



Karta przedmiotu

| | | | | | | | |
|--|---|---|-----------|------------------------|--|-----------------------|-------|
| Nazwa i kod przedmiotu | Inżynieria oprogramowania, PG_00058932 | | | | | | |
| Kierunek studiów | Informatyka | | | | | | |
| Data rozpoczęcia studiów | październik 2023 r. | Rok akademicki realizacji przedmiotu | | | 2025/2026 | | |
| Poziom kształcenia | I stopnia - inżynierskie | Grupa zajęć | | | Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki | | |
| Forma studiów | niestacjonarne | Sposób realizacji | | | na uczelni | | |
| Rok studiów | 3 | Język wykładowy | | | polski | | |
| Semestr studiów | 5 | Liczba punktów ECTS | | | 4.0 | | |
| Profil kształcenia | ogólnoakademicki | Forma zaliczenia | | | egzamin | | |
| Jednostka prowadząca | Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Oprogramowania | | | | | | |
| Imię i nazwisko wykładowcy (wykładowców) | Odpowiedzialny za przedmiot | dr inż. Aleksander Jarzębowicz | | | | | |
| | Prowadzący zajęcia z przedmiotu | dr inż. Maciej Kucharski | | | | | |
| Formy zajęć i metody nauczania | Forma zajęć | Wykład | Ćwiczenia | Laboratorium | Projekt | Seminarium | RAZEM |
| | Liczba godzin zajęć | 15.0 | 0.0 | 15.0 | 0.0 | 0.0 | 30 |
| | W tym liczba godzin zajęć na odległość: 0.0 | | | | | | |
| Aktywność studenta i liczba godzin pracy | Aktywność studenta | Udział w zajęciach dydaktycznych, objętych planem studiów | | Udział w konsultacjach | | Praca własna studenta | RAZEM |
| | Liczba godzin pracy studenta | 30 | | 4.0 | | 66.0 | 100 |
| Cel przedmiotu | Przedmiot "Inżynieria Oprogramowania" jest ukierunkowany na przybliżenie zagadnień związanych z wytwarzaniem oprogramowania w warunkach przemysłowych: złożone systemy, przeznaczone dla rzeczywistego klienta, związane z określoną potrzebą biznesową i gwarancjami jakości, wytwarzane przez duże zespoły deweloperów. | | | | | | |

| Efekty uczenia się przedmiotu | Efekt kierunkowy | Efekt z przedmiotu | Sposób weryfikacji i oceny efektu |
|-------------------------------|--|--|------------------------------------|
| | [K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską | Student potrafi wykonać modele analityczne i projektowe systemu informatycznego posługując się w tym celu narzędziami CASE (Computer Aided Software Engineering). | [SU1] Ocena realizacji zadania |
| | [K6_U43] potrafi analizować dane oraz formułować, stosować i oceniać właściwe modele formalne i algorytmy rozwiązywania problemów w zakresie systemów i aplikacji informacyjnych | Student posługuje się notacją modelowania Unified Modeling Language (UML) i konstruuje modele systemów informatycznych. | [SU1] Ocena realizacji zadania |
| | [K6_U09] potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych związanych z kierunkiem studiów i ocenić te rozwiązania, a także wykorzystać zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów technicznych typowych dla kierunku studiów | Student opracowuje wizję systemu informatycznego zawierającą krytyczną analizę obecnego sposobu funkcjonowania organizacji klienckiej oraz podstawowe wymagania i ograniczenia względem systemu. | [SU1] Ocena realizacji zadania |
| | [K6_W06] zna i rozumie podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów specyficznych dla danego kierunku studiów | Student rozumie znaczenie praktyk inżynierskich i organizacji pracy zespołowej w przedsięwzięciu informatycznym. Wymienia i opisuje kluczowe obszary procesu wytwarzania oprogramowania. Wyjaśnia dobór praktyk w zależności od specyfiki przedsięwzięcia. | [SW1] Ocena wiedzy faktograficznej |
| | [K6_W42] zna i rozumie w zaawansowanym stopniu architektury, zasady projektowania oraz metody wsparcia sprzętowego i programowego dla lokalnych i rozproszonych systemów informatycznych, w tym systemów obliczeniowych, baz danych, sieci komputerowych i aplikacji informacyjnych, a także zasady współpracy człowieka z komputerem i wspomaganej komputerowo pracy zespołowej | Student rozumie zagadnienia projektowania systemów informatycznych (na poziomie architektury i poszczególnych modułów składowych), w tym aspekty software reuse oraz projektowania interfejsu użytkownika. | [SW1] Ocena wiedzy faktograficznej |

