**SYLABUS**

DOTYCZY CYKLU KSZTAŁCENIA 2022 - 2026

Rok akademicki 2024/2025

## 1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

| | |
|---|---|
| Nazwa przedmiotu | *programowanie zespołowe* |
| Kod przedmiotu | |
| Nazwa jednostki prowadzącej kierunek | *Kolegium Nauk Przyrodniczych* |
| Nazwa jednostki realizującej przedmiot | *Kolegium Nauk Przyrodniczych* |
| Kierunek studiów | *informatyka* |
| Poziom studiów | *studia inżynierskie I-go stopnia* |
| Profil | *ogólnoakademicki* |
| Forma studiów | *stacjonarne* |
| Rok i semestr/y studiów | *rok III semestr 6* |
| Rodzaj przedmiotu | *inżynierski przedmiot kierunkowy* |
| Język wykładowy | *język polski* |
| Koordynator | *dr hab. Jan Bazan, prof UR* |
| Imię i nazwisko osoby prowadzącej / osób prowadzących | *dr hab Jan Bazan, prof. UR, dr inż. Marcin Ochab, mgr inż. Dawid Kosior* |

### 1.1.Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

| Semestr (nr) | Wykł. | Ćw. | Konw. | Lab. | Sem. | ZP | Prakt. | Projekt | **Liczba pkt. ECTS** |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 15 | | | 30 | | | | 15 | 5 |

### 1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

### 1.3 Forma zaliczenia przedmiotu (z toku)

dwa zaliczenia z oceną (pierwsze z laboratorium i drugie z zajęć projektowych)

## 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

| | |
|---|---|
| C1 | Poznanie najnowszych narzędzi tworzenia oprogramowania (które mają być używane przez studentów na zajęciach laboratoryjnych). |
| C2 | Nauka organizacji pracy zespołu informatycznego w procesie tworzenia, serwisowania i wdrażania oprogramowania |
| C3 | 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 [1] |
|---|---|---|
| EK_01 | Student ma wiedzę i umiejętności z języka programowania orientowanego obiektowo umożliwiającą realizację projektów. | K_W04, K_W07, K_U11, K_U12, K_U14 |
| EK_02 | Student zna i potrafi użyć wybrane narzędzia zespołowego wytwarzania oprogramowania. | K_W04, K_W06, K_W07, K_U11 |
| 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_W12, K_U07, K_U19, K_U21 |
| EK_04 | Student potrafi pracować w zespole nad wspólnym projektem. | K_U18, K_U19, K_U21 |

### 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. Narzędzia zespołowego wytwarzania oprogramowania na przykładzie współczesnych narzędzi wytwarzania oprogramowania w języku Java.
    a. Narzędzia kompilujące.
    b. Narzędzia kontroli wersji.
    c. Narzędzia ciągłej integracji.
    d. Narzędzia mierzenia pokrycie testami jednostkowymi.

[1] W przypadku ścieżki kształcenia prowadzącej do uzyskania kwalifikacji nauczycielskich uwzględnić również efekty uczenia się ze standardów kształcenia przygotowującego do wykonywania zawodu nauczyciela.

| |
|---|
| e. Narzędzie testów integracyjnych dla baz danych.<br>f. Narzędzia testów obciążeniowych i wydajnościowych.<br>g. Narzędzia testowania usług sieciowych.<br>h. Narzędzia testowania interfejsów użytkownika.<br>i. Narzędzia kontrolowania i wymuszania standardów kodowania.<br>j. Narzędzia zarządzenia problemami.<br>k. Narzędzia automatycznego tworzenia dokumentacji technicznej. |
| 6. Ogólne zalecenia związane z wykorzystaniem powyższych narzędzi do zespołowego tworzenia oprogramowania. |

B. Problematyka ćwiczeń laboratoryjnych

| |
|---|
| 1. Powtórzenie wiadomości z podstaw języka Java. |
| 2. Kolokwium z podstaw języka Java. |
| 3. Powtórzenie wiadomości ze standardów kodowania w języku Java. |
| 4. Kolokwium ze stosowania standardów kodowania. |
| 5. Praca z systemem kontroli wersji. |
| 6. Kolokwium ze znajomości pracy z systemem kontroli wersji. |
| 7. Powtórzenie wiadomości z języka SQL, ze szczególnym uwzględnieniem wykorzystywania go na poziomie języka Java. |
| 8. Kolokwium z języka SQL. |
| 9. Wybrane zaawansowane elementy programowania w języku Java. |
| 10. Kolokwium z zaawansowanych elementów programowania w języku Java. |
| 11. Zaliczenie przedmiotu. |

