Akademia Nauk Stosowanych

im. Hipolita Cegielskiego w Gnieźnie Uczelnia Państwowa

SYLABUS

Pozycja przedmiotu w planie:		R.III/S.V - 5
1. **OGÓLNY OPIS PRZEDMIOTU**		
1	Nazwa modułu	Moduł zajęć kierunkowych
2	Nazwa przedmiotu	Inżynieria oprogramowania II
3	Kierunek studiów	Informatyka
4	Poziom studiów	studia pierwszego stopnia
5	Forma studiów	niestacjonarne
6	Profil studiów	praktyczny
7	Rok studiów	trzeci
8	Semestr przedmiotu	piąty
9	Jednostka prowadząca kierunek studiów	Instytut Nauk Technicznych
10	Liczba punktów ECTS	3
11	Sposób zaliczenia:	Wykład: zaliczenie z oceną Projekt: zaliczenie z oceną
12	Imię i nazwisko nauczyciela (li) akademickiego (ich), stopień lub tytuł naukowy, adres e-mail	dr inż. Piotr Sujka p.sujka@ans-gniezno.edu.pl
13	Imię i nazwisko koordynatora(ów) przedmiotu, stopień lub tytuł naukowy, adres e-mail	dr inż. Piotr Sujka p.sujka@ans-gniezno.edu.pl
14	Język wykładowy	polski
15	Tryb prowadzenia zajęć	W salach i zdalny
16	Sposób prowadzenia zajęć	Synchroniczny i asynchroniczny
17	Narzędzia informatyczne wykorzystywane do prowadzenia zajęć, udostępniania materiałów i komunikacji ze studentami	Platforma Moodle. MS Teams
18	Przedmioty wprowadzające	Inżynieria oprogramowania I
19	Wymagania wstępne	1. Podstawowe wiadomości z zakresu technik programowania 2. Umiejętność efektywnego samokształcenia w dziedzinach związanych z informatyką jako wybranym kierunkiem studiów 3. Student ma świadomość konieczności poszerzania swoich kompetencji oraz gotowość do podjęcia współpracy w ramach zespołu.
20	**Cele przedmiotu:**	
C1	Poznanie zasad i metod wytwarzania oprogramowania w sposób systematyczny. Poznanie zasad zarządzania zespołami programistycznymi.	
C2	Student przyswaja sobie umiejętności: posługiwania się wzorcami projektowymi; projektowania oprogramowania zgodnie z metodyką obiektową.	

C3	Dokonywanie przeglądu projektu oprogramowania.
21	Forma zajęć, liczba godzin wymagająca bezpośredniego udziału nauczyciela akademickiego, liczba godzin nakładu pracy studenta

Forma zajęć	Liczba godzin
1. Wykład	16
2. Projekt	8
Suma godzin	24

lp.	**Całkowity nakład pracy studenta**	
1.	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi:	**Godzinowe obciążenie studenta**
	Udział w wykładach 16 godzin Udział w projektach 8 godzin	24 godzin
	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi 24 godziny, co odpowiada 0,75 punktom ECTS.	
2	Bilans nakładu pracy studenta: Samodzielne studiowanie literatury: 36 godzin Przygotowanie do zaliczenia z: 5 godzin Wykonywanie projektów: 15 godzin Łączny nakład pracy studenta wynosi 56 godzin, co odpowiada 2,25 punktom ECTS.	56 godzin
3	**Łączny nakład pracy studenta (pozycja 1+2)**	80 godzin
4	**Punkty ECTS za przedmiot**	3 ECTS
5	Liczba punktów ECTS, którą student musi osiągnąć w ramach zajęć o charakterze praktycznym w tym zajęć laboratoryjnych, warsztatowych, projektowych	1 ECTS

