Uniwersytet Gdański

Karta przedmiotu

Nazwa i kod przedmiotu	Metody programowania, PG_00191646		
Kierunek studiów	Informatyka (O)		
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu	2026/2027
Poziom kształcenia	I stopnia - licencjackie	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	1	Język wykładowy	polski
Semestr studiów	2	Liczba punktów ECTS	3.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca			
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr Tomasz Borzyszkowski	
	Prowadzący zajęcia z przedmiotu		

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	0.0	30.0	75

Cel przedmiotu

Zapoznanie z nowoczesnymi metodami zapewniania jakości w projektach informatycznych poprzez testowanie i walidację oprogramowania.Poznanie wymagań stawianych systemom informatycznym (np. dostępność, wydajność) oraz narzędzi wspierających automatyczne testowanie oprogramowanie. Zapoznanie z praktykami stosowanymi w ramach automatyzacji różnego typu testów (np. testy jednostkowe, integracyjne, interfejsu użytkownika).

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[INFOL3_U06] potrafi dobrać i stosować właściwe metody i narzędzia informatyczne do rozwiązywania złożonych problemów, potrafi wykorzystać posiadaną wiedzę poprzez właściwy dobór źródeł i informacji z nich pochodzących, dokonywanie oceny, krytycznej analizy i syntezy tych informacji	Student potrafi zaplanować proces testowania poprzez formułowanie scenariuszy testowych, dobrać odpowiednie narzędzia do automatyzacji testów oraz zintegrować je z procesem wdrażania oprogramowania (CI/CD), dokonując na podstawie wyników testów krytycznej oceny jakości analizowanego systemu.	[SU2] prezentacja/projekt/referat/raport
[INFOL3_W05] zna i rozumie zaawansowane zagadnienia w zakresie różnych paradygmatów programowania i języków programowania; a także metody i wzorce projektowania i programowania obiektowego	Student charakteryzuje i objaśnia zaawansowane metodologie testowania oprogramowania (w tym model piramidy testów) oraz tłumaczy, w jaki sposób zastosowanie odpowiednich paradygmatów i wzorców projektowych (np. we frameworkach testowych) wpływa na testowalność kodu i proces zapewnienia jakości.	[SW2] prezentacja/projekt/referat/raport

<table>
<tr><td>Treści przedmiotu</td><td>Metodologie i rodzaje testów oprogramowania (piramida testów)

Planowanie testów oprogramowania w ramach cyklu jego wytwarzania (formułowanie scenariuszy testowych)

Automatyzacja testów:

- narzędzia automatyzacji

- testy automatyczne jako sposób zapewnienia jakości w cyklu wytwarzania oprogramowania

- integracja testów automatycznych w proces wdrożenia</td></tr>
<tr><td>Wymagania wstępne i dodatkowe</td><td></td></tr>
<tr><td>Sposoby i kryteria oceniania osiąganych efektów uczenia się</td><td>
<table>
<tr><th>Sposób oceniania (składowe)</th><th>Próg zaliczeniowy</th><th>Składowa oceny końcowej</th></tr>
<tr><td>prezentacja projektu</td><td>51.0%</td><td>100.0%</td></tr>
</table>
</td></tr>
<tr><td>Zalecana lista lektur</td><td>
<table>
<tr><td>Podstawowa lista lektur</td><td>1. Beck Kent, TDD. Sztuka tworzenia dobrego kodu Wydawnictwo Helion, 2020.

2. Harry J.W. Percival. TDD w praktyce. Niezawodny kod w języku Python. Wydawnictwo Helion, 2020.</td></tr>
<tr><td>Uzupełniająca lista lektur</td><td>Brak</td></tr>
<tr><td>Adresy eZasobów</td><td></td></tr>
</table>
</td></tr>
<tr><td>Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania</td><td>1. Korzystając z metodologii TDD, napisz klasę/funkcję o nazwie hamming, obliczająca odległość Hamminga dla podanych danych testowych.

2. Dla zadanych wymagań, scenariuszy i przypadków użycia, zaprojektować i zaimplementować testy behawioralne wykorzystując bibliotekę behave w języku Python.</td></tr>
<tr><td>Praktyki zawodowe w ramach przedmiotu</td><td>Nie dotyczy</td></tr>
</table>

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.