Politechnika Krakowska im. Tadeusza Kościuszki

Karta przedmiotu

obowiązuje studentów rozpoczynających studia w roku akademickim 2020/2021

Wydział Mechaniczny

Kierunek studiów: Informatyka Stosowana | Profil: Ogólnoakademicki

Forma sudiów: stacjonarne | Kod kierunku: S

Stopień studiów: II

Specjalności: Bez specjalności

1 Informacje o przedmiocie

Nazwa przedmiotu	Programowanie obiektowe II
Nazwa przedmiotu w języku angielskim	
Kod przedmiotu	WM INFST oIIS B10 20/21
Kategoria przedmiotu	Przedmioty kierunkowe
Liczba punktów ECTS	4.00
Semestry	1

2 Rodzaj zajęć, liczba godzin w planie studiów

Semestr	Wykład	Ćwiczenia	Laboratorium	Laboratorium komputerowe	Projekt	Seminarium
1	15	0	0	30	15	0

3 Cele przedmiotu

Cel 1 Przekazanie wiedzy i umiejętności z zakresu zaawansowanego programowania obiektowego

4 Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

5 Efekty kształcenia

EK1 Wiedza Student zna i rozumie wzorce projektowe programowania obiektowego.

EK2 Wiedza Student zna i rozumie cele i metody refaktoryzacji kodu.

EK3 Wiedza Student zna i rozumie cele i metody testowania kodu obiektowego.

EK4 Umiejętności Student potrafi zastosować zdobytą wiedzę do programowania zaawansowanych aplikacji obiektowych.

6 Treści programowe

Wykład		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
W1	Wykład wstępny, wprowadzenie do zajęć, zasady oceniania z przedmiotu, plan wykładów, laboratoriów i projektów. Wprowadzenie do środowiska programistycznego i wybranego języka programowania obiektowego. Wprowadzenie do składni języka, standardy organizacji kodu. Korzystanie z zewnętrznych bibliotek i modułów, typy danych, instrukcje. Obiektowość: klasy, dziedziczenie, polimorfizm. Wzorce projektowe wprowadzenie. Wzorce i antywzorce. Przykłady w wybranych językach programowania obiektowego. Antywzorce implementacyjne i wzorce konstrukcyjne. Przykłady w wybranych językach programowania obiektowego. Wzorce projektowe strukturalne i czynnościowe. Przykłady w wybranych językach programowania obiektowego. Testowanie oprogramowania. Rodzaje testów, błędy, aksjomaty testowania, cele i poziomy testowania. Testy jednostkowe, integracyjne, systemowe i akceptacyjne. Metodyki i pokrycie kodu testami. Refaktoryzacja. Definicja, koszt i zastosowanie. Typy przekształceń, przykre zapachy w kodzie (m.in. Duplicated Code, Long Method, Large Class itp.) Metryki obiektowe - Złożoność cyklomatyczna McCabe'a, Zestaw MOOD, Metryki Chidamber&Kemerer, Metryki R. C. Martina, Prawo Demeter. Programowanie aspektowe - Rozdział zagadnień (ang. Separation of Concerns), Elementy języka AspectJ, Inne implementacje AOP	15

Laboratorium komputerowe		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin

Laboratorium komputerowe		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
K1	Podstawy obsługi programu Pycharm. Podstawy składni języka Python: Operatory, typy liczbowe i tekstowe. Kolejnymi funkcjonalności środowiska programistycznego PyCharm, oraz z języka Python: intrukcje warunkowe, pętle, Comprehensions, zasady definiowania funkcji i obsługa pakietów i modułów. Zasady programowania obiektowego w Pythonie. Wzorzec projektowy Metoda Wytwórcza (Factory Pattern) i Fabryka Abstrakcyjna (Abstract Factory) Wzorzec projektowy kreacyjny Budowniczy (ang. Builder Pattern) Wzorce projektowe kreacyjne Prototyp (ang. Prototype pattern) i Singleton (ang. Singleton Pattern) Wzorce strukturalne Adapter (ang. Adapter Pattern) i Dekorator (ang. Decorator Pattern) Wzorce strukturalne Most (ang. Bridge Pattern) i Fasada (ang. Faade Pattern) Wzorce strukturalne Pyłek (ang. Flyweight), MVC (ang. Model-View-Controler Pattern) i Pełnomocnik (ang. Proxy Pattern) Wzorce czynnościowe Łańcuch Odpowiedzialności (ang. Chain of responsibility Pattern) i Polecenie (ang. Command Pattern) Wzorce czynnościowe Obserwator (ang. Observer Pattern) i Stan (ang. State Pattern) Wzorce czynnościowe Interpreter (ang. Interpreter Pattern) i Strategia (Strategy Pattern) Wzorce czynnościowe Pamiątka (ang. Memento), Iterator (ang. Iterator Pattern) i Szablon (ang. Template Pattern) Testowanie jednostkowe w Pythonie. Zapoznanie się z istniejącymi Framworkami do testów jednostkowych. Przeprowadzenie prosty testów. Zaawansowane testowanie, testy po stronie systemu kontroli wersji. Refaktoryzacja. Odnajdowanie i poprawianie przykrych zapachów w kodzie	30

