POLITECHNIKA GDAŃSKA

Karta przedmiotu

| | | | |
|---|---|---|---|
| Nazwa i kod przedmiotu | Inżynieria oprogramowania, PG_00064064 | | |
| Kierunek studiów | Fizyka Techniczna | | |
| Data rozpoczęcia studiów | październik 2026 r. | Rok akademicki realizacji przedmiotu | 2028/2029 |
| Poziom kształcenia | I stopnia - inżynierskie | Grupa zajęć | Grupa zajęć fakultatywnych<br>Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki |
| Forma studiów | stacjonarne | Sposób realizacji | na uczelni |
| Rok studiów | 3 | Język wykładowy | polski |
| Semestr studiów | 6 | Liczba punktów ECTS | 7.0 |
| Profil kształcenia | ogólnoakademicki | Forma zaliczenia | egzamin |
| Jednostka prowadząca | Wydziały Politechniki Gdańskiej -> Wydział Fizyki Technicznej i Matematyki Stosowanej -> Instytut Fizyki i Informatyki Stosowanej -> Zakład Fizyki Teoretycznej i Informatyki Kwantowej | | |
| Imię i nazwisko wykładowcy (wykładowców) | Odpowiedzialny za przedmiot | dr hab. inż. Marta Łabuda | |
| | Prowadzący zajęcia z przedmiotu | | |

| Formy zajęć | | | | | | |
|---|---|---|---|---|---|---|
| Forma zajęć | Wykład | Ćwiczenia | Laboratorium | Projekt | Seminarium | RAZEM |
| Liczba godzin zajęć | 30.0 | 0.0 | 0.0 | 45.0 | 0.0 | 75 |
| 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 | 75 | 5.0 | 95.0 | 175 |

| Cel przedmiotu |
|---|
| Celem przedmiotu jest zapoznanie studentów z zasadami planowania, projektowania architektury i metod wytwarzania oprogramowania, również naukowego. Celem zajęć projektowych jest praktyczne zastosowanie wiedzy z zakresu inżynierii oprogramowania poprzez realizację zespołowego projektu informatycznego. Studenci uczą się planowania i prowadzenia projektu, analizy i specyfikacji wymagań, doboru odpowiedniego modelu wytwarzania oprogramowania oraz projektowania architektury i struktury systemu. Zajęcia rozwijają umiejętności modelowania systemów z wykorzystaniem UML, pracy zespołowej, dokumentowania projektu oraz stosowania nowoczesnych narzędzi i technologii wspierających proces tworzenia oprogramowania. |

