



Karta przedmiotu

| | | | | | | | |
|--|---|---|---|------------------------|--|---|-------|
| Nazwa i kod przedmiotu | Software engineering, PG_00045302 | | | | | | |
| Kierunek studiów | Inżynieria danych, Inżynieria danych | | | | | | |
| Data rozpoczęcia studiów | październik 2026 r. | Rok akademicki realizacji przedmiotu | | | 2027/2028 | | |
| 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 | stacjonarne | Sposób realizacji | | | na uczelni | | |
| Rok studiów | 2 | Język wykładowy | | | angielski | | |
| Semestr studiów | 3 | Liczba punktów ECTS | | | 3.0 | | |
| Profil kształcenia | ogólnoakademicki | Forma zaliczenia | | | egzamin | | |
| Jednostka prowadząca | Wydziały Politechniki Gdańskiej -> 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. Aleksander Jarzębowicz | | | | |
| Formy zajęć | Forma zajęć | Wykład | Ćwiczenia | Laboratorium | Projekt | Seminarium | RAZEM |
| | Liczba godzin zajęć | 15.0 | 0.0 | 30.0 | 0.0 | 0.0 | 45 |
| | 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 | 45 | | 6.0 | | 24.0 | 75 |
| Cel przedmiotu | Celem przedmiotu jest zapoznanie studentów z obszarami analizy i projektowania na tle innych obszarów działań projektu informatycznego oraz praktyczna nauka wykorzystania języka modelowania UML do analizy i projektowania systemów IT. | | | | | | |
| Efekty uczenia się przedmiotu | Efekt kierunkowy | | Efekt z przedmiotu | | | Sposób weryfikacji i oceny efektu | |
| | [K6_U04] formułuje logiczne rozwiązania złożonych lub nieustrukturyzowanych problemów | | Student potrafi tworzyć modele analityczne obejmujące różne perspektywy spojrzenia na system (funkcjonalna, strukturalna, dynamiczna) oraz dbać o spójność pomiędzy tymi modelami. | | | [SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania | |
| | [K6_K03] wykazuje się zdolnością do krytycznego i analitycznego myślenia oraz integruje wiedzę z wielu dyscyplin w celu podejmowania efektywnych decyzji | | Student realizuje w zespole projekt dla wybranego tematu, wykorzystując w tym celu wiedzę z wykładów oraz narzędzia w postaci języka modelowania i systemu CASE (Computer Aided Software Engineering). | | | [SK2] Ocena postępów pracy [SK1] Ocena umiejętności pracy w grupie | |
| | [K6_W01] identyfikuje uwarunkowania procesów zachodzących w analizowanych systemach i dobiera metody ich rozwiązania, wykorzystując zgromadzoną wiedzę i uwzględniając wzajemne relacje między analizowanymi zjawiskami | | Student rozumie zagadnienia procesów wytwarzania oprogramowania i organizacji projektu IT, potrafi dostosować organizację projektu do specyfiki jego kontekstu (obszaru problemowego i wytwarzanego systemu). | | | [SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym | |

