

PROGRAMOWANIE ZESPOŁOWE

Kod przedmiotu: IO-PGZ2

Rodzaj przedmiotu: kierunkowy, obieralny

Specjalność: Inżynieria programowania

Wydział: Informatyki

Kierunek: Informatyka

Poziom studiów: drugiego stopnia – VII poziom PRK

Profil studiów: praktyczny

Forma studiów: stacjonarna/niestacjonarna

Rok: 1

Semestr: 1

Formy zajęć i liczba godzin:

Forma stacjonarna

 wykłady – 15

 laboratorium – 35

Forma niestacjonarna

 wykłady – 10

 laboratorium – 18

Zajęcia prowadzone są w języku polskim.

Liczba punktów ECTS: 3

Osoby prowadzące:

 wykład:

 laboratorium:

1. Założenia i cele przedmiotu:

Realizacja oprogramowania dedykowanego dla rzeczywistych zastosowań wymaga zwykle zaangażowania wielu programistów. Zespołowe tworzenie oprogramowania stawia nowe wymagania przed programistami — umiejętność pracy grupowej. Ten rodzaj pracy wymaga wykorzystania dodatkowych narzędzi wspomagających, obejmujących repozytoria kodu, systemy kontroli wersji, systemy zgłaszania błędów i usterek, oraz środowiska programistyczne pozwalające na wykorzystanie tych narzędzi. Moduł Programowanie zespołowe ukierunkowany jest na wyrobienie praktycznych umiejętności programowania zespołowego z wykorzystaniem współczesnych metod i narzędzi.

2. Określenie przedmiotów wprowadzających wraz z wymaganiami wstępnymi:

Przedmioty wprowadzające to: Programowanie obiektowe, Systemy baz danych.

3. Opis form zajęć

a) *Wykłady*

• **Treści programowe:**

- Analiza problemu projektowego opracowanie studium wykonalności.
- Opracowywanie specyfikacji wymagań.
- Zasady organizacji zespołu projektowego, role, kompetencje.
- Zasady ustalania harmonogramu realizacji projektu zespołowego.
- Rola metodyki programowania, harmonogram a wybrana metodyka.
- Weryfikacja postępu realizacji projektu.
- Synchronizacja prac projektowych, narzędzia wspomagające.
- Synchronizacja prac programistycznych, repozytoria kodu, zasady ich wykorzystania.
- Integracja ciągła systemu, koncepcja, zasady, narzędzia.
- Testowanie i weryfikacja oprogramowania.
- Zasady postępowania w sytuacjach konfliktowych.

• **Metody dydaktyczne:**

- Wykład prowadzony metodą tradycyjną z wykorzystaniem rzutnika multimedialnego, z wykorzystaniem materiałów udostępnianych studentom w postaci elektronicznej.

• **Forma i warunki zaliczenia:**

- Pozytywna ocena testu zaliczeniowego realizowanego na ostatnim wykładzie w semestrze.

• **Wykaz literatury podstawowej:**

1. P. Wróblewski. Zarządzanie projektami z wykorzystaniem darmowego oprogramowania. Helion, ostatnie wydanie.
2. Weisfeld M.: Myślenie obiektowe w programowaniu. Gliwice: Helion, cop. 2020.
3. J. Rumbaugh G. Booch, I. Jacobson. The Unified Modeling Language User Guide (2nd Edition). Addison-Wesley Professional, ostatnie wydanie.
4. R. Johnson John M. Vlissides E. Gamma, R. Helm. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman, ostatnie wydanie.
5. Włodzimierz Gajda, Git. Rozproszony system kontroli wersji, Helion, ostatnie wydanie.

• **Wykaz literatury uzupełniającej:**

1. Peter Bell, Brent Beer, GitHub. Przyjazny przewodnik, Helion, ostatnie wydanie.
2. Strona: <https://git-scm.com/book/pl/v1/Pierwsze-kroki-Podstawy-Git>.
3. Jarosław Krochmalski, Developing with Docker, Packt

b) *Laboratorium*

• **Treści programowe:**

- Dobór tematyki zakresu projektu i określanie wymagań wstępnych.
- Organizacja zespołu projektowego, ustalanie ról — studium przypadku.
- Przygotowanie harmonogramu zadań — studium przypadku.
- Realizacja kolejnych zadań związanych z projektem.
- Praca grupowa z narzędziami do zarządzania projektem i wersjonowania kodu.

