

# PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA W NOWYM SĄCZU

## KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2016/2017

Instytut Techniczny

Kierunek studiów: Informatyka

Profil: Praktyczny

Forma studiów: Stacjonarne

Kod kierunku: 11.3

Stopień studiów: I

Specjalności: Informatyka stosowana

### 1 PRZEDMIOT

NAZWA PRZEDMIOTU	Inżynieria oprogramowania A
KOD PRZEDMIOTU	IT 11.3 PIS B17 16/17
KATEGORIA PRZEDMIOTU	Przedmioty podstawowe i kierunkowe
LICZBA PUNKTÓW ECTS	8
SEMESTRY	5 6

### 2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	PROJEKT	SEMINARIUM
5	15	15	15		
6	15			30	

### 3 CELE PRZEDMIOTU

**Cel 1** Przedstawienie najważniejszych zagadnień związanych z rozwojem systemów informatycznych.

**Cel 2** Zaznajomienie z podstawowymi metodami projektowania, realizacji i utrzymywania systemów informatycznych.

**Cel 3** Nabycie umiejętności praktycznego wykorzystania uzyskanej wiedzy.



## 4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

- a Algorytmy i struktury danych.
- b Matematyczne podstawy w informatyce.
- c Wprowadzenie do algorytmów i programowania.

## 5 EFEKTY KSZTAŁCENIA

**EK1** Wiedza: Posiada wiedzę dotyczącą najważniejszych pojęć i technik inżynierii oprogramowania, obejmującą zagadnienia projektowania, implementacji i utrzymywania systemów informatycznych.

**EK2** Umiejętności: Umie zastosować uzyskaną wiedzę przy realizacji zadań praktycznych (np. projektowanie i implementacja prostego systemu informatycznego), w tym wykorzystać stosowne narzędzia komputerowe, a także tworzyć własne programy.

**EK3** Kompetencje społeczne: Potrafi pracować w grupie przy realizacji projektu zespołowego.

## 6 TREŚCI PROGRAMOWE

### WYKŁAD

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Wprowadzenie do inżynierii oprogramowania, podstawowe pojęcia i wyzwania.	2
W2	Zasady skutecznego wytwarzania oprogramowania.	4
W3	Specyfikacja wymagań.	2
W4	Kontrola jakości artefaktów.	2
W5	Wstęp do projektowania w języku UML.	4
W6	Metody formalne.	4
W7	Wzorce projektowe.	2
W8	Zarządzanie konfiguracją.	2
W9	Wstęp do testowania oprogramowania.	4
W10	Programowanie ekstremalne.	2
W11	Ewolucja i refaktoryzacja oprogramowania.	2
	RAZEM	<b>30</b>

### LABORATORIUM

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Manager projektów i zarządzanie projektami.	2
L2	Posługiwanie się pomocą Projekt Wizard, szablony i wzorce dokumentów.	3
L3	Praca w środowisku zintegrowanym IDE (Integrated Development Environment).	3
L4	Praca z debuggerem.	4
L5	Tworzenie aplikacji na współpracy software i hardware.	3
	RAZEM	<b>15</b>

### PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Rational Unified Process: dokładny opis metodyki i procesu produkcji oprogramowania.	2



## PROJEKT

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P2	Capability Maturity Model: dokładny opis metodyki i procesu produkcji oprogramowania.	2
P3	Narzędzia zarządzania wymaganiami.	2
P4	Narzędzia wspomagające zarządzanie zmianami (wymagań, kodu) w procesie produkcji oprogramowania.	2
P5	Metryki dotyczące jakości oprogramowania.	2
P6	Narzędzia generujące diagramy w UML.	2
P7	Zintegrowane środowiska programistyczne elementy systemów CASE.	2
P8	Technika zarządzania ryzykiem w projektach informatycznych.	2
P9	Metody i techniki testowania oprogramowania.	2
P10	Notacje graficzne w modelowaniu i analizie systemów baz danych.	2
P11	Metody estymacji wielkości kodu: metoda COSMIC function point.	2
P12	Narzędzia przeprowadzające statyczną analizę kodu źródłowego.	2
P13	Notacje graficzne w modelowaniu obiektowym.	2
P14	Narzędzia szybkiego tworzenia interfejsu użytkownika (RAD).	2
P15	Analiza i projekt aplikacji typu Klient Serwer z gwarancją jakości usług QoS.	2
	RAZEM	30

## ĆWICZENIA

LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
C1	Dowodzenie poprawności prostych algorytmów.	3
C2	Planowanie projektu a zasady skutecznego działania, formułowanie celów i zadań metodą SMART, praktyki zarządzania czasem, technika wizualizacji i afirmacji.	4
C3	Specyfikacja wymagań.	4
C4	Metody formalne, sieci Petriego.	4
	RAZEM	15

## 7 METODY DYDAKTYCZNE

M1 Wykłady

M2 Ćwiczenia laboratoryjne

M3 Ćwiczenia projektowe

M4 Prezentacje multimedialne

M5 Konsultacje

M6 Słowne objaśnienie

M7 Projekty



## 8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
<b>Godziny kontaktowe z nauczycielem akademickim, w tym:</b>	
Godziny wynikające z planu studiów	90
Konsultacje przedmiotowe	7
Egzaminy i zaliczenia w sesji	4
<b>Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:</b>	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	30
Opracowanie wyników	30
Przygotowanie raportu, projektu, prezentacji, dyskusji	39
<b>SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA</b>	<b>200</b>
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	8

