

**SYLABUS**

DOTYCZY CYKLU KSZTAŁCENIA 2014-2018  
realizacja w roku akademickim 2016/17

**1.1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE/MODULE**

Nazwa przedmiotu/ modułu	<b>Inżynieria oprogramowania</b>
Kod przedmiotu/ modułu*	
Wydział (nazwa jednostki prowadzącej kierunek)	<b>Wydział Matematyczno - Przyrodniczy</b>
Nazwa jednostki realizującej przedmiot	<b>Wydział Matematyczno - Przyrodniczy</b>
Kierunek studiów	<b>Informatyka</b>
Poziom kształcenia	<b>studia inżynierskie I-go stopnia</b>
Profil	<b>ogólnoakademicki</b>
Forma studiów	<b>stacjonarne</b>
Rok i semestr studiów	<b>rok III, semestr 5</b>
Rodzaj przedmiotu	<b>przedmiot kształcenia kierunkowego</b>
Koordinator	<b>dr inż. Michał Kępski</b>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<b>dr inż. Michał Kępski</b>

\* - zgodnie z ustaleniami na wydziale

**1.2. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS**

Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne ( jakie?)	Liczba pkt ECTS
<b>30</b>			<b>30</b>					<b>5</b>

**1.3. Sposób realizacji zajęć** zajęcia w formie tradycyjnej zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość**1.4. Forma zaliczenia przedmiotu/ modułu ( z toku) ( egzamin, zaliczenie z oceną, zaliczenie bez oceny)**  
**ZALICZENIE Z OCENĄ****2. WYMAGANIA WSTĘPNE**

- Znajomość zagadnień z przedmiotów matematycznych: logika (rachunek zdań i tautologie), algebra zbiorów i relacje, funkcje
- Znajomość zagadnień z przedmiotów Algorytmy i struktury danych oraz Podstawy programowania (w tym zagadnienie rekurencji),
- Znajomość zagadnień z przedmiotów Podstawy programowania i Programowanie obiektowe: umiejętność programowania w języku obiektowym

**3. CELE, EFEKTY KSZTAŁCENIA , TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE****3.1. Cele przedmiotu/modułu**

C1	Zapoznanie studentów z podstawowymi zagadnieniami, narzędziami i technikami z zakresu inżynierii oprogramowania
C2	Nabywanie przez studentów praktycznych umiejętności z zakresu stosowania inżynierii oprogramowania

### 3.2 EFEKTY KSZTAŁCENIA DLA PRZEDMIOTU/ MODUŁU ( WYPEŁNIA KOORDYNATOR)

EK ( efekt kształcenia)	Treść efektu kształcenia zdefiniowanego dla przedmiotu (modułu)	Odniesienie do efektów kierunkowych (KEK)
EK_01	Student zna obiektowe i strukturalne metodyki modelowania systemów informatycznych. Zna poziomy testowania, typy testów, ich role oraz metody testowania. Zna metody procesu wytwórczego oprogramowania (tradycyjne oraz zwinne, np. <i>Scrum</i> ). Posiada wiedzę z zakresu wzorców projektowych.	K_W24
EK_02	Student zna narzędzia do obiektowego modelowania oprogramowania (w języku UML). Zna systemy kontroli wersji i automatycznego przeprowadzania testów oprogramowania.	K_W25
EK_03	Student zna dobre praktyki tworzenia dokumentacji. Jest świadom istnienia standardów tworzenia dokumentacji (ISO, IEEE) oraz korzyści płynących z ich wykorzystania. Zna metody pozyskiwania wymagań w procesie tworzenia specyfikacji wymagań systemu informatycznego.	K_W38
EK_04	Potrafi tworzyć diagramy UML: przypadków użycia, klas, sekwencji, aktywności, maszyny stanowej. Zna narzędzia do tworzenia i przeprowadzania testów oprogramowania (JUnit).	K_U22
EK_05	Potrafi wykorzystać system kontroli wersji Git.	K_U23
EK_06	Potrafi wykorzystać narzędzia automatycznego tworzenia i rozwijania dokumentacji na podstawie kodu źródłowego. Zna znaczniki <i>JavaDoc</i> i ich rolę w procesie tworzenia dokumentacji.	K_U35 K_K09
EK_07	Potrafi wykorzystać narzędzia planowania projektów (metoda PERT), tworzyć diagramy Ganta. Zna rolę określania priorytetów w procesie tworzenia oprogramowania opartym o metodyki <i>Agile</i> .	K_K07

### 3.3 TREŚCI PROGRAMOWE (wypełnia koordynator)

#### A. Problematyka wykładu

Wprowadzenie do inżynierii oprogramowania (podstawowe pojęcia i problematyka, przegląd zagadnień).
Fazy (etapy) procesu produkcji oprogramowania.
Klasyczne metodologie wytwarzania oprogramowania. Zwinne metodologie wytwarzania oprogramowania, w tym programowanie ekstremalne i <i>Scrum</i> .
Wstęp do modelowania w języku UML.
Podstawowe wzorce projektowe.
Zagadnienie definiowania celów i zakresu systemu informatycznego.
Podstawowe zagadnienia związane z odkrywaniem wymagań do projektu informatycznego.
Zagadnienia związane z modelowaniem systemu, w tym podstawowe diagramy dokumentujące wyniki modelowania.