| | | | |
|---|---|--|-------------------------|
| Treści przedmiotu | <ol style="list-style-type: none"> 1. Wprowadzenie do przedmiotu 2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia. 3. Obszary działania inżynierii oprogramowania 4. Faza przedprojektowa: planowanie i zakres przedsięwzięcia. Podejście SSM i Rich Picture. 5. Ryzyko i odpowiedzialność społeczna związane z systemami informatycznymi 6. Podstawy inżynierii wymagań 7. Pojęcie modelowania konceptualnego 8. Przypadki użycia 9. Obiektowe podejście do analizy systemu w UML 10. Modelowanie logicznej struktury systemu: diagramy klas 11. Modelowanie struktury: inne diagramy struktury 12. Modelowanie dynamiki: diagramy sekwencji i komunikacji 13. Modelowanie dynamiki: reprezentowanie stanu obiektów 14. Projektowanie: Projekt ogólny (wysokiego poziomu) 15. Projektowanie: Projekt klas (szczegółowy) 16. Zagadnienia software reuse, wzorce projektowe 17. Projektowanie interfejsu użytkownika: motywacje, pojęcia, techniki 18. Testowanie: pojęcia, umiejscowienie w procesie wytwarzania 19. Testowanie: techniki (czarna i biała skrzynka), poziomy testowania, zarządzanie testowaniem 20. Wdrażanie i utrzymanie oprogramowania 21. Zarządzanie konfiguracją i ewolucja oprogramowania 22. Klasyczny cykl życia oprogramowania 23. Nieklasyczne cykle życia i modele wytwarzania oprogramowania 24. Dobór modelu wytwarzania do specyfikacji projektu 25. Metodyki wytwórcze (sterowane planem i zwinne) | | |
| Wymagania wstępne i dodatkowe | Obowiązkowa obecność na zajęciach laboratoryjnych. Konieczne oddanie i akceptacja wszystkich zadań laboratoryjnych. Oddawanie zadań z opóźnieniem skutkuje punktami ujemnymi. Do egzaminu dopuszczeni są jedynie studenci, którzy zaliczyli laboratorium. | | |
| Sposoby i kryteria oceniania osiągniętych efektów uczenia się | Sposób oceniania (składowe) | Próg zaliczeniowy | Składowa oceny końcowej |
| | Laboratorium (zadania i sprawdziany) | 50.0% | 50.0% |
| | Egzamin pisemny | 50.0% | 50.0% |
| Zalecana lista lektur | Podstawowa lista lektur | <ol style="list-style-type: none"> 1. Górski J. (red.): Inżynieria oprogramowania w projekcie informatycznym, MIKOM, 2000 2. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 3. Pressman R., Software Engineering: a Practitioner's Approach, 7th edition, McGraw-Hill, 2009 4. Sommerville I., Inżynieria oprogramowania, WNT, 2003 5. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010 6. Szejko S. (red): Metody wytwarzania oprogramowania, MIKOM 2002 7. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002 8. Fowler M., Scott K.: UML w kropelce 2.0 (ang. UML distilled), Lupus 2005 | |
| | Uzupełniająca lista lektur | Nie ma wymagań | |
| | Adresy eZasobów | Adresy na platformie eNauczanie: | |
| Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania | | | |
| Praktyki zawodowe w ramach przedmiotu | Nie dotyczy | | |