

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2024-2028

Rok akademicki 2026/2027

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Programowanie zespołowe
Kod przedmiotu*	
nazwa jednostki prowadzącej kierunek	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Kierunek studiów	Informatyka i ekonometria
Poziom studiów	studia pierwszego stopnia
Profil	praktyczny
Forma studiów	studia stacjonarne
Rok i semestr/y studiów	rok III, sem. VI
Rodzaj przedmiotu	przedmiot kierunkowy
Język wykładowy	polski
Koordynator	dr inż. Przemysław Pardel
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr inż. Przemysław Pardel, mgr inż. Dawid Kosior

* -opcjonalnie, zgodnie z ustaleniami w Jednostce

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
6	15			30					4

1.2. Sposób realizacji zajęć☒ zajęcia w formie tradycyjnej☐ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość**1.3 Forma zaliczenia przedmiotu (z toku) (egzamin, zaliczenie z oceną, zaliczenie bez oceny)**

Zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Znajomość zagadnień z przedmiotów: programowanie obiektowe, inżynieria oprogramowania, algorytmy i struktury danych, bazy danych

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Poznanie najnowszych narzędzi tworzenia oprogramowania (które mają być używane przez studentów na zajęciach laboratoryjnych).
C ₂	Nauka organizacji pracy zespołu informatycznego w procesie tworzenia, serwisowania i wdrażania oprogramowania
C ₃	Wykonanie przez studentów złożonego, praktycznego projektu informatycznego w grupie.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student ma wiedzę i umiejętności z języka programowania orientowanego obiektowo umożliwiającą realizację projektów.	K_W03, K_U01
EK_02	Student zna i potrafi użyć wybrane narzędzia zespołowego wytwarzania oprogramowania.	K_W03, K_U01, K_K01
EK_03	Student rozumie rolę dokumentacji technicznej zadania informatycznego i ma podstawową wiedzę na temat tworzenia dokumentacji technicznej tworzonego oprogramowania w wybranym narzędziu jej tworzenia.	K_W03, K_U13
EK_04	Student potrafi pracować w zespole nad wspólnym projektem.	K_U01, K_U10, K_U11, K_U13, K_U16, K_K01

3.3 Treści programowe

A. Problematyka wykładu

1. Wprowadzenie do projektowania zespołowego (duża waga i trudności programowania zespołowego, duży wysiłek inżynierii oprogramowania celem opracowania efektywnych metod, cykl produkcji oprogramowania). 2. Metodologia zarządzania Scrum oraz przykładowe narzędzia do zarządzania projektami. 3. Zasady tworzenia dokumentacji technicznej oraz standardów kodowania oprogramowania. 4. Przegląd wybranych metod przydatnych do efektywnego programowania (wybrane wzorce projektowe, asercje, dzienniki, techniki testowania). 5. Ogólne wprowadzenie do narzędzi zespołowego wytwarzania oprogramowania na przykładzie współczesnych narzędzi wytwarzania oprogramowania w języku Java. 6. Narzędzia kompilujące (ANT, MAVEN, GRADLE). 7. Narzędzia kontroli wersji (GIT). 8. Narzędzia ciągłej integracji. 9. Narzędzia mierzenia pokrycia testami jednostkowymi. 10. Narzędzia automatycznego tworzenia dokumentacji technicznej. 11. Docker. 12. Ogólne zalecenia związane z wykorzystaniem powyższych narzędzi do zespołowego tworzenia oprogramowania.
--

B. Problematyka ćwiczeń laboratoryjnych

1. Powtórzenie wiadomości z języka Java.
2. Ustalenie 4-5 osobowych zespołów. Wyłonienie liderów grup. Wybranie tematów projektów.
3. Rozpoczęcie pracy z systemem kontroli wersji.
4. Rozpoczęcie pracy z narzędziem do zarządzania projektami.
5. Ustalenie celu i zakresu systemów tworzonych przez każdą z grup.
6. Ustalenie wstępnych wymagań tworzonego systemu w każdej grupie.
7. Modelowanie tworzonego systemu w każdej grupie.
8. Projektowanie tworzonego systemu w każdej grupie.

9. Implementacja wszystkich modułów systemu, w tym dokumentowanie systemu (JavaDoc) oraz pisanie testów białego-skrzynkowego jednostkowych i integracyjnych dla każdego systemu.
10. Wykonanie dokumentacji systemu.
11. Testowanie systemu.
12. Wdrożenie systemu.
13. Zaliczenie projektów.

3.4 Metody dydaktyczne

Wykład z prezentacją multimedialną

Laboratorium: wykonywanie ćwiczeń tablicowych i programistycznych

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć
EK_01	Kolokwium, obserwacja w trakcie zajęć, projekt	lab
EK_02	Kolokwium, obserwacja w trakcie zajęć, projekt	lab
EK_03	Kolokwium, obserwacja w trakcie zajęć, projekt	lab
EK_04	Projekt, obserwacja w trakcie zajęć	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

- Zaliczenie przedmiotu odbywa się na podstawie dwóch składowych: - zaliczenia wykładu, - oceny z laboratorium. - Zaliczenie laboratorium obejmuje: - projekt grupowy, - publiczną prezentację projektu grupowego, - kolokwium zaliczeniowe - Projekt grupowy polega na opracowaniu rozwiązania konkretnego problemu poprzez przygotowanie i wdrożenie aplikacji oraz publiczną prezentację wyników projektu. - Student zalicza laboratorium, jeżeli: - uzyska ocenę co najmniej 3.0 z projektu, - ocenę co najmniej 3.0 z kolokwium zaliczeniowego. - Student zalicza przedmiot, jeśli: - otrzymał z laboratorium ocenę co najmniej 3.0. Ocenę dst. z kolokwium/projektu student uzyskuje w przypadku uzyskania minimum połowy możliwych do zdobycia punktów. Kolejne oceny równomiernie pokrywają skalę punktową.
--

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	45
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	2
Godziny niekontaktowe – praca własna studenta	55

(przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	
SUMA GODZIN	102
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1. Wykłady Jana Bazana dostępne na MS Teams.. 2. J. Ferguson Smart, Java: praktyczne narzędzia, Helion, Gliwice (2009) 3. Schildt, H.: Java. Przewodnik dla początkujących, wydanie VI, Gliwice: Helion (2015). 4. Horstmann, C., S., Cornell, G.: Java, Podstawy, wydanie IX, Gliwice: Helion (2016). 5. Kane, S.,P., Matthias, K.: Docker – praktyczne zastosowania, wydanie II, Gliwice: Helion (2019).
Literatura uzupełniająca: 1. Hunt, D. Thomas, JUnit: pragmatyczne testy jednostkowe w Javie, Helion, Gliwice, (2006). 2. Horstmann, C., S., Cornell, G.: Java, Techniki zaawansowane, wydanie IX, Gliwice: Helion (2017).

Akceptacja Kierownika Jednostki lub osoby upoważnionej