| Treści przedmiotu | <p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> 1. Wprowadzenie do przedmiotu 2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia 3. Obszary i procesy inżynierii oprogramowania 4. Cykle życia oprogramowania (zarys) 5. Metodyki wytwarzania oprogramowania (zarys) 6. Pojęcie modelowania conceptualnego. Języki specyfikacji i modelowania. 7. Przypadki użycia 8. Obiektowe podejście do analizy systemu w UML 9. Modelowanie logicznej struktury systemu: diagramy klas 10. Modelowanie struktury: inne diagramy struktury 11. Modelowanie dynamiki: diagramy sekwencji i komunikacji 12. Modelowanie dynamiki: diagramy stanów i czynności 13. Inżynieria wymagań: pozyskiwanie, analiza i walidacja wymagań 14. Inżynieria wymagań: specyfikowanie wymagań 15. Projektowanie: architektura systemu 16. Projektowanie: projekt ogólny (wysokiego poziomu) i projekt klas (szczegółowy) 17. Projektowanie: wzorce projektowe 18. Projektowanie: zasady SOLID 19. Metodyka Scrum <p>Treści przedmiotu - laboratoria</p> <ol style="list-style-type: none"> 1. Opracowanie dokumentu Wizji Systemu 2. Opracowanie modelu przypadków użycia systemu 3. Opracowanie analitycznego diagramu klas systemu 4. Opracowanie wybranych diagramów dynamicznych opisujących elementy zachowania systemu (diagramy: sekwencji, komunikacji, czynności, stanów) | | | | | | | | | | | |
|---|---|-------------------------|--|-----------------------------|---|-------------------------|----------------------------|---|-------|-----------------|-------|-------|
| Wymagania wstępne i dodatkowe | | | | | | | | | | | | |
| Sposoby i kryteria oceniania osiągniętych efektów uczenia się | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Sposób oceniania (składowe)</th> <th style="width: 33%;">Próg zaliczeniowy</th> <th style="width: 33%;">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td>Laboratorium</td> <td>50.0%</td> <td>50.0%</td> </tr> <tr> <td>Egzamin</td> <td>50.0%</td> <td>50.0%</td> </tr> </tbody> </table> | | | Sposób oceniania (składowe) | Próg zaliczeniowy | Składowa oceny końcowej | Laboratorium | 50.0% | 50.0% | Egzamin | 50.0% | 50.0% |
| Sposób oceniania (składowe) | Próg zaliczeniowy | Składowa oceny końcowej | | | | | | | | | | |
| Laboratorium | 50.0% | 50.0% | | | | | | | | | | |
| Egzamin | 50.0% | 50.0% | | | | | | | | | | |
| Zalecana lista lektur | <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 33%;">Podstawowa lista lektur</td> <td colspan="2" style="width: 66%;"> <ol style="list-style-type: none"> 1. Pressman R., Maxim B., Software Engineering: a Practitioners Approach, McGraw-Hill, 9th edition, 2019 2. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide (2nd Edition), Addison-Wesley, 2005 </td> </tr> <tr> <td>Uzupełniająca lista lektur</td> <td colspan="2"> <ol style="list-style-type: none"> 1. Sommerville I., Software Engineering, 10th edition, Addison-Wesley, 2015 2. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 3. Fowler M., Scott K.: UML distilled 3rd ed, Addison-Wesley, 2003 4. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, O'Reilly Media, 2006 </td> </tr> <tr> <td>Adresy eZasobów</td> <td colspan="2"></td> </tr> </tbody> </table> | | | Podstawowa lista lektur | <ol style="list-style-type: none"> 1. Pressman R., Maxim B., Software Engineering: a Practitioners Approach, McGraw-Hill, 9th edition, 2019 2. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide (2nd Edition), Addison-Wesley, 2005 | | Uzupełniająca lista lektur | <ol style="list-style-type: none"> 1. Sommerville I., Software Engineering, 10th edition, Addison-Wesley, 2015 2. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 3. Fowler M., Scott K.: UML distilled 3rd ed, Addison-Wesley, 2003 4. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, O'Reilly Media, 2006 | | Adresy eZasobów | | |
| Podstawowa lista lektur | <ol style="list-style-type: none"> 1. Pressman R., Maxim B., Software Engineering: a Practitioners Approach, McGraw-Hill, 9th edition, 2019 2. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide (2nd Edition), Addison-Wesley, 2005 | | | | | | | | | | | |
| Uzupełniająca lista lektur | <ol style="list-style-type: none"> 1. Sommerville I., Software Engineering, 10th edition, Addison-Wesley, 2015 2. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 3. Fowler M., Scott K.: UML distilled 3rd ed, Addison-Wesley, 2003 4. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, O'Reilly Media, 2006 | | | | | | | | | | | |
| Adresy eZasobów | | | | | | | | | | | | |
| Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania | <ul style="list-style-type: none"> • Narysuj diagram UML (np. klas, przypadków użycia, stanów) na podstawie podanego opisu wymagań względem systemu. • Przedstaw określony model wytwarzania oprogramowania i omów jego zalety i wady. • Wymień i opisz techniki specyfikacji wymagań względem systemu. | | | | | | | | | | | |
| Zajęcia praktyczne w ramach przedmiotu | Nie dotyczy | | | | | | | | | | | |

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.