

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Modelowanie i analiza systemów informatycznych 1		Kod 1010515311010514556
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: 28 Ćwiczenia: - Laboratoria: 32 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 nauki techniczne		Podział ECTS (liczba i %) 5 100% 5 100%
Odpowiedzialny za przedmiot / wykładowca: Bartosz Walter email: bartosz.walter@cs.put.poznan.pl tel. 616652980 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu inżynierii oprogramowania oraz programowania obiektowego.
2	Umiejętności:	Student 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.
3	Kompetencje społeczne	Student powinien potrafić pozyskiwać informację ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć 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: 1. Przekazanie studentom wiedzy z zakresu projektowania obiektowego, w zakresie rozumienia czym są obiekty, jak definiować ich odpowiedzialność i kształtować relacje między nimi. 2. Przedstawienia studentom metod oceny jakości projektu obiektowego z wykorzystaniem metryk obiektowych i tzw. Code smells 3. Kształtowanie u studentów umiejętności pracy zespołowej w kontekście projektowania systemu informatycznego 4. Przedstawienia studentom testowania jednostkowego jako metody weryfikacji realizacji odpowiedzialności przez obiekty 5. Przedstawienie studentom wzorców obiektowych oraz kształcenie u nich umiejętności ich stosowania w tworzeniu systemów informatycznych. 6. Przedstawienie studentom metod refaktoryzacji oprogramowania.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		

1. student ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów informatycznych, projektowania oprogramowania, testowania oprogramowania, weryfikacji oprogramowania, inżynierii oprogramowania - [K_W4+++]
2. student 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+++]
3. student ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, - [K_W6++]
4. student ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych, - [K_W7++]
5. student zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki, - [K_W8++]

Umiejętności:

1. student 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++]
2. student potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, - [K_U5+++]
3. student potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K_U9+]
4. student 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++]
5. student potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12++]
6. student potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13++]
7. student potrafi stworzyć model obiektowy prostego systemu (np. w języku UML) - [K_U17+++]
8. student 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++]

Kompetencje społeczne:

1. student rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1++]
2. student 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+]
3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6+]

Sposoby sprawdzenia efektów kształcenia

<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:</p> <p>Ocena formująca:</p> <p>a) w zakresie wykładów: ? na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach,</p> <p>b) w zakresie laboratoriów: ? na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych,</p> <p>Ocena podsumowująca:</p> <p>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,</p> <p>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.</p> <p>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:</p> <p>? 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.</p>	
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>	
Literatura podstawowa:	
<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. R. C. Martin: Czysty kod. Podręcznik dobrego programisty. Helion, 2010 	
Literatura uzupełniająca:	
<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: 15 x 2 godz.	30
2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	4 45
3. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	30
4. udział w wykładach	6
5. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 60 stron	2 7
6. omówienie wyników egzaminu	
7. przygotowanie do egzaminu i egzamin	
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