| Efekty uczenia się przedmiotu | Efekt kierunkowy | Efekt z przedmiotu | Sposób weryfikacji i oceny efektu |
|---|---|---|---|
| | [K6_K03] jest gotów do odpowiedzialnego pełnienia ról zawodowych, przestrzegania zasad etyki oraz dbałości o bezpieczeństwo pracy. | Student potrafi przygotować i zaprezentować wyniki zespołowego projektu informatycznego w sposób zrozumiały dla odbiorców o różnym poziomie wiedzy technicznej.<br>Student potrafi dokonać samooceny własnej pracy oraz konstruktywnie ocenić wkład i efekty pracy innych członków zespołu projektowego. | [SK4] Ocena umiejętności komunikacji, w tym poprawności językowej<br>[SK3] Ocena umiejętności organizacji pracy |
| | [K6_U03] posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania | Student potrafi tworzyć, kompilować i uruchamiać programy w wybranym języku programowania np. Python, stosując podstawowe konstrukcje językowe (instrukcje sterujące, struktury danych, funkcje/ metody).<br>Student potrafi stosować wybrane pakiety oprogramowania oraz narzędzia programistyczne (np. środowisko IDE, system kontroli wersji, repozytorium, biblioteki standardowe, frameworki frontendowe) w procesie tworzenia oprogramowania | [SU4] Ocena umiejętności korzystania z metod i narzędzi<br>[SU1] Ocena realizacji zadania |
| | [K6_W05] posiada wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice. | Student zna metodyki wytwarzania oprogramowania (np. kaskadową, iteracyjną, zwinną) oraz podstawowe techniki programowania stosowane w inżynierii oprogramowania.<br>Student zna wybrane narzędzia informatyczne wspomagające proces tworzenia oprogramowania (środowiska programistyczne, systemy kontroli wersji, narzędzia do modelowania (np CASE do. diagramów UML) i testowania, potrafi z nich korzystać oraz rozumie ich zastosowanie w problemach fizyki i techniki. | [SW1] Ocena wiedzy faktograficznej |
| | [K6_K02] jest gotów do twórczego wykorzystania swoich kompetencji dla dobra ogółu, również w sposób przedsiębiorczy. | Student zna podstawowe role zespołowe w projektach informatycznych (np. analityk, projektant, programista, tester, lider zespołu, scrum master, product owner) oraz zasady efektywnej współpracy zespołowej w inżynierii oprogramowania.<br>Student potrafi efektywnie pracować w zespole projektowym, realizując odpowiedzialnie powierzoną rolę i respektując ustalone zasady współpracy oraz dostosowując się do zmieniających się potrzeb zespołu.<br>Student potrafi komunikować się w zespole projektowym, dzielić się zadaniami, raportować postępy oraz współuczestniczyć w podejmowaniu decyzji zespołowych. | [SK1] Ocena umiejętności pracy w grupie<br>[SK3] Ocena umiejętności organizacji pracy |

| | |
|---|---|
| Treści przedmiotu | Treści przedmiotu - wykład<br>1. Wprowadzenie do inżynierii oprogramowania. Właściwości inżynierii systemów komputerowych<br><br>2. Planowanie projektu informatycznego: podstawowe charakterystyki, pojęcia, udziałowcy projektu; cykl życia i zakres projektu. Planowanie zadań. Identyfikacja problemu; Wzbogacony wizerunek.<br><br>3. Studium wykonalności projektu informatycznego. Cele, płaszczyzny oceny techniczna, ekonomiczna, organizacyjna i prawna; ryzyko przedsięwzięcia.<br><br>4. Proces inżynierii wymagań. Określenie i analiza wymagań. Wymagania stawiane oprogramowaniu i ich dokumentacja. Cechy dobrego wymagania. Metody pozyskiwania wymagań. Podział i klasyfikacja wymagań. Zatwierdzenie wymagań. Zarządzanie wymaganiami.<br><br>5. Modelowanie procesu wytwarzania oprogramowania. Cykl życia projektu informatycznego.<br><br>6. Strategie i procesy prowadzenia projektów informatycznych: tradycyjne (model kaskadowy, model V, prototypowanie, przyrostowy, spiralny)<br><br>7. Zwinne metodyki wytwarzania oprogramowania (Agile, reuse i komponentowość). Programowanie ekstremalne. SCRUM: procesy, artefakty, role. Dobór strategii prowadzenia projektu.<br><br>8. Projektowanie architektoniczne. Strukturalizacja systemu, modele sterowania, rozkład na moduły, architektury charakterystyczne dla różnych produktów informatycznych.<br><br>9. Projektowanie obiektowe. Obiekty i klasy obiektów, procesy projektowania obiektowego, ewolucja projektu.<br><br>10. Język UML Narzędzia CASE do komputerowego wspomagania projektowania oprogramowania.<br><br>11. Projektowanie systemów rozproszonych. Wzorce analizy i projektowania. Klasyfikacja i przykłady.<br><br>12. Projektowanie systemów czasu rzeczywistego. Architektura sprzętu. Projektowanie systemów krytycznych. Analiza bezpieczeństwa i awarii.<br><br>13. Administracja systemem. Konteneryzacja, mikrousługi, chmura obliczeniowa.<br><br>14. Projektowanie dostępu i organizacja danych.<br><br>15. Prototypowanie: środowiska wytwórcze i technologie. AI w projektowaniu systemów.<br><br>Treści przedmiotu - projekt<br>Ćwiczenia projektowe obejmują identyfikację i analizę wymagań oraz modelowanie obiektowe z użyciem narzędzi CASE i projekt architektury. Praca odbywa się w zespołach kilkuosobowych. Każda grupa wykonuje zestaw ćwiczeń w odniesieniu do wybranego przez siebie i przeznaczonego do zinformatyzowania obszaru. Efektem jest kompletna dokumentacja (założenia projektowe, raport wykonalności, specyfikacja wymagań na system) i projekt architektury systemu: diagramy UML, administracja systemem, projekt interfejsu oraz zarządzania danymi a także prototypowy zarys implementacji. |
| Wymagania wstępne i dodatkowe | Brak |
| Sposoby i kryteria oceniania osiąganych efektów uczenia się | |

