

Zespół Autorski:

*mgr inż. Marcin Bączyk
dr hab. inż. Michał Borecki,
dr inż. Dominik Kasprowicz,
dr hab. Marek Nałęcz, prof. PW,
dr inż. Marek Niewiński,
dr inż. Andrzej Wielgus
dr inż. Adam Wojtasik*

Programowanie Obiektowe (PROO)
Object-oriented Programming

Poziom kształcenia: *I stopień*

Forma i tryb prowadzenia przedmiotu: *stacjonarna*

Kierunek studiów: *Elektronika*

Specjalność: *Elektronika i Fotonika*

Grupa przedmiotów:

Poziom przedmiotu: *podstawowy*

Status przedmiotu: *obowiązkowy*

Język przedmiotu: *polski*

Semestr nominalny: *4*

Minimalny numer semestru: *2*

Wymagania wstępne, zalecane przedmioty poprzedzające: *WINF, PAPRO, PROS*

Limit liczby studentów: *150*

Powód zgłoszenia przedmiotu: *modyfikacja i unowocześnienie programu studiów dla kierunku Elektronika, zmiana specjalności EiK na Elektronika i Fotonika*

Cel przedmiotu: *Celem przedmiotu jest pogłębienie wiedzy studentów na temat programowania obiektowego, zaznajomienie ich ze składnią wybranego języka obiektowego oraz nauka praktycznego wykorzystywania nabytej wiedzy implementacji rozwiązań problemów programistycznych w języku Java.*

Treść kształcenia:

Opis wykładu:

(2h) Elementy inżynierii oprogramowania dla programowania obiektowego

Metody projektowania programu obiektowego: zstępująca i wstępująca

Zasady SOLID

Zasady programowania techniką TDD

Implementacja paradygmatu obiektowego w języku Java

(2h) Środowisko

Maszyna wirtualna

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

- Bezpieczeństwo kodu*
- Gospodarka pamięcią (odśmiecanie)*
- Program w języku Java: Styl programowania, Organizacja projektu, Pliki Jar, Manifest, Pakiety, Dokumentacja*
- (1h) Podstawy składni*
 - Typy danych*
 - Obiekty niereferencyjne i referencyjne*
 - Tablice, łańcuchy znakowe, itp.*
 - Operatory, wyrażenia, organizacja pętli*
- (2h) Podstawy obiektowości*
 - Klasa, obiekt, metoda*
 - Enkapsulacja, ukrywanie danych*
 - Konstrukcja obiektu*
 - Relacje między obiektami – asocjacje i agregacje*
 - Dziedziczenie*
 - Klasa Object*
 - Klasy abstrakcyjne*
 - Interfejsy*
 - Polimorfizm*
- (1h) Obsługa sytuacji wyjątkowych*
 - Klasy i obiekty z interfejsem Throwable*
- (1h) Wejście/wyjście*
 - Koncepcja klas obudowujących*
 - Strumienie*
 - Serializacja obiektów*
 - Sieciowość – gniazda*
- (1h) Przetwarzanie współbieżne*
 - Interfejsy Runnable i Callable*
 - Obiekty wątków i uruchamianie zadań w wątkach*
 - Egzekutory*
 - Synchronizacja wątków*
- (2h) Interfejs graficzny (GUI) – programowanie zdarzeniowe*
 - Hierarchia klas interfejsu graficznego*
 - Rodzaje komponentów*
 - Rodzaje zdarzeń*
 - Wzorzec Obserwator - nasłuchiwanie i obsługa zdarzeń*
 - Metodologia JavaBeans*
- (2h) Przykładowy projekt (studium przypadku):*
 - Wielowątkowy sieciowy serwer z klientami wyposażonymi w GUI*
- (2h) Programowanie mobilne*
 - Java a system Android*
 - Podstawowe konstrukcje i zasady*
 - Przykładowa aplikacja*
- (2h) Implementacja paradygmatu obiektowego w innych językach programowania na przykładzie C++*
 - Obiektość w języku C++ – porównanie z językiem Java*

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

Wieloparadygmatowość programowania w C++

Laboratorium:

- (2h) Pierwsza aplikacja w Javie – kompilacja i uruchamianie z poziomu konsoli. Używanie zmiennych typów podstawowych. Definiowanie klas, tworzenie obiektów, wywoływanie metod i przekazywanie do nich argumentów. IDE i praca z debuggerem.*
- (2h) Zadany problem programistyczny: model obiektowy, UML, tworzenie testów jednostkowych. Wykorzystanie metodyki SOLID – klasy abstrakcyjne i interfejsy.*
- (2h) GUI, mechanizmy wielowątkowości.*
- (2h) Obsługa operacji wejścia/wyjścia, serializacja obiektów, komunikacja sieciowa*
- (2h) Aplikacja mobilna na platformę android.*

Projekt:

Zadania programistyczne realizowane zespołowo (grupy dwuosobowe – skład losowany).

