

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie pragmatyczne		Kod 1010511371010510096
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 4 / 7
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stoień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 15 Ćwiczenia: - Laboratoria: 15 Projekty/seminaria: -		Liczba punktów 2
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 %) 2 100% 2 100%
Odpowiedzialny za przedmiot / wykładowca: Bartosz Wieloch email: bartosz.wieloch@cs.put.poznan.pl tel. 61 6653057 Instytut Informatyki Piotrowo 2		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu programowania imperatywnego, paradygmatu programowania obiektowego, oraz inżynierii oprogramowania.
2	Umiejętności:	Student powinien posiadać umiejętność projektowania i implementowania prostych systemów informatycznych.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i wykazywać 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 podstawowej wiedzy na temat sposobów tworzenia kodu zrozumiałego dla innych programistów, a w szczególności takiego który umożliwia późniejsze wprowadzanie zmian w systemie przy ciągłym zachowaniu czytelności kodu źródłowego. 2. Rozwijanie u studentów umiejętności analizy i refaktoryzacji kodu źródłowego, oraz testowania tworzonego oprogramowania. 3. Kształtowanie u studentów umiejętności efektywnej pracy jako programista w zespole programistycznym		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza: 1. Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie inżynierii oprogramowania - [K1st_W4] 2. Ma wiedzę o trendach w inżynierii oprogramowania i sposobach utrzymania kodu - [K1st_W5] 3. Zna podstawowe techniki i narzędzia wykorzystywane w procesie tworzenia i utrzymania oprogramowania - [K1st_W7]		
Umiejętności: 1. Potrafi pozyskiwać informacje na temat zasad tworzenia oprogramowania z literatury oraz innych źródeł w języku polskim jak i angielskim, integrować je, dokonywać interpretacji, wyciągać wnioski i wyczerpująco uzasadniać opinie - [K1st_U1] 2. Potrafi posługiwać się technikami informacyjno-komunikacyjnymi przy tworzeniu oprogramowania - [K1st_U2] 3. Potrafi zaprojektować i zaimplementować przy użyciu adekwatnych narzędzi oprogramowanie zgodnie z zadaną specyfikacją - [K1st_U10] 4. Potrafi opracować i zaimplementować algorytmy w zakresie programowania pragmatycznego - [K1st_U11]		
Kompetencje społeczne:		

1. Rozumie, że w informatyce wiedza i umiejętności szybko się starzeją i należy ciągle się dokształcać - [K1st_K1]
 2. Ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich, w szczególności tworzenia oprogramowania - [K1st_K2]

Sposoby sprawdzenia efektów kształcenia

Ocena formująca:

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

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- ocenianie ciągle, na każdym zajęciach (odpowiedzi ustne) ? premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,
- ocenę realizacji 2-3 zadań domowych,
- ocenę wiedzy i umiejętności uzyskanych na zajęciach w formie pisemnego kolokwium składającego się z około 5 pytań otwartych oraz zadań; każde pytanie jest punktowane w skali 0-5 punktów.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za aktywność na zajęciach laboratoryjnych.

Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

- ? wprowadzenie, w tym podstawowe informacje dotyczące zapewniania jakości kodu,
- ? konwencje nazywania klas, metod i zmiennych,
- ? sposoby tworzenia czytelnych funkcji,
- ? zalecenia dotyczące komentowania kodu,
- ? formatowanie kodu źródłowego,
- ? obsługa błędów,
- ? testy jednostkowe i programowanie sterowane testami (ang. Test Driven Development, TDD)

Program zajęć laboratoryjnych obejmuje następujące zagadnienia:

- ? zintegrowane środowisko programistyczne, np. Eclipse, IntelliJ IDEA, Visual Studio,
- ? inspekcja kodu,
- ? refaktoryzacja istniejącego kodu źródłowego,
- ? posługiwanie się systemem wersji, np. Git,
- ? implementacja na podstawie specyfikacji,
- ? tworzenie testów jednostkowych,
- ? wykorzystanie techniki test driven development.

Literatura podstawowa:

1. Andrew Hunt, David Thomas, "Pragmatyczny programista : od czeladnika do mistrza"
2. Robert C. Martin, "Czysty kod : podręcznik dobrego programisty"

Literatura uzupełniająca:

1. ----

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych	15
2. przygotowanie do ćwiczeń laboratoryjnych	5
3. zadania domowe (w ramach pracy własnej)	5
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu	2 15
5. udział w wykładach	10
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	3
7. przygotowanie do zaliczenia wykładów.	

Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	55	2
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	32	1
Zajęcia o charakterze praktycznym	25	1