

# KARTA PRZEDMIOTU

## 1. Informacje ogólne

<b>Nazwa przedmiotu i kod (wg planu studiów):</b>	Projekt zespołowy D1_13
<b>Nazwa przedmiotu (j. ang.):</b>	Team Project
<b>Kierunek studiów:</b>	Informatyka
<b>Specjalność/specjalizacja:</b>	Sieciowe Systemy Informacyjne
<b>Poziom kształcenia:</b>	studia I stopnia
<b>Profil kształcenia:</b>	praktyczny (P)
<b>Forma studiów:</b>	studia stacjonarne
<b>Obszar kształcenia:</b>	nauki techniczne
<b>Dziedzina:</b>	nauki techniczne
<b>Dyscyplina nauki:</b>	informatyka
<b>Koordynator przedmiotu:</b>	dr hab. Jan Bazan

## 2. Ogólna charakterystyka przedmiotu

<b>Przynależność do modułu:</b>	kształcenia kierunkowego
<b>Status przedmiotu:</b>	obowiązkowy
<b>Język wykładowy:</b>	polski
<b>Rok studiów, semestr:</b>	III, 5 i 6
<b>Forma i wymiar zajęć według planu studiów:</b>	stacjonarne - ćwiczenia projektowe 75 h niestacjonarne - ćwiczenia projektowe 30 h
<b>Interesariusze i instytucje partnerskie (nieobowiązkowe)</b>	
<b>Wymagania wstępne / Przedmioty wprowadzające:</b>	Algorytmy i struktury danych, Programowanie I, Programowanie II

### 3. Bilans punktów ECTS

<b>Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):</b>	8 (A+B)	stacjonarne	stacjonarne
<b>A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach</b>	ćwiczenia seminaryjne	75	30
	konsultacje	4	10
	<b>W sumie:</b>	79	40
	ECTS	3	1,6
<b>B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS</b>	praca nad projektami	100	120
	studiowanie materiałów w sieci	20	20
	przygotowanie do konsultacji	5	5
	studiowanie zalecanej literatury	5	11
	<b>w sumie:</b>	130	166
ECTS	5	6,4	
<b>C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS</b>	praca nad projektami	150	150
	ECTS	6	6

### 4. Opis przedmiotu

<p><b>Cel przedmiotu:</b></p> <p>Celem tego przedmiotu jest zespołowe zrealizowanie przez studentów dużego projektu programistycznego. Powinno być to podsumowanie i praktyczne zweryfikowanie wiedzy programistycznej nabytej przez studentów na I etapie studiów. Studenci powinni w trakcie tych zajęć przejść przez pełen cykl produkcji oprogramowania: od specyfikacji wymagań po testowanie. Faza pozyskiwania specyfikacji powinna być jak najbardziej zbliżona do rzeczywistości (tzn. studenci nie powinni dostawać gotowej specyfikacji wymagań, lecz stworzyć ją na podstawie rozmów z prowadzącym zajęcia lub innymi osobami – pełniącym wówczas rolę klienta). W trakcie zajęć zwracana jest uwaga na umiejętności miękkie niezbędne do podjęcia efektywnej pracy zespołowej.</p>
<p><b>Metody dydaktyczne:</b> wykład, dyskusja, prezentacje studentów (zajęcia konwersatoryjne)</p>
<p><b>Treści kształcenia:</b></p>

### Ćwiczenia laboratoryjno-seminaryjne:

1. Wprowadzenie do projektowania zespołowego (duża waga i trudności programowania zespołowego, duży wysiłek inżynierii oprogramowania celem opracowania efektywnych metod, cykl produkcji oprogramowania).
2. Przedstawienie metodologii według której będą realizowane projekty oraz zasad dokumentowania prac projektowych.
3. Omówienie umiejętności miękkich związanych z pracą zespołową
4. Przedstawienie aplikacji komputerowych ułatwiających pracę zespołową w większej grupie.
5. Przedstawianie bieżących wyników działalności zespołów projektowych w postaci prezentacji komputerowych i ich omawianie
6. Końcowa prezentacja projektu oraz oddanie dokumentacji i arkusza oceny projektu.

### 5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia		
Efekt przedmiotu (kod przedmiotu + kod efektu kształcenia)	Student, który zaliczył przedmiot (spełnił minimum wymagań)	Efekt kierunkowy
D1.13_W01	<b>Wiedza:</b>  Student ma wiedzę na temat etapów zespołowego tworzenia oprogramowania oraz wie jak dzielić prace w zespole stosując indywidualne predyspozycje członków zespołu.	K_W14  K_W16
D1.13_U01	<b>Umiejętności</b>  Student potrafi pracować w zespole nad wspólnym projektem, przyjmując ustaloną rolę.	K_U04 K_U05 K_U06 K_U07 K_U09 K_U10  K_U16 K_U17 K_U18

		K_U22 K_U27 K_U30 K_U31
D1.13_K01	<b>Kompetencje społeczne</b>  Potrafi pracować w zespole.	K_K01 K_K04 K_K05 K_K06 K_K07 K_K08 K_K09

**Sposoby weryfikacji efektów kształcenia:**

Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca – przykładowe sposoby jej wystawienia poniżej	Ocena końcowa  przykładowe sposoby jej wystawienia poniżej
1	D1.13_W01, D1.13_U01	Zaliczenie projektu (prezentacja, inspekcja kodu przez prowadzącego, analiza zaawansowania projektu poprzez arkusz projektowy wypełniony przez studentów)	Ocena z zaliczenia projektu	Ocena z zaliczenia projektu, ewentualnie zmodyfikowana o 1/2 stopnia w przypadku dużego zaawansowania studenta w dyskusje podczas zajęć