Tematyka zadań ustalana indywidualnie dla każdej z grup:

tematy pochodzące od studentów – po akceptacji opiekuna

tematy narzucone przez opiekuna – np. symulacja i wizualizacja wybranego zjawiska fizycznego z dziedziny szeroko rozumianej elektroniki

Tematy rozdanie lub zatwierdzenie – 2 tydzień semestru

Etapy kontrolne – zatwierdzane przez opiekuna:

Specyfikacja w formie dokumentu README.md (opis funkcjonalności, scenariusz użycia, diagram UML, prototyp GUI) – w formie repozytorium git - 4 tydzień semestru

Wersja testowa – 8 tydzień semestru

Wersja ostateczna – 14 tydzień semestru

Egzamin: *nie*

Literatura:

Cay S. Horstmann *Java. Podstawy. Wydanie X*

Bruce Eckel *Thinking in Java. Edycja polska. Wydanie IV*

[Erich Gamma](#), [Richard Helm](#), [Ralph Johnson](#), [John Vlissides](#) *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku*

Robert C. Martin *Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki*

Oprogramowanie:

Wymiar godzinowy zajęć:	W	C	L	P	
	4/3	-	2/3	1	(45h/sem.)

Wymiar w jednostkach ECTS: 3

Liczba godzin pracy studenta związanych z osiągnięciem efektów kształcenia (opis):

- 1. liczba godzin kontaktowych – 40 godz., w tym*

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

- *obecność na wykładach 20 godz.,*
 - *obecność na ćwiczeniach audytoryjnych 0 godz.,*
 - *obecność na laboratorium 10 godz.,*
 - *udział w konsultacjach 10 godz.*
2. *praca własna studenta – 40 godz., w tym*
- *przygotowanie do ćwiczeń 0 godz.,*
 - *przygotowanie do laboratoriów 10 godz.,*
 - *przygotowanie do kolokwiów 10 godz.,*
 - *wykonywania zadań projektowych 15 godz.,*
 - *przygotowanie sprawozdań (projekty i laboratoria) 5 godz.*

Łączny nakład pracy studenta wynosi 80 godz., co odpowiada 3 pkt ECTS.

Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich: 1,5 pkt ECTS, co odpowiada 40 godz. kontaktowym.

Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym: 1,5 pkt ECTS, co odpowiada 10 godz. ćwiczeń laboratoryjnych plus 10 godz. przygotowań do laboratorium oraz 20 godz. zadań projektowych

Efekty uczenia się:

Efekty kształcenia/uczenia się student, który zaliczył przedmiot:	forma zajęć/ technika kształcenia	sposób weryfikacji (oceny)	odniesienie do efektów uczenia się dla programu
WIEDZA			
<i>Ma uporządkowaną wiedzę o mechanizmach obiektowych w języku Java</i>	wykład	sprawdzian przeprowadzany w trakcie wykładu	K1_W03, K1_W04
<i>Rozumie zasady budowy aplikacji obiektowej w języku Java</i>	wykład	sprawdzian przeprowadzany w trakcie wykładu	K1_W03, K1_W04
<i>Zna i rozumie zasady implementacji paradygmatu obiektowego w językach programowania</i>	wykład	sprawdzian przeprowadzany w trakcie wykładu	K1_W03, K1_W04
UMIĘTNOŚCI			
<i>Potrafi zaproponować strukturę i hierarchię klas dla aplikacji tworzonej w języku Java</i>	laboratorium / projekt	ocena pracy w trakcie zajęć laboratoryjnych / ocena projektu	K1_U18
<i>Potrafi zaproponować obiektową architekturę wielowątkowej, sieciowej aplikacji tworzonej w języku Java</i>	laboratorium / projekt	ocena pracy w trakcie zajęć laboratoryjnych / ocena projektu	K1_U18
<i>Potrafi wykorzystać dostępne w języku Java klasy i obiekty interfejsu graficznego</i>	laboratorium / projekt	ocena pracy w trakcie zajęć laboratoryjnych / ocena projektu	K1_U18
KOMPETENCJE SPOŁECZNE			
<i>Potrafi pracować w grupie oraz wspólnie rozwiązywać problemy</i>			K1_K03
<i>Jest odpowiedzialny za projekt oraz własny kod, podejmuje zobowiązania, z których jest w stanie się wywiązać</i>			K1_K04
<i>potrafi zarządzać swoim czasem oraz ma umiejętność planowania w czasie pracy własnej oraz pracy członków zespołu i dotrzymuje terminów</i>			K1_K03, K1_K04