Zagadnienia związane z projektowaniem systemu, w tym podstawowe diagramy dokumentujące wyniki projektowania.
Zagadnienia związane z implementacją systemu (system operacyjny, język programowania, GUI, biblioteki, komponenty).
Dokumentacja (standardy tworzenia dokumentacji).
Testowanie systemu i typowe metody testowania (testy biało-skrzynkowe i czarno-skrzynkowe, testy jednostkowe i integracyjne oraz efektywnościowe).
Wdrożenie (typowe działania wdrożeniowe). Konserwacja oprogramowania (motywacja, typowe działania konserwacyjne).

**B. Problematyka ćwiczeń audytoryjnych, konwersatoryjnych, laboratoryjnych, zajęć praktycznych**

Język UML: diagramy przypadków użycia. Diagramy klas.
Język UML: diagramy sekwencji. Diagramy aktywności. Diagramy maszyny stanowej.
Wzorce projektowe w języku JAVA. Implementacja – odwzorowanie modeli na kod.
Refaktoryzacja do wzorców projektowych.
Narzędzia planowania projektów informatycznych. Metoda PERT. Diagramy Ganta.
Podstawy obsługi systemów kontroli wersji.
Tworzenie dokumentacji kodu źródłowego. JavaDoc.
Wprowadzenie do testowania.

### 3.4 METODY DYDAKTYCZNE

*Wykład: wykład z prezentacją multimedialną*

*Laboratorium: dyskusja, rozwiązywanie zadań z użyciem narzędzi informatycznych*

## 4 METODY I KRYTERIA OCENY

### 4.1 Sposoby weryfikacji efektów kształcenia

Symbol efektu	Metody oceny efektów kształcenia ( np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych ( w, ćw, ...)
EK_01	zaliczenie wykładu w formie testu	wykład
EK_02	zaliczenie wykładu w formie testu	wykład
EK_03	zaliczenie wykładu w formie testu	wykład
EK_04	obserwacja w trakcie zajęć, kolokwium	laboratorium
EK_05	obserwacja w trakcie zajęć, kolokwium	laboratorium
EK_06	obserwacja w trakcie zajęć, kolokwium	laboratorium
EK_07	obserwacja w trakcie zajęć, kolokwium	laboratorium

#### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

*Wykład : zaliczenie bez oceny (test)*

*Zajęcia laboratoryjne : zaliczenie z oceną - na podstawie ocen z kolokwium i aktywności podczas zajęć*

*Zaliczenie zajęć : ocena z laboratoriów z uwzględnieniem konieczności zaliczenia wykładów*

#### 5. Całkowity nakład pracy studenta potrzebny do osiągnięcia założonych efektów w godzinach oraz punktach ECTS

Aktywność	Liczba godzin/ nakład pracy studenta
godziny zajęć wg planu z nauczycielem	60
przygotowanie do zajęć	60
udział w konsultacjach	3
przygotowanie do zaliczenia wykładu	15
SUMA GODZIN	138
<b>SUMARYCZNA LICZBA PUNKTÓW ECTS</b>	<b>5</b>

#### 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU/ MODUŁU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

#### 7. LITERATURA

Literatura podstawowa:

1. K. Beck, C. Anders, *Wydajne programowanie – ExtremeProgramming*, Mikom, 2006
2. A. CockBurn, *Jak pisać efektywne przypadki użycia*, WNT, Warszawa, 2004
3. S. Covey, *7 nawyków skutecznego działania*, REBIS, 2012
4. M. Fowler, K. Scott, *UML w kropelce*, LTP, 2002
5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Wzorce projektowe : elementy oprogramowania obiektowego wielokrotnego użytku*, Gliwice, Wydawnictwo Helion, 2017.

Literatura uzupełniająca:

1. K. Sacha: *Inżynieria oprogramowania*, Warszawa, PWN, 2010
2. E. Yourdon, C. Argila: *Analiza obiektowa i projektowanie*, WNT, 2000
3. A. Roman, *Testowanie i jakość oprogramowania : modele, techniki, narzędzia*, Warszawa, PWN, 2015

Akceptacja Kierownika Jednostki lub osoby upoważnionej