

<b>Nazwa przedmiotu</b> <i>Projekt zespołowy</i> <i>Team project</i>		<b>Kod ECTS</b>		
<b>Nazwa jednostki prowadzącej przedmiot</b> <i>Uniwersytet Opolski, Wydział Matematyki, Fizyki i Informatyki, Instytut Matematyki i Informatyki</i>				
<b>Studia</b>				
	<b>Kierunek</b>	<b>stopień</b>	<b>tryb</b>	<b>specjalność</b>
	<i>Informatyka</i>	<i>Pierwszy</i>	<i>Stacjonarne</i> <i>Niestacjonarne *)</i>	
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b> <i>Krótkiewicz, Wojtkiewicz</i>				
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS: 3</b>		
<b>A. Formy zajęć</b> • <i>laboratorium (L)</i>		<i>Bilans nakładu pracy przeciętnego studenta:</i> • 75 godz. – udział w laboratoriach [ <sup>*)</sup> 40] • 5 godz. – udział w konsultacjach [ <sup>*)</sup> 2] • 10 godz. – przygotowanie dokumentacji i aplikacji zaliczeniowej na laboratorium. [ <sup>*)</sup> 48]  <b><i>Łączny nakład pracy studenta: 90 godzin, co odpowiada 3 pkt ECTS</i></b>		
<b>B. Sposób realizacji</b> • <i>zajęcia w sali laboratoryjnej</i>		<i>w tym</i> • <i>nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 75+5=80 godz., co odpowiada 3 pkt ECTS;</i> • <i>nakład pracy związany z zajęciami o charakterze praktycznym: 75+10= 85 godz., co odpowiada 3 pkt ECTS</i>		
<b>C. Liczba godzin</b>  <i>Laboratorium – 75 godzin</i>  <i>*) Studia niestacjonarne: Laboratorium –40 godzin</i>		<i>*) na studiach niestacjonarnych:</i> • <i>nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 40+2=42 godz., co odpowiada 2 pkt. ECTS;</i> • <i>nakład pracy związany z zajęciami o charakterze praktycznym: 40+48 = 88 godz., co odpowiada 3 pkt. ECTS</i>		
<b>Status przedmiotu</b> • <i>obowiązkowy (kanon)</i>		<b>Język wykładowy</b> <i>Polski (możliwość realizacji w języku angielskim)</i>		
<b>Metody dydaktyczne</b> • <i>ćwiczenia laboratoryjne: opracowanie dokumentacji projektowej wg zadanego szablonu, wykonanie aplikacji komputerowej</i>		<b>Forma i sposób zaliczenia oraz podst. kryteria oceny lub wymagania egzaminacyjne</b> <i>Na ogólnych zasadach określonych w programie kształcenia, a w szczególności</i>		
		<b>A. Sposób zaliczenia</b> • <i>zaliczenie (0-20 pkt.) (laboratorium), waga: 100% oceny końcowej</i>		
		<b>B. Formy zaliczenia</b> • <i>(L) zaliczenie (0-20 pkt.): ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru za dokumentację projektową, aplikację komputerową napisaną w ramach realizowanego projektu oraz kolokwium</i>		
		<b>C. Podstawowe kryteria</b> • <i>(L) uzyskanie pozytywnej oceny;</i>		
<b>Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi</b> <i>Należy określić:</i> <b>A. <u>Wymagania formalne:</u></b> <i>pozytywna ocena z przedmiotu: Inżynieria oprogramowania</i> <b>B. <u>Wymagania wstępne:</u></b> <i>umiejętność programowania obiektowego, umiejętność projektowania i implementacji relacyjnych baz danych, znajomość UML, umiejętność wykorzystania pakietu biurowego MS Office oraz umiejętność korzystania z oprogramowania do modelowania np. Sybase Power Designer.</i>				
<b>Cele przedmiotu</b> <i>Przekazanie wiedzy i umiejętności z zakresu realizacji zespołowych projektów informatycznych, ze szczególnym uwzględnieniem kooperacji w obrębie zespołu, zarządzania projektem informatycznym.</i>				
<b>Treści programowe</b> <b>A. Problematyka laboratorium:</b> <i>Zbudowanie zespołu projektowego, przydział zadań, harmonogramowanie, kontrola przebiegu procesu produkcji oprogramowania, ocena realizacji zadań na poszczególnych etapach przez każdego z członków zespołu. Ponadto klasyczne zagadnienia związane z wytwarzaniem oprogramowania, tj.:</i> • <i>Wykonanie dokumentacji projektowej (zgodnie z szablonem) zawierającej m.in.: studium wykonalności, identyfikację punktów widzenia, hierarchiczną strukturę wymagań, diagram przypadków użycia, rozkład funkcjonalny, model syste-</i>				

