

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Modelowanie i analiza systemów informatycznych 1		Kod 1010515311010504556
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) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
Odpowiedzialny za przedmiot / wykładowca: Odpowiedzialny za przedmiot / wykładowca:		
dr inż. Bartłomiej Prędko email: Bartlomiej.Predki@cs.put.poznan.pl tel. 616652932 Wydział Informatyki ul. Piotrowo 2 60-965 Poznań		mgr inż. Michał Maćkowiak email: michal.mackowiak@cs.put.poznan.pl tel. 616652944 Wydział Informatyki ul. Piotrowo 2 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:		
<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:		
<ol style="list-style-type: none"> ma zaawansowaną i pogłębioną wiedzę z zakresu obiektowych systemów informatycznych, ich budowania, projektowania i testowania - [K2st_W1] ma zaawansowaną wiedzę szczegółową dotyczącą programowania zorientowanego obiektowo, testowania jednostkowego, metryk systemów obiektowych - [K2st_W3] ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia obiektowych systemów informatycznych - [K2st_W5] zna zaawansowane techniki liczenia metryk obiektowych, projektowania systemów obiektowych, testowania systemów obiektowych - [K2st_W6] 		
Umiejętności:		

<p>1. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych dotyczący projektowania i testowania systemów obiektowych metody analityczne - [K2st_U4]</p> <p>2. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich związanych z projektowaniem obiektowym, testowaniem - integrować wiedzę z różnych obszarów informatyki, takich jak inżynieria oprogramowania i bazy danych, oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]</p> <p>3. potrafi ocenić przydatność i możliwość wykorzystania nowych metod i narzędzi z zakresu projektowania obiektowego - [K2st_U6]</p> <p>4. potrafi dokonać krytycznej analizy istniejących rozwiązań technicznych oraz zaproponować ich ulepszenia (usprawnienia) - [K2st_U8]</p> <p>5. potrafi - zgodnie z zadaną specyfikacją, uwzględniającą aspekty pozatechniczne - zaprojektować złożony obiektowy system informatyczny - co najmniej w części - używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]</p> <p>6. potrafi współdziałać w zespole, przyjmując w nim różne role - [K2st_U15]</p>
Kompetencje społeczne:
<p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K2st_K1]</p> <p>2. rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki w rozwiązywaniu problemów badawczych i praktycznych - [K2st_K2]</p>

Sposoby sprawdzenia efektów kształcenia
<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:</p> <p>Ocena formująca:</p> <p>a) w zakresie wykładów:</p> <ul style="list-style-type: none">* na podstawie odpowiedzi na pytania dotyczące materiału omówionego na wykładach, <p>b) w zakresie laboratoriów:</p> <ul style="list-style-type: none">* na podstawie oceny bieżącego postępu realizacji zadań laboratoryjnych, <p>Ocena podsumowująca:</p> <p>a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">* 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>b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez:</p> <ul style="list-style-type: none">* 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>Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:</p> <ul style="list-style-type: none">* 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</p>

sterowania i wstrzykiwania zależności. Zaprojektowanie oraz implementacja prostego programu wykorzystującego technologię JEE.		
Literatura podstawowa:		
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:		
1. B. Meyer: Programowanie zorientowane obiektowo. Helion, 2005.		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych	32	
2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych	2	
3. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	45	
4. udział w wykładach	28	
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 i egzamin	8	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	64	3
Zajęcia o charakterze praktycznym	77	3