

KARTA PRZEDMIOTU DLA NABORU 2023/2024
FORMA STUDIÓW: STACJONARNA/NIESTACJONARNA

INFORMACJE OGÓLNE

1. Nazwa przedmiotu Architektura oprogramowania

2. Nazwa kierunku Informatyka

3. Poziom kształcenia studia drugiego stopnia

4. Liczba punktów ECTS 4

5. Liczba godzin w semestrze

semestr	W S/NS	ćw	lab/lek S/NS	prj/zp S/NS	prk
II	15/9		15/9	15/9	

6. Język wykładowy polski

7. Wykładowca mgr inż. Andrzej Jasiński

INFORMACJE SZCZEGÓŁOWE

8. Wymagania wstępne

1. Znajomość podstaw programowania
2. Znajomość podstaw inżynierii oprogramowania

9. Cele przedmiotu

C1 Zapoznanie studentów z pojęciami i zagadnieniami architektury oprogramowania

C2 Nauczenie studentów projektowania systemów informatycznych (oprogramowania)

10. Efekty uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych

Student, który zaliczył przedmiot:	odniesienie do kierunkowych efektów uczenia się
------------------------------------	---

WIEDZA

EU01	Objaśnia pojęcia i zagadnienia architektury oprogramowania	K_W02, K_W04
EU02	Omawia cele i cechy dobrej architektury	K_W02, K_W04

UMIEJĘTNOŚCI

EU03	Stosuje zasady SOLID	K_U02, K_U03
EU04	Ocenia jakość architektury oprogramowania	K_U01
EU05	Projektuje moduły oprogramowania	K_U02, K_U03
EU06	Tworzy architekturę systemu informatycznego (oprogramowania)	K_U02, K_U03

KOMPETENCJE SPOŁECZNE

EU07	Stale poszerza swoją wiedzę z architektury oprogramowania	K_K01
EU08	Projektuje oprogramowanie, uwzględniając kwestie nietechniczne	K_K03

11. Treści programowe

Forma zajęć – wykłady/ ćwiczenia/laboratoria/zajęcia praktyczne itp.

Wykłady

- 1) Czym jest architektura oprogramowania i czemu służy?
- 2) Paradygmaty programowania.
- 3) Architektura a „projekt” (ang. design) – zasady SOLID.
- 4) Cechy architektury – na co wpływa wybór architektury?
- 5) Miary jakości architektury – nie ma uniwersalnych, idealnych rozwiązań.
- 6) Moduły i komponenty.
- 7) Granice i komunikacja pomiędzy komponentami.
- 8) Style architektoniczne.
- 9) Czysta architektura.
- 10) Dlaczego bazy danych, frameworki i chmura nie powinny interesować architekta?
- 11) Kompetencje miękkie w pracy architekta oprogramowania.

Laboratoria

- 1) Single Responsibility Principle.
- 2) Open-Closed Principle.
- 3) Liskov Substitution Principle.
- 4) Interface Segregation Principle.
- 5) Dependency Inversion Principle.

Projekt

- 1) Zaprojektowanie systemu informatycznego dla konkretnego przypadku biznesowego.

12. Narzędzia/metody dydaktyczne

1. Wykłady w formie prezentacji
2. Platforma Microsoft Teams
3. Monitor interaktywny i komputery z oprogramowaniem biurowym i specjalistycznym
4. Konsultacje

13. Sposoby oceny (częstkowe, końcowe)

1. Każde zadanie laboratoryjne wykonane przez studenta jest oceniane.
2. Zadanie projektowe jest oceniane na koniec semestru.
3. Na koniec semestru przeprowadzany jest egzamin.

14. Obciążenie pracą studenta

Forma aktywności	liczba godzin S/NS
1. Zajęcia z bezpośrednim udziałem nauczyciela oraz konsultacje	47/27
2. Nakład pracy studenta	53/73
suma	100
liczba punktów ECTS	4

15. Literatura

Literatura podstawowa:

1. Robert C. Martin, *Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów*, Helion, 2018.
2. Mark Richards, Neal Ford, *Podstawy architektury oprogramowania dla inżynierów*, Helion, 2020.

Literatura uzupełniająca:

1. Len Bass, Paul Clements, Rick Kazman, *Architektura oprogramowania w praktyce. Wydanie IV*, Helion, 2022.
2. Vlad Khononov, *Koncepcja Domain-Driven Design. Dostosowywanie architektury aplikacji do strategii biznesowej*, Helion, 2022.
3. Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghani, *Złożone zagadnienia architektury oprogramowania. Jak analizować kompromisy i podejmować trudne decyzje*, Helion, 2023.

16. Formy oceny – szczegóły

Warunki uzyskania zaliczenia przedmiotu: zajęcia kończą się zaliczeniem z oceną

Zaliczenie laboratorium

Na każdych zajęciach laboratoryjnych studenci realizują zadanie przydzielone przez prowadzącego. Jest ono oceniane w skali od 2 do 6. Studenci, którzy nie zdążyli skończyć zadania na zajęciach, mogą je dokończyć w domu i oddać na kolejnych zajęciach (za każdy tydzień opóźnienia, ocena jest obniżana o pół stopnia).

Ocena końcowa z laboratorium jest wystawiana na podstawie średniej z ocen cząstkowych.

Średnia wymagana na poszczególne oceny:

< 2,50	–	2.0 (ndst)
2,50 - 3,24	–	3.0 (dst)
3,25 - 3,74	–	3.5 (dst+)
3,75 - 4,24	–	4.0 (db)
4,25 - 4,74	–	4.5 (db+)
> 4,74	–	5.0 (bdb)

Zaliczenie projektu

Studenci realizują w ciągu semestru projekt, który składa się z następujących etapów:

- 1) Zaprojektowanie systemu informatycznego – podział na komponenty (40 pkt)
- 2) Zaprojektowanie wybranych komponentów (40 pkt)
- 3) Prezentacja projektu (20 pkt)

Ocena końcowa z projektu wynika z łącznej ilości punktów, uzyskanych za poszczególne etapy:

< 50	–	2.0 (ndst)
50 - 59	–	3.0 (dst)
60 - 69	–	3.5 (dst+)
70 - 79	–	4.0 (db)
80 - 89	–	4.5 (db+)
> 89	–	5.0 (bdb)

Zaliczenie wykładu

Na koniec semestru studenci piszą egzamin, który sprawdza ich wiedzę i umiejętności. Czas trwania egzaminu to 30 minut. Większość pytań ma charakter otwarty. Na podstawie punktów uzyskanych z egzaminu wystawiana jest ocena na koniec semestru.

% uzyskanych punktów wymagany na poszczególne oceny:

0% - 50%	–	2.0 (ndst)
50% - 59%	–	3.0 (dst)
60% - 69%	–	3.5 (dst+)
70% - 79%	–	4.0 (db)
80% - 89%	–	4.5 (db+)
90% - 100%	–	5.0 (bdb)

17. Inne przydatne informacje o przedmiocie

1. Materiały z wykładu są umieszczane na platformie Microsoft Teams
2. Zajęcia odbywać się będą w Akademii Bialskiej Nauk Stosowanych im. Jana Pawła II lub na platformie e-learningowej
3. Zajęcia odbywać się będą zgodnie z aktualnym planem zajęć
4. Konsultacje odbywać się będą zgodnie z obowiązującym terminarzem