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. B. Meyer: Programowanie zorientowane obiektowo. Helion, 2005. 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych	30	
2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych (częściowo mogą być realizowane drogą elektroniczną)	4	
3. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	45	
4. udział w wykładach	30	
5. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 60 stron	6	
6. omówienie wyników egzaminu	2	
7. przygotowanie do egzaminu	5	
8. egzamin	2	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	124	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	68	3
Zajęcia o charakterze praktycznym	75	3