

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu: Inżynieria oprogramowania

Rok akademicki: 2016/2017 Kod: MEI-1-516-s Punkty ECTS: 3

Wydział: Inżynierii Metali i Informatyki Przemysłowej

Kierunek: Edukacja Techniczno - Informatyczna Specjalność: —

Poziom studiów: Studia I stopnia Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 5

Strona www: —

Osoba odpowiedzialna: prof. nadzw. dr hab. inż. Banaś Krzysztof (banas@metal.agh.edu.pl)

Osoby prowadzące: prof. nadzw. dr hab. inż. Banaś Krzysztof (banas@metal.agh.edu.pl)

Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W001	Student posiada wiedzę na temat procesu wytwarzania oprogramowania oraz rozumie znaczenie podziału procesu wytwarzania oprogramowania na czynności związane ze zebraniem wymagań, specyfikacją, projektowaniem, implementacją, testowaniem i wdrażaniem.		Egzamin
M_W002	Student posiada wiedzę na temat projektowania oprogramowania oraz zna typy diagramów i narzędzia UML wspierające prace projektowe.		Egzamin
M_W003	Student wie jakie kryteria powinny spełniać poprawne projekty oprogramowania oraz rozumie znaczenie podstawowych mechanizmów obiektowości.		Egzamin
M_W004	Student zna i rozumie ideę komponentów oprogramowania wielokrotnego użytku, wzorców projektowych oraz refaktoryzacji		Egzamin
Umiejętności			
M_U001	Student umie wykonać projekt architektury aplikacji z wykorzystaniem narzędzi UML oraz diagramów UML.		Projekt

M_U002	Student potrafi prawidłowo oraz efektywnie wykorzystać mechanizmy abstrakcji, enkapsulacji oraz polimorfizmu podczas projektowania prostej aplikacji zgodnie z paradygmatem obiektowym oraz techniką wzorców projektowych.		Projekt
M_U003	Student potrafi wykorzystać narzędzia automatycznie generujące kod źródłowy na podstawie diagramów klas UML, potrafi dokumentować kod źródłowy oraz używać narzędzi do generacji dokumentacji oraz kontroli wersji.		Projekt
Kompetencje społeczne			
M_K001	Student potrafi pracować w zespole podczas prac nad aplikacją, a także potrafi uzasadnić i obronić rozwiązania przyjęte podczas prac nad aplikacją		Wykonanie projektu, Zaangażowanie w pracę zespołu

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Student posiada wiedzę na temat procesu wytwarzania oprogramowania oraz rozumie znaczenie podziału procesu wytwarzania oprogramowania na czynności związane ze zebraniem wymagań, specyfikacją, projektowaniem, implementacją, testowaniem i wdrażaniem.	+	-	-	-	-	-	-	-	-	-	-
M_W002	Student posiada wiedzę na temat projektowania oprogramowania oraz zna typy diagramów i narzędzia UML wspierające prace projektowe.	+	-	-	-	-	-	-	-	-	-	-
M_W003	Student wie jakie kryteria powinny spełniać poprawne projekty oprogramowania oraz rozumie znaczenie podstawowych mechanizmów obiektowości.	+	-	-	-	-	-	-	-	-	-	-
M_W004	Student zna i rozumie ideę komponentów oprogramowania wielokrotnego użytku, wzorców projektowych oraz refaktoryzacji	+	-	-	-	-	-	-	-	-	-	-
Umiejętności												

M_U001	Student umie wykonać projekt architektury aplikacji z wykorzystaniem narzędzi UML oraz diagramów UML.	-	-	-	+	-	-	-	-	-	-	-
M_U002	Student potrafi prawidłowo oraz efektywnie wykorzystać mechanizmy abstrakcji, enkapsulacji oraz polimorfizmu podczas projektowania prostej aplikacji zgodnie z paradygmatem obiektowym oraz techniką wzorców projektowych.	-	-	-	+	-	-	-	-	-	-	-
M_U003	Student potrafi wykorzystać narzędzia automatycznie generujące kod źródłowy na podstawie diagramów klas UML, potrafi dokumentować kod źródłowy oraz używać narzędzi do generacji dokumentacji oraz kontroli wersji.	-	-	-	+	-	-	-	-	-	-	-
Kompetencje społeczne												
M_K001	Student potrafi pracować w zespole podczas prac nad aplikacją, a także potrafi uzasadnić i obronić rozwiązania przyjęte podczas prac nad aplikacją	-	-	-	+	-	-	-	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Wykład

1. Modele cyklu życia oprogramowania.
2. Odkrywanie wymagań, zarządzanie procesem określania wymagań.
3. Język UML. Scenariusze i diagramy przypadków użycia UML.
4. Analiza i modelowanie strukturalne – diagramy czynności UML.
5. Modele systemów – diagramy stanów UML.
6. Modelowanie architektury systemów. Diagramy komponentów i pakietów UML.
7. Analiza i modelowanie obiektowe – diagramy statyczne klas i obiektów UML.
8. Zasady tworzenia przejrzystych, zrozumiałych i efektywnych zorientowanych obiektowo projektów oprogramowania.
9. Analiza i modelowanie obiektowe – diagramy dynamiczne interakcji, przebiegu i kooperacji.
10. Ponowne wykorzystanie kodu – inżynieria komponentowa, refaktoryzacja, szkielety.
11. Wzorce projektowe
12. Implementacja oprogramowania – metody i procesy, szybkie prototypowanie.
- Niezawodność oprogramowania
13. Testowanie oprogramowania – walidacja i weryfikacja. Dokumentacja, wdrożenie i konserwacja oprogramowania
14. Zarządzanie projektem informatycznym.
15. Przykładowe metodologie wytwarzania oprogramowania