mu, architekturę wdrożeniową systemu, wykres Gantt, dokumentację użytkownika.

- Wykonanie autorskiej aplikacji komputerowej zgodnej z przygotowaną dokumentacją, przy zastosowaniu wybranych technologii. Aplikacja komputerowa musi zawierać co najmniej jedną funkcjonalność zrealizowaną w formie modułu napisanego w języku programowania.

## Wykaz literatury

### A. Literatura wymagana do ostatecznego zaliczenia zajęć (zdania egzaminu):

#### A.1. wykorzystywana podczas zajęć

- Wrycza, Stanisław : Język UML 2.0 w modelowaniu systemów informatycznych : diagramy języka UML, modelowanie biznesowe, metodyki projektowe oparte na UML, narzędzia CASE / Gliwice : Wydawnictwo Helion, cop. 2005, ISBN 83-7361-892-9
- Malina, Witold. : Metodologia i techniki programowania / Warszawa : Wydawnictwo Naukowe PWN : Mikom, cop. 2008, ISBN 978-83-01-15469-1 (PWN).

#### A.2. studiowana samodzielnie przez studenta

- Cadle, James. : Zarządzanie procesem tworzenia systemów informacyjnych / Warszawa : Wydawnictwa Naukowo-Techniczne, cop. 2004, Inżynieria Oprogramowania, ISBN 83-204-2928-5
- Cantor, Murray. : Jak kierować zespołem programistów / Warszawa : Wydawnictwa Naukowo-Techniczne, 2004, Inżynieria Oprogramowania, ISBN 83-204-2880-7

### B. Literatura uzupełniająca

- Inżynieria systemów informatycznych, Beynon-Davies Paul, WNT 2004
- Metryki i modele w inżynierii jakości oprogramowania, Stephen H.Kan, PWN 2006
- inne podręczniki dostępne on-line poprzez Bibliotekę Główną UO („ibuk”)

Efekty kształcenia	<b>Wiedza</b>			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	W01	Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie inżynierii oprogramowania	praca kontrolna/ projekt	K_W03
	W02	Zna podstawowe techniki programistyczne z zakresu programowania obiektowego.		K_W04
	W03	Ma wiedzę na temat projektowania oprogramowania; specyfikacji i analizy wymagań; walidacji i testowania oprogramowania; zarządzania przedsięwzięciem programistycznym; zna narzędzia i środowiska wytwarzania oprogramowania; procesy wytwarzania oprogramowania		K_W12, K_W13
	W04	Ma wiedzę na temat ergonomii w aspekcie UI.		K_W10
	<b>Umiejętności:</b>			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	U01	Potrafi w sposób czytelny przygotowywać dokumenty projektowe.	praca kontrolna/ projekt	K_U04
	U02	Potrafi przygotować dokumentację projektową, a w tym dokumentację użytkownika w sposób odpowiadający poziomowi technicznego zaawansowania odbiorcy.		K_U03, K_U05
	U03	Potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym		K_U09
	U04	Ma umiejętność budowy prostych systemów informatycznych oraz aplikacji bazodanowych wykorzystujących przynajmniej jeden z najbardziej popularnych systemów zarządzania bazą danych		K_U28, K_U29, K_U15
	<b>Kompetencje społeczne (postawy)</b>			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	K01	Rozumie potrzebę pracy zespołowej	Konwersacja / obserwacja	K_K02
K02	Potrafi zaplanować czynności związane z pracą przy projekcie zespołowym	K_K03		

### Kontakt:

Wykaz numerów telefonicznych i adresów mailowych pracowników znajduje się na stronie Instytutu Matematyki i Informatyki:  
www.math.uni.opole.pl