**Kryteria oceny** (oceny 3,0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):

<b>w zakresie wiedzy</b>		<b>Efekt kształcenia</b>
Na ocenę 3,0	<p>Student ma wiedzę na temat etapów zespołowego tworzenia oprogramowania oraz wie jak dzielić prace w zespole stosując indywidualne predyspozycje członków zespołu. Jednak Jego wiedza</p> <p>nie jest na tyle obszerna, aby zapewnić pełną realizację specyfikacji prac projektowych sformułowanych przez prowadzącego według współczesnych standardów znanych z literatury.</p>	D1.13_W14 D1.13_W16
Na ocenę 5,0	<p>Student ma wiedzę na temat etapów zespołowego tworzenia oprogramowania oraz wie jak dzielić prace w zespole stosując indywidualne predyspozycje członków zespołu. Ponadto, jego wiedza jest na tyle obszerna, aby zapewnić pełną realizację specyfikacji prac projektowych sformułowanych przez prowadzącego według współczesnych standardów znanych z literatury.</p>	D1.13_W14 D1.13_W16
<b>w zakresie umiejętności</b>		
Na ocenę 3,0	<p>Student potrafi pracować w zespole nad wspólnym projektem, przyjmując ustaloną rolę. Jednak nie potrafi w pełni zrealizować wszystkich zadań wyspecyfikowanych przez prowadzącego w ramach prac projektowych według współczesnych standardów znanych z literatury.</p>	D1.13_U04 D1.13_U05 D1.13_U06 D1.13_U07 D1.13_U09 D1.13_U10 D1.13_U16 D1.13_U17 D1.13_U18 D1.13_U22 D1.13_U27 D1.13_U30 D1.13_U31
Na ocenę 5,0	<p>Student potrafi pracować w zespole nad wspólnym projektem, przyjmując ustaloną rolę. Ponadto, potrafi w pełni zrealizować wszystkie zadania wyspecyfikowane przez prowadzącego w ramach prac projektowych według współczesnych standardów</p>	D1.13_U04 D1.13_U05

	znanych z literatury.	D1.13_U06 D1.13_U07 D1.13_U09 D1.13_U10 D1.13_U16 D1.13_U17 D1.13_U18 D1.13_U22 D1.13_U27 D1.13_U30 D1.13_U31
<b>w zakresie kompetencji społecznych</b>		
Na ocenę 3,0	Student potrafi pracować w zespole.	D1.13_K01 D1.13_K04 D1.13_K05 D1.13_K06 D1.13_K07 D1.13_K08 D1.13_K09
Na ocenę 5,0	Student potrafi być liderem lub bierze aktywny udział w dyskusjach w trakcie pracy w zespole.	D1.13_K01 D1.13_K04 D1.13_K05 D1.13_K06 D1.13_K07 D1.13_K08 D1.13_K09
<p><b>Kryteria oceny końcowej</b></p> <p><u>Zaliczenie przedmiotu</u> następuje na podstawie zaliczenia projektu w oparciu arkusz projektowy, który zawiera kilkadziesiąt kryteriów, które w arkuszu są agregowane w jedną ocenę.</p> <p>Student otrzymuje ocenę <b>niedostateczny</b>, gdy ocena za projekt jest niższa od 3.0.</p>		

Student otrzymuje ocenę **dostateczny**, gdy ocena za projekt jest co najmniej 3.0, ale nie przekracza 3.75;  
Student otrzymuje ocenę **dobry**, gdy ocena za projekt jest co najmniej 3.75, ale nie przekracza 4.75;  
Student otrzymuje ocenę **bardzo dobry**, gdy ocena za projekt jest co najmniej 4.75.

**Zalecana literatura** (w podziale na literaturę podstawową i uzupełniającą):

**Podstawowa:**

1. Wykłady: <http://fenix.univ.rzeszow.pl/bazan/>
2. Viktor Farcic, Alex Garcia: TDD. Programowanie w Javie sterowane testami, Helion, (2016) (dostępna w bibliotece PWSZ).
3. Schildt, H.: Java. Przewodnik dla początkujących, wydanie VI, Gliwice: Helion (2015) (dostępna w bibliotece PWSZ).
4. Horstman, C., S., Cornell, G.: Java, Podstawy, wydanie IX, Gliwice: Helion (2013) (dostępna w bibliotece PWSZ).

**Uzupełniająca:**

1. Jon Loeliger, Matthew McCullough: Kontrola wersji z systemem Git. Narzędzia i techniki programistów, Helion (2014) (dostępna w bibliotece PWSZ).
2. Schildt, H.: Java. Kompendium programisty, wydanie IX, Gliwice: Helion (2015) (dostępna w bibliotece PWSZ).
3. Horstman, C., S., Cornell, G.: Java, Techniki zaawansowane, wydanie IX, Gliwice: Helion (2013) (dostępna w bibliotece PWSZ).
4. Dokumentacje narzędzi programowania zespołowego dostępne w sieci Internet.

**Informacje dodatkowe:**

**Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:**

Konsultacje – 4 godziny

Weryfikacja dokumentacji projektowej, w tym arkuszy projektowych - 6 godzin .

W sumie: 10 godzin