Ćwiczenia projektowe

1. Opracowanie specyfikacji prostej aplikacji.
2. Opracowanie wymagań funkcjonalnych i нефункциональных.
3. Opracowanie scenariuszy i diagramów UML przypadków użycia.
4. Opracowanie diagramów czynności UML.
5. Opracowanie projektu architektury aplikacji z wykorzystaniem odpowiednich diagramów UML.
6. Identyfikacja oraz wykorzystanie wzorców projektowych.
7. Generacja kodu źródłowego na podstawie diagramów klas UML.
8. Wykorzystanie narzędzi wspierających tworzenie dokumentacji aplikacji.

Sposób obliczania oceny końcowej

Średnia ważona ocen z egzaminu (66%) i projektu (34%) – po uzyskaniu co najmniej 3.0 z każdej z nich

Wymagania wstępne i dodatkowe

Zgodnie z Regulaminem Studiów AGH podstawowym terminem uzyskania zaliczenia jest ostatni dzień zajęć w danym semestrze. Termin zaliczenia poprawkowego (tryb i warunki ustala prowadzący moduł na zajęciach początkowych) nie może być późniejszy niż ostatni termin egzaminu w sesji poprawkowej (dla przedmiotów kończących się egzaminem) lub ostatni dzień trwania semestru (dla przedmiotów niekończących się egzaminem).

Zalecana literatura i pomoce naukowe

1. Booch G., Jacobson I., Rumbaugh J., „UML przewodnik użytkownika” WNT 2002
2. Gamma E., Helm R., Johnson R., Vlissides J., „Wzorce projektowe. Elementy programowania obiektowego wielokrotnego użytku”, WNT 2005
3. Steve McConnell „Kod doskonały. Jak tworzyć oprogramowanie pozbawione błędów. Wydanie II”
4. Hunt A., Thomas D., „The Pragmatic Programmer: From Journeyman to Master”
5. Martin Fowler „UML Distilled”

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Nowe możliwości MES i metody alternatywne dla MES — New capabilities of FEM and methods alternative for FEM / T. Burczyński, K. BANAŚ, M. PIETRZYK // W: PLASTMET'2008 : zintegrowane studia podstaw deformacji plastycznej metali : VI seminarium naukowe : 25 listopada - 28 listopada 2008 Łańcut - Zamek / Sekcja Teorii Procesów Przeróbki Plastycznej Komitetu Metalurgii PAN, Sekcja Mechaniki Materiałów Komitetu Mechaniki PAN. — [Polska : PAN, 2008]. — S. [1-2]

Rheological models of blood: sensitivity analysis and benchmark simulations / Danuta SZELIGA, Piotr MACIOŁ, Krzysztof BANAŚ, Magdalena KOPERNIK, Maciej PIETRZYK // W: NUMIFORM 2010 : the 10th international conference on Numerical Methods in Industrial Forming Processes : Pohang, Republic of Korea : June 13-17, 2010 : book of abstracts / eds. Frédéric Barlat, Young Hoon Moon, Myoung-Gyu Lee ; Pohang University of Science and Technology. — S. l. : s. n., 2010. — S. 77. — Pełny tekst W: NUMIFORM 2010 [Dokument elektroniczny] : proceedings of the 10th international conference on Numerical Methods in Industrial Forming Processes : Pohang, Republic of Korea : 13-17 June 2010. — Wersja do Windows. — Dane tekstowe / eds. Frédéric Barlat, Young Hoon Moon, Myoung-Gyu Lee; American Institute of Physics. — S. l. : American Institute of Physics. — 1 dysk optyczny. — s. 1184-1192. — Wymagania systemowe : Adobe Acrobat Reader ; napęd CD-ROM. — Bibliogr. s. 1192, Abstr. — (AIP Conference Proceedings ; vol. 1252). — ISBN 978-0-7354-0799-2

Study on development of an adaptive finite element - cellular automata model for austenite-ferrite phase transformation / Ł. MADEJ, P. Spytkowski, K. MICHALIK, F. Kružel, P. Macioł, K. BANAŚ, M. PIETRZYK // W: ECCM 2010 [Dokument elektroniczny] : IV European Conference on Computational Mechanics : solids, structures and coupled problems in engineering: France : May 16-21, 2010. — Wersja do Windows. — Dane tekstowe / ECCOMAS European Community on Computational Methods in Applied Sciences. — [S. l. : s. n, 2010]. — 1 dysk optyczny. — S. 1-2. — Wymagania systemowe: Adobe Acrobat Reader ; napęd CD-ROM. — Bibliogr. s. 2.

Informacje dodatkowe

Brak

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	28 godz
Udział w ćwiczeniach projektowych	28 godz
Wykonanie projektu	10 godz
Przygotowanie do zajęć	10 godz
Samodzielne studiowanie tematyki zajęć	10 godz
Sumaryczne obciążenie pracą studenta	86 godz
Punkty ECTS za moduł	3 ECTS