C. Problematyka zajęć projektowych

| |
|---|
| 1. Ustalenie 4-5 osobowych zespołów. Wyłonienie liderów grup. Wybranie tematów projektów. |
| 2. Ustalenie celu i zakresu systemów tworzonych przez każdą z grup. |
| 3. Ustalenie wstępnych wymagań tworzonego systemu w każdej grupie. |
| 4. Modelowanie tworzonego systemu w każdej grupie. |
| 5. Projektowanie tworzonego systemu w każdej grupie. |
| 6. Implementacja wszystkich modułów systemu, w tym dokumentowanie systemu oraz pisanie testów biało-skrzynkowych jednostkowych i integracyjnych. |
| 7. Wykonanie dokumentacji systemu. |
| 8. Testowanie systemu |
| 9. Zaliczenie projektów. |

### 3.4 Metody dydaktyczne

Wykład z prezentacją multimedialną
Laboratorium: wykonywanie ćwiczeń tablicowych i programistycznych
Zajęcia projektowe: praca nad projektem.

## 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 dydaktycznych (w, ćw, ...) |
|---|---|---|
| EK_01 | Dwa kolokwia przy komputerze z programowania obiektowego (podstawy Javy oraz zaawansowana Java). Kolokwium praktyczne przy komputerze z tworzenia złożonych kwerend SQL. | lab |
| EK_02 | Kolokwium praktyczne przy komputerze z ogólnej znajomości narzędzia GIT. | lab |
| EK_03 | Pisemne kolokwium ze stosowania standardów kodowania i dokumentowania. | lab |
| EK_ 04 | Weryfikacja repozytoriów GitLab oraz tablic sprintów Jira podczas przeglądów sprintu oraz przy zaliczeniu projektu. Ponadto, uwzględnia się tutaj stopień systematyczności pracy nad projektem. | projekt |

### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

| Efekt | Ocena | Kryteria otrzymania oceny |
|---|---|---|
| EK_01 | dst | Student ma wiedzę i umiejętności z języka programowania orientowanego obiektowo umożliwiającą realizację projektów. W szczególności wie jak i potrafi utworzyć prosty interfejs graficzny użytkownika, w elementarny sposób skorzystać z bazy danych z poziomu języka oraz utworzyć i użyć prostą bibliotekę funkcji. |
| | db | Student ma wiedzę i umiejętności z języka programowania orientowanego obiektowo umożliwiającą realizację projektów. W szczególności wie jak i potrafi utworzyć interfejs graficzny użytkownika zawierający bardziej zaawansowane elementy, rutynowo korzystać z bazy danych z poziomu języka oraz utworzyć i użyć bardziej zaawansowana bibliotekę funkcji (np. z użyciem prostych standardowych struktur danych lub zaawansowanych metod konstruowania algorytmów). |
| | bdb | Student ma wiedzę z języka programowania orientowanego obiektowo umożliwiającą realizację projektów. W szczególności wie jak i potrafi utworzyć interfejs graficzny użytkownika zawierający zaawansowane elementy uwzględniając aspekty intuicyjności i prostoty w użytkowaniu, optymalnie skorzystać z bazy danych z poziomu języka oraz utworzyć i użyć bardziej zaawansowana bibliotekę funkcji (np. z użyciem optymalnie dobranych standardowych struktur danych lub zaawansowanych metod konstruowania algorytmów). |