Projekt		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
P1	Wybór tematyki projektu własnego. Specyfikacja wymagań funkcjonalnych i niefunkcjonalnych, określenie docelowej platformy oraz języka i środowiska programowania. Wykonanie projektu aplikacji: określenie przypadków użycia, wykonanie niezbędnych diagramów UML. Implementacja aplikacji mobilnej w wybranym środowisku programowania z uwzględnieniem wymaganych funkcjonalności oraz obsługą sytuacji wyjątkowych. Równoczesne przygotowywanie i przeprowadzanie testów jednostkowych i integracyjnych. Przygotowanie w wybranym systemie i prezentacja dokumentacji projektowej. Prezentacja projektu i zaliczenie	15

7 Narzędzia dydaktyczne

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Ćwiczenia projektowe

8 Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	60
Konsultacje przedmiotowe	4
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	16
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	40
Sumaryczna liczba godzin dla przedmiotu wynikająca z całego nakładu pracy studenta	**120**
Sumaryczna liczba punktów ECTS dla przedmiotu	4.00

9 Sposoby oceny

Ocena formująca

F1 Test z wykładu

F2 Ćwiczenie praktyczne

F3 Projekt indywidualny

Ocena podsumowująca

P1 Średnia ważona ocen formujących

Warunki zaliczenia przedmiotu

W1 Pozytywna ocena z wykładu

W2 Pozytywne oceny z laboratoriów

W3 Pozytywne oceny z projektów

W4 Obecność studenta na min. 75% zajęć laboratoryjnych

Kryteria oceny

Efekt kształcenia 1

Na ocenę 3.0	Student zna i rozumie podstawowe wzorce projektowe programowania obiektowego.
Efekt kształcenia 2	
Na ocenę 3.0	Student zna i rozumie w podstawowym zakresie cele i metody refaktoryzacji kodu.
Efekt kształcenia 3	
Na ocenę 3.0	Student zna i rozumie w elementarnym stopniu cele i metody testowania kodu obiektowego.
Efekt kształcenia 4	
Na ocenę 3.0	Student potrafi zastosować w elementarnym stopniu zdobytą wiedzę do programowania zaawansowanych aplikacji obiektowych.

10 Macierz realizacji przedmiotu

Efekt kształcenia	Odniesienie danego efektu do szczegółowych efektów zdefiniowanych dla programu	Cele przedmiotu	Treści programowe	Narzędzia dydaktyczne	Sposoby oceny
EK1		Cel 1	W1 K1 P1	N1 N2 N3	F1 F2 F3 P1
EK2		Cel 1	W1 K1 P1	N1 N2 N3	F1 F2 F3 P1
EK3		Cel 1	W1 K1 P1	N1 N2 N3	F1 F2 F3 P1
EK4		Cel 1	W1 K1 P1	N1 N2 N3	F1 F2 F3 P1

11 Wykaz literatury

12 Informacje o nauczycielach akademickich

Osoba odpowiedzialna za kartę

dr hab. inż., prof. PK Jacek Pietraszek (kontakt: jacek.pietraszek@pk.edu.pl)

Osoby prowadzące przedmiot

1 pracownicy Instututu Informatyki Stosowanej (kontakt:)

13 Zatwierdzenie karty przedmiotu do realizacji

(miejscowość, data) (odpowiedzialny za przedmiot) (dziekan)

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot)

.....