- Zarządzanie zmianami, wykorzystanie refaktoryzacji w projekcie zespołowym.
- **Metody dydaktyczne:**
 - Prezentacja treści i dyskusja moderowana.
 - Metoda problemowa – studium przypadku, burza mózgów.
 - Metoda laboratoryjna –ćwiczenia laboratoryjne z wykorzystaniem komputerów.
- **Forma i warunki zaliczenia:**
 - Pozytywna ocena realizacji wskazanych zadań.
 - Pozytywna ocena aktywności studenta podczas zajęć, w tym ocena biegłości w zespołowym tworzeniu programów.
- **Wykaz literatury podstawowej:**
 - Jak w przypadku wykładu.
- **Wykaz literatury uzupełniającej:**
 - Jak w przypadku wykładu.

4. Opis sposobu wyznaczania punktów ECTS

a. forma stacjonarna

Forma zajęć	Formy aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności
Wykład	Kontakt z nauczycielem	15
	Czytanie wskazanej literatury	5
Laboratorium	Kontakt z nauczycielem	35
	Projekt grupowy	20

Całkowita ilość godzin aktywności studenta	75
Liczba punktów ECTS dla modułu/przedmiotu	3

b. forma niestacjonarna

Forma zajęć	Formy aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności
Wykład	Kontakt z nauczycielem	10
	Czytanie wskazanej literatury	10
Laboratorium	Kontakt z nauczycielem	18
	Projekt grupowy	30

Całkowita ilość godzin aktywności studenta	75
Liczba punktów ECTS dla modułu/przedmiotu	3

5. Wskaźniki sumaryczne

a. forma stacjonarna

- a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich

- Liczba godzin kontaktowych – 50
 - Liczba punktów ECTS – 2,0
- b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.
- Liczba godzin kontaktowych – 35
 - Liczba punktów ECTS – 2,2

b. forma niestacjonarna

- a) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich
- Liczba godzin kontaktowych – 28
 - Liczba punktów ECTS – 1,2
- b) liczba godzin dydaktycznych (tzw. kontaktowych) i liczba punktów ECTS na zajęciach o charakterze praktycznym.
- Liczba godzin kontaktowych – 18
 - Liczba punktów ECTS – 2,2

6. Zakładane efekty uczenia się.

Efekt przedmiotowy (Symbol)	Efekty uczenia się dla przedmiotu	Odniesienie do kierunkowych efektów uczenia się
IO-PGZ2_W1	Student posiada wiedzę na temat zarządzania zespołem programistycznym oraz zna i rozumie rolę metodyk programowania w projekcie zespołowym.	IIK_W02 IIK_W04
IO-PGZ2_W1	Student rozumie znaczenie harmonogramów realizacji prac, zna i rozumie rolę członków zespołu projektowego.	IIK_W06 IIK_W08
IO-PGZ2_U1	Student potrafi zorganizować zespół, dobrać metodykę realizacji projektu, opracować harmonogram, wcielać się w różne role członków zespołu projektowego.	IIK_U04 IIK_U12
IO-PGZ2_U2	Potrafi stosować narzędzia wspomagające pracę grupową, potrafi korzystać z systemów kontroli wersji, repozytoriów kodu.	IIK_U07 IIK_U09 IIK_U10
IO-PGZ2_K1	Potrafi współpracować w zespole programistycznym, rozumie rolę dobrej współpracy i jej wpływ na terminową realizację projektu.	IIK_K01 IIK_K02 IIK_K03

7. Odniesienie efektów uczenia się do form zajęć i sposób oceny osiągnięcia przez studenta efektów uczenia się.

Efekt przedmiotowy (Symbol)	Forma zajęć		Sposób sprawdzenia osiągnięcia efektu
	Wykład	Laboratorium	
IO-PGZ2_W1	ν		Test zaliczeniowy
IO-PGZ2_W2	ν	ν	Test zaliczeniowy
IO-PGZ2_U1	ν	ν	Prace kontrolne
IO-PGZ2_U2		ν	Prace kontrolne
IO-PGZ2_K1		ν	Prace kontrolne, ocena aktywności

8. Kryteria uznania osiągnięcia przez studenta efektów uczenia się.

Efekt przedmiotowy (Symbol)	Efekt jest uznawany za osiągnięty, gdy student:
IO-PGZ2_W1	Zalicza ponad 50% pytań w teście sprawdzającym.
IO-PGZ2_W2	Poprawnie rozwiązuje zadania w czasie zajęć. Zalicza ponad 50% pytań w teście sprawdzającym.
IO-PGZ2_U1	Osiąga ponad 50% punktów w pracach kontrolnych.
IO-PGZ2_U2	Osiąga ponad 50% punktów w pracach kontrolnych.
IO-PGZ2_K1	Osiąga ponad 50% punktów w pracach kontrolnych.