## 9 SPOSOBY OCENY

### OCENA FORMUJĄCA

F1 Egzamin

F2 Odpowiedź ustna

F3 Projekt zespołowy

F4 Aktywność na zajęciach

F5 Projekt indywidualny

F6 Sprawozdanie z ćwiczenia laboratoryjnego

### OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA AKADEMICKIEGO

1 Projekt zespołowy

### KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 1
NA OCENĘ 3	Zna niektóre z wymaganych pojęć i technik inżynierii oprogramowania.	wykład	Ocena z egzaminu.
NA OCENĘ 4	Zna większość z wymaganych pojęć i technik inżynierii oprogramowania, jednakże wiedza ta nie jest wystarczająco ugruntowana.		
NA OCENĘ 5	Zna i rozumie wymagane pojęcia i techniki inżynierii oprogramowania.		
EFEKT KSZTAŁCENIA 2		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 2



NA OCENĘ 3	Potrafi realizować niektóre z wymaganych zadań praktycznych, a także przy pomocy nauczyciela akademickiego posługiwać się odpowiednimi narzędziami komputerowymi.	ćwiczenia, laboratorium, projekt	Po semestrze 5: średnia arytmetyczna ocen z ćwiczeń tradycyjnych i laboratoryjnych. Po semestrze 6: ocena z ćwiczeń projektowych.
NA OCENĘ 4	Potrafi realizować większość z wymaganych zadań praktycznych, a także samodzielnie posługiwać się odpowiednimi narzędziami komputerowymi.		
NA OCENĘ 5	Potrafi realizować wymagane zadania praktyczne, a także samodzielnie posługiwać się odpowiednimi narzędziami komputerowymi oraz tworzyć własne programy.		
EFEKT KSZTAŁCENIA 3		MIEJSCE WERYFIKACJI	OPIS WERYFIKACJI EK 3
NA OCENĘ 3	Potrafi wykonywać większość zadań przydzielonych przez zespół.	projekt	Ocena aktywności przy realizacji projektu zespołowego.
NA OCENĘ 4	Potrafi wykonywać wszystkie zadania przydzielone przez zespół.		
NA OCENĘ 5	Potrafi wykonywać wszystkie zadania przydzielone przez zespół oraz kierować jego pracą.		

**OCENA DO INDEKSU (OCENA PODSUMOWUJĄCA)**

Po semestrze 5: ocena efektu kształcenia EK2 (średnia arytmetyczna ocen z ćwiczeń tradycyjnych i laboratoryjnych).  
 Po semestrze 6: średnia ważona ocen poszczególnych efektów kształcenia, tj. EK1 (ocena z egzaminu, waga 40%), EK2 (ocena z ćwiczeń projektowych, waga 40%), EK3 (ocena aktywności przy realizacji projektu zespołowego, 20%).

**WARUNKI ZALICZENIA PRZEDMIOTU**

a Uzyskanie pozytywnych ocen ze wszystkich efektów kształcenia.

**10 MACIERZ REALIZACJI PRZEDMIOTU**

EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK1	INFP_W13, INFP_W10, INFP_W14	Cel1, Cel2	W1, W2, W3, W4, W5, W6, W7, W8, W9, W10, W11	M1, M4, M5, M6
EK2	INFP_UP01, INFP_UP03, INFP_UB04, INFP_UB02, INFP_UP02	Cel2, Cel3	L1, L2, L3, L4, L5, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, C1, C2, C3, C4	M2, M3, M5, M6, M7



EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU	ODNIESIENIE DO EFEKTÓW KIERUNKOWYCH	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	METODY DYDAKTYCZNE
EK3	INFP_UO02	Cel3	P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15	M7

## 11 WYKAZ LITERATURY

### LITERATURA PODSTAWOWA:

- [1] J. Górski — *Inżynieria oprogramowania w projekcie programistycznym*, Warszawa, 2000, MIKOM
- [2] A. Jaskiewicz — *Inżynieria oprogramowania*, Gliwice, 1997, HELION
- [3] S. H Kan — *Metryki i modele w inżynierii jakości oprogramowania*, Warszawa, 2006, PWN
- [4] I. Sommerville — *Inżynieria oprogramowania*, Warszawa, 2003, WNT
- [5] E. Yourdon, C. Argila — *Analiza obiektowa i projektowanie. Przykłady zastosowań*, Warszawa, 2000, WNT
- [6] B. Wiszniewski, B. Bereza-Jarociński — *Teoria i praktyka testowania programów*, Warszawa, 2002, PWN
- [7] M. Fowler, K. Scot — *UML w kropelce, wersja 2.0*, Warszawa, 2005, LTP
- [8] P. Beynon-Davies — *Inżynieria systemów informacyjnych*, Warszawa, 2004, WNT

### LITERATURA UZUPEŁNIAJĄCA:

- [1] D. Hamlet, J. Maybee — *Podstawy techniczne inżynierii oprogramowania*, Warszawa, 2003, WNT
- [2] S. Roger — *Praktyczne podejście do inżynierii oprogramowania*, Warszawa, 2004, WNT

## 12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

### OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. Marcin Mazur, prof. PWSZ (kontakt: mazur@im.uj.edu.pl)

### OSOBY PROWADZĄCE PRZEDMIOT

dr inż. Ihor Pazdriy (kontakt: irpazdriy@gmail.com)

dr hab. Marcin Mazur, prof. PWSZ (kontakt: mazur@im.uj.edu.pl)

## 13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)                      (odpowiedzialny za przedmiot)                      (kierownik zakładu)                      (dyrektor instytutu)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....  
.....