

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

| | | | | |
|---|--------------------|--|-----------------------------------|---------------------|
| Kod przedmiotu | | Nazwa przedmiotu | INŻYNIERIA OPROGRAMOWANIA | |
| E/O/2/ST/C1A-5-AII | | | SOFTWARE ENGINEERING | |
| Język wykładowy | | język polski | | |
| Rok akademicki | | 2023/2024 | | |
| Kierunek | | Elektrotechnika | | |
| w zakresie | | Automatyka i informatyka | | |
| Poziom studiów | | studia drugiego stopnia | | |
| Profil studiów | | ogólnoakademicki | | |
| Forma studiów | | studia stacjonarne | | |
| Semestr / semestry | | 2 | | |
| Przynależność do grupy zajęć | | C1A. Grupa zajęć obieralnych – zajęcia obowiązkowe | | |
| Status przedmiotu | | obowiązkowy | | |
| Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS | | Forma zajęć | Liczba godzin zajęć dydaktycznych | Liczba punktów ECTS |
| | | Wykład | 15 [h] | 1,5 ECTS |
| | | Laboratorium | 15 [h] | |
| Powiązanie przedmiotu | z profilem studiów | związany z prowadzoną działalnością naukową w dyscyplinach, do których przyporządkowany jest kierunek studiów | | 0,5 ECTS |
| | z uprawnieniami | służy do zdobywania przez studenta kompetencji inżynierskich | | 1 ECTS |
| | z dyscypliną | automatyka, elektronika, elektrotechnika i technologie kosmiczne | | 1,5 ECTS |
| Forma nauczania | | tradycyjna – zajęcia zorganizowane w Uczelni i/lub zajęcia z wykorzystaniem metod i technik kształcenia na odległość (max. 0,6 ECTS) | | |
| Wymagania wstępne | | - | | |
| Jednostka prowadząca | | Katedra Informatyki i Teleinformatyki | | |
| Koordynator | | dr hab. inż. Tomasz Ciszewski, prof. UTHRad | | |
| Adres strony internetowej pjo | | www.wteii.uniwersytetradom.pl | | |
| Adres e-mail, telefon koordynatora | | t.ciszewski@uthrad.pl, +48 48 361 77 33 | | |

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

| | |
|--|---|
| Cel kształcenia: | Celem zajęć jest przedstawienie całego procesu związanego z projektowaniem, tworzeniem i wykorzystywaniem systemów informatycznych. Powinien uświadomić słuchaczom, że programowanie jest tylko elementem składowym tego procesu oraz że na powodzenie przedsięwzięcia mają wpływ wszystkie fazy cyklu życia oprogramowania. Głównym składnikiem zajęć jest kurs kształcący umiejętność wykorzystania Unified Modeling Language (UML) w modelowaniu i projektowaniu systemów. |
| Treści programowe: | <p>Wykład [BN, W1, W2]: Wprowadzenie do inżynierii oprogramowania, złożoność projektów oprogramowania, modelowanie oprogramowania, obiektowość w UML, związki, agregacje i agregacje całkowite, przypadki użycia, diagramy stanu, diagramy przebiegu, diagramy kooperacji i czynności, diagramy komponentów i wdrożenia, wzorce projektowe. Suma 15 [h]</p> <p>Laboratorium [BN, U1, U2, K1]: Definiowanie wymagań systemowych, założenia projektowe i funkcjonalne, budowa modelu obiektowego systemu w UML, przygotowanie szkieletu aplikacji na podstawie diagramów, testowanie i weryfikacja wymagań. Suma 15 [h]</p> |
| Metody dydaktyczne (kształcenia): | <ul style="list-style-type: none"> – metody podające (wykład informacyjny) – metody problemowe (wykład problemowy) – metody programowane (z wykorzystaniem komputera), – metody praktyczne (pokaz, ćwiczenia laboratoryjne, metoda projektów) |
| Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej: | Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów uczenia się określonych dla danego przedmiotu. Uzyskanie pozytywnych ocen ze wszystkich form zajęć wchodzących w skład danego przedmiotu jest równoznaczne z jego zaliczeniem i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi. Sposób obliczenia oceny końcowej z przedmiotu określa regulamin studiów. |