| | | |
|---|---|---|
| EK_02 | dst | Student zna narzędzia zespołowego wytwarzania oprogramowania i potrafi użyć wybrane narzędzie zespołowego wytwarzania oprogramowania. |
| | db | Student zna narzędzia zespołowego wytwarzania oprogramowania, potrafi porównać dostępne narzędzia oraz potrafi użyć co najmniej dwa wybrane narzędzie zespołowego wytwarzania oprogramowania. |
| | bdb | Student zna narzędzia zespołowego wytwarzania oprogramowania, potrafi je porównać oraz wskazać optymalne do prac nad danym projektem. Potrafi także użyć jednocześnie (w kooperacji ze sobą) co najmniej dwa wybrane narzędzie zespołowego wytwarzania oprogramowania. |
| EK_03 | dst | Student rozumie rolę dokumentacji technicznej zadania informatycznego i ma podstawową wiedzę na temat tworzenia dokumentacji technicznej tworzonego oprogramowania w wybranym narzędziu jej tworzenia. |
| | db | Student rozumie rolę dokumentacji technicznej zadania informatycznego i ma rozszerzoną wiedzę na temat tworzenia dokumentacji technicznej tworzonego oprogramowania w wybranym narzędziu jej tworzenia. Jednak nie zawsze rozumie potrzebę i użyteczność stosowania niektórych zaawansowanych elementów tworzenia dokumentacji. |
| | bdb | Student rozumie rolę dokumentacji technicznej zadania informatycznego i ma rozszerzoną wiedzę na temat tworzenia dokumentacji technicznej tworzonego oprogramowania w wybranym narzędziu jej tworzenia oraz rozumie potrzebę i użyteczność stosowania zaawansowanych elementów tworzenia dokumentacji. |
| EK_04 | dst | Student potrafi pracować w zespole nad wspólnym projektem, przyjmując ustaloną rolę. Jednak potrafi w pełni zrealizować tylko podstawowe zadania wyspecyfikowane przez prowadzącego w ramach prac projektowych według współczesnych standardów znanych z literatury. Ponadto, postępy prac nad projektem są skokowe (brak systematyczności). |
| | db | Student potrafi pracować w zespole nad wspólnym projektem, przyjmując różne role. Jednak nie potrafi w pełni zrealizować wszystkich zadań wyspecyfikowanych przez prowadzącego w ramach prac projektowych według współczesnych standardów znanych z literatury. Nad projektem pracuje raczej systematycznie, tzn. na każdych zajęciach można zaobserwować istotne postępy. |
| | bdb | Student potrafi pracować w zespole nad wspólnym projektem, przyjmując różne role. Potrafi także w pełni zrealizować wszystkie zadania wyspecyfikowane przez prowadzącego w ramach prac projektowych według współczesnych standardów znanych z literatury. Nad projektem pracuje systematycznie. |

**Zasady uzyskania oceny końcowej:**

Zaliczenie laboratorium następuje na podstawie zaliczenia efektów EK_01 – EK_03 weryfikowanych przez planowane w danym okresie metody weryfikacji. Przy czym zakłada się, że każda metoda weryfikacji dostarcza osobne oceny dla każdego z weryfikowanych przez nią efektów uczenia się. Jeśli dany efekt jest weryfikowany przez więcej niż jedną metodę, to ocena weryfikująca osiągnięcie tego efektu jest obliczana jako średnia arytmetyczna ocen uzyskanych w poszczególnych metodach weryfikowania tego efektu.

Zaliczenie zajęć projektowych następuje na podstawie zaliczenia efektu EK_04, co jest weryfikowane przez planowane metody weryfikacji (patrz wyżej).

Zaliczenie wykładu i przedmiotu następuje na podstawie zaliczonych w trakcie laboratoriów i zajęć projektowych EK_01 – EK_04.

Zarówno w przypadku oceny z laboratorium jak i z zajęć projektowych obowiązują następujące zasady.

Student otrzymuje z zaliczenia ocenę **niedostateczny,** gdy metody weryfikacji wykażą, iż co najmniej jeden z efektów wymaganych do osiągnięcia nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę **dostateczny,** gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę **dobry,** gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę **bardzo dobry,** gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 4.75

## 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 | 60 |
| Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie) | |
| Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.) | 65 |
| SUMA GODZIN | 125 |
| **SUMARYCZNA LICZBA PUNKTÓW ECTS** | **5** |

## 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

| | |
|---|---|
| wymiar godzinowy | - |
| zasady i formy odbywania praktyk | - |

## 7. LITERATURA

Literatura podstawowa:


1. Prezentacje wykładowe Jana Bazana dostępne na MS Teams.
2. Oficjalne witryny internetowe dotyczące technologii: JavaFX, JDBC, JUnit.
3. J. Ferguson Smart, Java: praktyczne narzędzia, Helion, Gliwice (2009)
4. Schildt, H.: Java. Przewodnik dla początkujących, wydanie VI, Gliwice: Helion (2015).
5. Horstmann, C., S., Cornell, G.: Java, Podstawy, wydanie IX, Gliwice: Helion (2016).


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).