| Sposób oceniania (składowe) | Próg zaliczeniowy | Składowa oceny końcowej |
|---|---|---|
| Prezentacja | 50.0% | 10.0% |
| Zadania projektowe | 50.0% | 75.0% |
| Sprawdziany do egzaminu | 50.0% | 15.0% |

<table>
<tr><td rowspan="3">Zalecana lista lektur</td><td>Podstawowa lista lektur</td><td>1. Krzysztof Sacha, Inżynieria Oprogramowania, PWN 2022<br><br>2. Ian Sommerville, Inżynieria Oprogramowania, wyd.11, PWN 2019<br><br>3. Max Kanat-Alexander, Zrozumieć oprogramowanie, Helion 2019<br><br>4. Robert C. Martin, Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów, Helion 2022<br><br>5. Mike Cohn, Agile. Metodyki zwinne w planowaniu projektów, Helion 2019<br>6. Robert C. Martin, Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki, Helion 2017</td></tr>
<tr><td>Uzupełniająca lista lektur</td><td>1. Bernd Bruegge, Allen H. Dutoit, Inżynieria oprogramowania w ujęciu obiektowym UML, wzorce projektowe i Java Helion 2011<br><br>2. Keeling Michael, Zostań architektem oprogramowania, PWN 2019<br><br>3. Piotr Gaczkowski, Adrian Ostrowski, Architektura oprogramowania bez tajemnic, Helion 20224.<br><br>4. Praca zbiorowa, Inżynieria oprogramowania w praktyce, PWN, 2022.<br><br>5. Strony domowe do wybranych narzędzi informatycznych. Instrukcje obsługi, przykłady.<br><br>6. Roger S. Pressman, Bruce R. Maxim, Inżynieria oprogramowania. Praktyczne podejście, Helion, 2021 (wyd. 9)</td></tr>
<tr><td>Adresy eZasobów</td><td></td></tr>
<tr><td>Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania</td><td colspan="2">1.Planowanie i ocena wykonalności projektu informatycznego analiza techniczna, ekonomiczna, organizacyjna i ryzyka.<br><br>2. Inżynieria wymagań metody pozyskiwania, dokumentowania i zarządzania wymaganiami oprogramowania.<br><br>3. Cykl życia projektu informatycznego i modele wytwarzania oprogramowania porównanie podejść tradycyjnych i zwinnych.<br><br>4. Projektowanie architektury systemów informatycznych struktura systemu, podział na moduły i style architektoniczne.<br><br>5. Projektowanie obiektowe i modelowanie UML klasy, obiekty oraz diagramy wspomagające projektowanie systemów.<br><br>6. Zwinne metodyki zarządzania projektami IT SCRUM, Agile, programowanie ekstremalne i dobór strategii projektu.<br><br>7. Nowoczesne technologie w inżynierii oprogramowania konteneryzacja, mikrousługi, chmura obliczeniowa i AI w projektowaniu systemów.<br><br>Wycieczka fakultatywna do Centrum Informatycznego Trójmiejskiej Akademickiej Sieci Komputerowej.</td></tr>
<tr><td>Zajęcia praktyczne w ramach przedmiotu</td><td colspan="2">Nie dotyczy</td></tr>
</table>

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.