| | |
|--|--|
| | <p>Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco:</p> <p>Ocenę z wykładu stanowi wynik otwartego testu pisemnego.</p> <p>Na ocenę z laboratorium składa się: punktowa ocena wykonanych zadań laboratoryjnych (60%) i punktowa ocena z pisemnego \ praktycznego kolokwium (40%).</p> <p>Zdobyte punkty przeliczane zostają na ocenę z laboratorium wg skali:</p> <p>Ocena 2 poniżej 51%</p> <p>Ocena 3 od 51%</p> <p>Ocena 3,5 od 61%</p> <p>Ocena 4 od 71%</p> <p>Ocena 4,5 od 81%</p> <p>Ocena 5 od 91%</p> |
|--|--|

| Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć | | | | Metody weryfikacji efektów uczenia się | |
|---|---|------------------------------------|--------------|--|---|
| Numer efektu uczenia się | Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi /(K) jest gotów do: | Kierunkowy efekt uczenia się (KEU) | Forma zajęć | Forma weryfikacji (zaliczeń) | Metody sprawdzania i oceny |
| W1 | kluczowe zagadnienia i pojęcia z zakresu inżynierii oprogramowania | K_WG03 | wykład | zaliczenie | test pisemny - pytania otwarte |
| W2 | symbole stosowane w dokumentacji i modelowaniu systemów oraz typy diagramów UML | K_WG09 | wykład | zaliczenie | test pisemny - pytania otwarte |
| U1 | tworzyć diagramy UML | K_UW02 | laboratorium | zaliczenie | punktacja zadań laboratoryjnych i/lub kolokwium praktyczne |
| U1 | odczytywać dokumentację i modele systemów informatycznych | K_UK09 | laboratorium | zaliczenie | punktacja zadań laboratoryjnych i/lub kolokwium praktyczne |
| K1 | włączenia modelowania systemów do procesów rozwojowych i wytwórczych | K_KO02 | laboratorium | obserwacja | dyskusja, aktywność na zajęciach, prezentacja prac, terminowość |

| Literatura i pomoce naukowe | |
|--|--|
| <ol style="list-style-type: none"> Schmuller J., UML dla każdego, Helion, Gliwice 2003 Sommerville I., Inżynieria oprogramowania, PWN, 2020 Stevens P., UML. Inżynieria oprogramowania, Helion, Gliwice 2007 Kuchta J.: Współczesne metody zapewniania jakości oprogramowania, PWN, Warszawa 2020 Fowler M., UML w kropelce, Oficyna Wydawnicza LTP, 2005 (wyd. III) Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika, WNT 2001 Sacha K., Inżynieria oprogramowania, PWN, Warszawa 2022 Subieta K, Wprowadzenie do inżynierii oprogramowania, Wydawnictwo PJWSTK, Warszawa 2002 Płodzeń J., Stemposz E., Analiza i projektowanie systemów informatycznych, Wydawnictwo PJWSTK, Warszawa 2005 Dąbrowski W., Stasiak A., Wolski M., Modelowanie systemów informatycznych w języku UML 2.1 w praktyce, MIKOM, Warszawa 2007 Koszłajda A., Zarządzanie projektami IT. Przewodnik po metodykach, Helion, Gliwice 2010. | |

| Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS | | | |
|--|-----------------------------|---|---------------------|
| Udział w zajęciach, aktywność | Obciążenie studenta [h] | | |
| | Inne godz. kontaktowe (IGK) | Zajęcia bez nauczyciela-praca własna studenta (ZBN) | Zajęcia dydaktyczne |
| Udział w wykładach | X | X | 15 [h] |
| Udział w ćwiczeniach / laboratoriach / projektach / seminariach | X | X | 15 [h] |
| Udział w konsultacjach | 3 [h] | X | X |
| Przygotowanie do wykładów / ćwiczeń / laboratoriów / projektów / seminariów | X | 4,5 [h] | X |
| Przygotowanie do zaliczenia/egzaminu | | | |
| Sumaryczne obciążenie pracą studenta | 3 [h] /0,1 ECTS | 4,5 [h] /0,2 ECTS | 30 [h] /1,2 ECTS |
| Punkty ECTS za przedmiot | 1,5 ECTS | | |

| Informacje dodatkowe, uwagi |
|---|
| <p>W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.</p> <p>Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekłe chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekłe chorych.</p> |