Efekty uczenia się - wiedza	K_W17: Student ma wiedzę w zakresie projektowania oprogramowania, narzędzi i środowisk wytwarzania oprogramowania, procesów wytwarzania oprogramowania; ma wiedzę w zakresie specyfikacji wymagań, walidacji i testowania oprogramowania, zna metody zarządzania przedsięwzięciami programistycznymi oraz ich jakością. K_W18: Student ma wiedze o cyklu życia oprogramowania, oraz cyklu życia układów cyfrowych, sprzętu komputerowego, sieciowego i systemów wbudowanych, ma wiedzę o stanie oraz najnowszych trendach i uwarunkowaniach rozwojowych informatyki, elektroniki, automatyki, robotyki i systemów ICT (IoT, BIoT, IIoT). K_W19: Student ma podstawową wiedzę w zakresie budowy relacyjnych i nierelacyjnych systemów baz danych, modelowania danych, projektowania relacyjnych i nierelacyjnych bazy danych, języków zapytań do baz danych, definicji danych oraz przetwarzania transakcji.
Efekty uczenia się - umiejętności	K_U07: Student potrafi utworzyć specyfikację, zaprojektować i zaimplementować system informatyczny, cyfrowy lub wbudowany z zastosowaniem wybranych narzędzi wspierających budowę oprogramowania, wzorców projektowych, zgodnie z opracowanym harmonogramem. K_U08: Student ma umiejętności: posługiwania się wzorcami projektowymi; projektowania oprogramowania zgodnie z metodyką obiektową; dokonywania przeglądu projektu oprogramowania; wybierania narzędzi wspomagających budowę oprogramowania; doboru modelu procesu wytwarzania oprogramowania do specyfiki przedsięwzięcia; specyfikowania wymagań dotyczących oprogramowania i przeprowadzania ich przeglądu; tworzenia, oceny i realizacji planu testowania; uczestniczenia w inspekcji kodu; zarządzania konfiguracją oprogramowania; opracowywania planu przedsięwzięcia dotyczącego budowy oprogramowania.
Efekty uczenia się – kompetencje społeczne	K_K01: Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, krytycznie odnosi się do posiadanej wiedzy, podnoszenia kompetencji zawodowych, osobistych i

	społecznych. K_K02: Student ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka, w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje; dba o dobre tradycje zawodu informatyka. K_K04: Student ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania, potrafi określić priorytety służące realizacji określonego przez siebie lub zespół zadania.

2. **TREŚCI PROGRAMOWE ODNIESIONE DO EFEKTÓW UCZENIA SIĘ**

Forma zajęć	Treści programowe	liczba godzin
Forma: Wykład		
W1	Zarządzanie przedsięwzięciem programistycznym.	2
W2	Szacowanie złożoności, czasu i rozmiaru produktu.	3
W3	Metodyki tworzenia oprogramowania, ewolucja oprogramowania.	3
W4	Projektowanie obiektowe.	3
W5	Przegląd wybranych technologii programowych.	3
W6	Zarządzanie zespołami programistów.	2
Forma: Projekt		
P1	Identyfikacja klas i obiektów.	1
P2	Karty CRC.	2
P3	Przypadki użycia.	1
P4	Projekt struktur i hierarchii klas.	2
P5	Testowanie oprogramowania obiektowego.	2

3. **Literatura**

Literatura podstawowa	1. Sommerville I., *Inżynieria oprogramowania.* WNT, Warszawa 2005. 2. Sacha, Inżynieria oprogramowania, PWN, 2010.
Literatura uzupełniająca	1. Pressman R., Praktyczne podejście do inżynierii oprogramowania. WNT, Warszawa 2004. 2. Jaszkiewicz A., Inżynieria oprogramowania. Helion, Gliwice 1997

4. **Metody dydaktyczne**

Forma	Metody dydaktyczne
Wykład	Wykład informacyjny (prezentacje multimedialne, materiały na platformie Moodle).
Projekt	Projekt, dyskusja.

5. **Metody i kryteria oceniania**

Forma zajęć:	**Forma zaliczenia:**

Uzyskane punkty są przeliczane na oceny według następującej skali:

Procent punktów	Ocena
91-100%	Bardzo dobry
85-90%	Dobry plus
76-84%	Dobry
66-75%	Dostateczny plus
51-65%	Dostateczny
0-50%	Niedostateczny

Opis:

Forma zaliczenia: wykłady
Ocena na podstawie prac pisemnych (kolokwia zaliczeniowe).

Forma zaliczenia: projekty
Zaliczenie końcowe na podstawie wykonania poszczególnych etapów projektu i zadań. Wykonanie poszczególnych zadań jest oceniane na bieżąco po zakończeniu ćwiczenia.

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen z każdej z form zajęć

	Zatwierdzenie karty opisu zajęć	
	Stanowisko Tytuł/stopień naukowy, imię nazwisko	**Podpis**
Opracował	dr inż. Piotr Sujka	
Zatwierdził	Dyrektor Instytutu Nauk Technicznych dr inż. Łukasz Józefowski	