

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu zajęć:	Inżynieria oprogramowania				
Rok akademicki:	2019/2020	Kod:	RAIR-1-605-n	Punkty ECTS:	3
Wydział:	Inżynierii Mechanicznej i Robotyki				
Kierunek:	Automatyka i Robotyka	Specjalność:	—		
Poziom studiów:	Studia I stopnia	Forma studiów:	Niestacjonarne		
Język wykładowy:	Polski	Profil:	Ogólnoakademicki (A)	Semestr:	6
Strona www:	—				
Prowadzący moduł:	dr inż. Miękina Lucjan (miekina@agh.edu.pl)				

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Celem kursu jest zapoznanie studentów z nowoczesnym podejściem do tworzenia systemów informatycznych o średniej i dużej skali opartym o metody obiektowe i model w języku UML. Kurs obejmuje procesy inżynierii oprogramowania, analizę wymagań, modelowanie w języku UML, obiektowe wzorce projektowe, uruchamianie i testowanie, pracę grupową i ciągłą integrację oprogramowania.

Opis efektów uczenia się dla modułu zajęć

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Powiązania z KEU	Sposób weryfikacji i oceny efektów uczenia się osiągniętych przez studenta w ramach poszczególnych form zajęć i dla całego modułu zajęć
Wiedza: zna i rozumie			
M_W001	zna i rozumie zasady obiektowego podejścia do tworzenia oprogramowania	AIR1A_W12	Kolokwium, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W002	zna i rozumie pojęcie systemu informatycznego i główne problemy związane z procesem jego rozwoju	AIR1A_W12	Kolokwium, Wykonanie ćwiczeń laboratoryjnych
M_W003	zna ogólną organizację procesu tworzenia i rozwoju oprogramowania; zna wiodące techniki i narzędzia typu CASE wspierające ten proces	AIR1A_W12	Kolokwium
M_W004	zna podstawowe założenia koncepcji architektury systemu opartej na modelu (MDA), zna własności i rolę języka UML w tym kontekście	AIR1A_W12	Kolokwium, Wykonanie ćwiczeń laboratoryjnych

M_W005	zna i rozumie strukturę modelu systemu informatycznego w języku UML 2.*	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_W006	zna i rozumie przeznaczenie diagramów przypadków użycia, stosowane symbole i ich znaczenie	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W007	zna i rozumie przeznaczenie diagramów sekwencji, stosowane symbole i ich znaczenie, zna rolę diagramu sekwencji w procesie modelowania struktury na podstawie przypadków użycia, zna rolę diagramu sekwencji w procesie modelowania interakcji realizujących przypadki użycia	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W008	zna i rozumie przeznaczenie diagramów klas i obiektów, stosowane symbole i ich znaczenie	AIR1A_W12	Kolokwium, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W009	zna i rozumie przeznaczenie diagramów maszyny stanowej, stosowane symbole i ich znaczenie	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W010	zna i rozumie przeznaczenie diagramów czynności, stosowane symbole i ich znaczenie	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W011	zna i rozumie przeznaczenie diagramów interfejsu użytkownika (paneli), stosowane symbole, ich znaczenie i sposób powiązania z innymi elementami modelu	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W012	zna i rozumie podstawowe obiektowe wzorce projektowe (Singleton, Multipleton, Iterator, Obserwator, Stan, Composite)	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
M_W013	zna podstawowe cykle życia systemów i metodyki projektowe	AIR1A_W12	Kolokwium
M_W014	zna podstawowe metody uruchamiania i testowania systemów informatycznych	AIR1A_W12	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
Umiejętności: potrafi			
M_U001	potrafi zastosować zasady obiektowego podejścia do tworzenia oprogramowania	AIR1A_U04, AIR1A_U09	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U002	potrafi dokonać analizy problemu i zaplanować ogólny przebieg jego rozwiązania w postaci systemu informatycznego, dobrać technologie i narzędzia jego realizacji	AIR1A_U04, AIR1A_U09	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U003	potrafi zbudować model systemu informatycznego w języku UML, wykorzystując diagramy przypadków użycia, sekwencji, klas i obiektów, maszyny stanowej i czynności	AIR1A_U04, AIR1A_U09	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych

M_U004	potrafi dokonać uruchomienia i symulacji modelu w środowisku udostępnianym przez stosowane narzędzia (IBM Rhapsody lub inne)	AIR1A_U04, AIR1A_U09	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U005	potrafi zrealizować uruchamianie i testowanie systemu różnymi metodami (logowanie danych, debugger, profiler, asercje, kod testujący, automatyczne narzędzia testujące)	AIR1A_U04, AIR1A_U09	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
M_U006	potrafi dobrać i zastosować wzorce projektowe odpowiednio do rozwiązywanego problemu	AIR1A_U04, AIR1A_U09	Kolokwium, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego
Kompetencje społeczne: jest gotów do			
M_K001	potrafi zaplanować działania zmierzające do realizacji złożonego projektu systemu informatycznego	AIR1A_K02	Wykonanie projektu
M_K002	potrafi pracować samodzielnie i w małych zespołach nad realizacją zadania informatycznego	AIR1A_K02	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych

Liczba godzin zajęć w ramach poszczególnych form zajęć

Suma	Forma zajęć dydaktycznych										
	Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
24	8	0	8	8	0	0	0	0	0	0	0

Matryca kierunkowych efektów uczenia się w odniesieniu do form zajęć i sposobu zaliczenia, które pozwalają na ich uzyskanie

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Forma zajęć dydaktycznych										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
Wiedza: zna i rozumie												
M_W001	zna i rozumie zasady obiektowego podejścia do tworzenia oprogramowania	+	-	+	+	-	-	-	-	-	-	-

M_W002	zna i rozumie pojęcie systemu informatycznego i główne problemy związane z procesem jego rozwoju	+	-	+	+	-	-	-	-	-	-	-
M_W003	zna ogólną organizację procesu tworzenia i rozwoju oprogramowania; zna wiodące techniki i narzędzia typu CASE wspierające ten proces	+	-	+	+	-	-	-	-	-	-	-
M_W004	zna podstawowe założenia koncepcji architektury systemu opartej na modelu (MDA), zna własności i rolę języka UML w tym kontekście	+	-	+	+	-	-	-	-	-	-	-
M_W005	zna i rozumie strukturę modelu systemu informatycznego w języku UML 2.*	+	-	+	+	-	-	-	-	-	-	-
M_W006	zna i rozumie przeznaczenie diagramów przypadków użycia, stosowane symbole i ich znaczenie	+	-	+	+	-	-	-	-	-	-	-
M_W007	zna i rozumie przeznaczenie diagramów sekwencji, stosowane symbole i ich znaczenie, zna rolę diagramu sekwencji w procesie modelowania struktury na podstawie przypadków użycia, zna rolę diagramu sekwencji w procesie modelowania interakcji realizujących przypadki użycia	+	-	+	+	-	-	-	-	-	-	-
M_W008	zna i rozumie przeznaczenie diagramów klas i obiektów, stosowane symbole i ich znaczenie	+	-	+	+	-	-	-	-	-	-	-
M_W009	zna i rozumie przeznaczenie diagramów maszyny stanowej, stosowane symbole i ich znaczenie	+	-	+	+	-	-	-	-	-	-	-
M_W010	zna i rozumie przeznaczenie diagramów czynności, stosowane symbole i ich znaczenie	+	-	+	+	-	-	-	-	-	-	-
M_W011	zna i rozumie przeznaczenie diagramów interfejsu użytkownika (paneli), stosowane symbole, ich znaczenie i sposób powiązania z innymi elementami modelu	+	-	+	+	-	-	-	-	-	-	-
M_W012	zna i rozumie podstawowe obiektowe wzorce projektowe (Singleton, Multipleton, Iterator, Obserwator, Stan, Composite)	+	-	+	+	-	-	-	-	-	-	-
M_W013	zna podstawowe cykle życia systemów i metodyki projektowe	+	-	+	+	-	-	-	-	-	-	-
M_W014	zna podstawowe metody uruchamiania i testowania systemów informatycznych	+	-	+	+	-	-	-	-	-	-	-

Umiejętności: potrafi												
M_U001	potrafi zastosować zasady obiektowego podejścia do tworzenia oprogramowania	-	-	+	+	-	-	-	-	-	-	-
M_U002	potrafi dokonać analizy problemu i zaplanować ogólny przebieg jego rozwiązania w postaci systemu informatycznego, dobrać technologie i narzędzia jego realizacji	-	-	+	+	-	-	-	-	-	-	-
M_U003	potrafi zbudować model systemu informatycznego w języku UML, wykorzystując diagramy przypadków użycia, sekwencji, klas i obiektów, maszyny stanowej i czynności	-	-	+	+	-	-	-	-	-	-	-
M_U004	potrafi dokonać uruchomienia i symulacji modelu w środowisku udostępnianym przez stosowane narzędzia (IBM Rhapsody lub inne)	-	-	+	+	-	-	-	-	-	-	-
M_U005	potrafi zrealizować uruchamianie i testowanie systemu różnymi metodami (logowanie danych, debugger, profiler, asercje, kod testujący, automatyczne narzędzia testujące)	-	-	+	+	-	-	-	-	-	-	-
M_U006	potrafi dobrać i zastosować wzorce projektowe odpowiednio do rozwiązywanego problemu	-	-	+	+	-	-	-	-	-	-	-
Kompetencje społeczne: jest gotów do												
M_K001	potrafi zaplanować działania zmierzające do realizacji złożonego projektu systemu informatycznego	-	-	-	+	-	-	-	-	-	-	-
M_K002	potrafi pracować samodzielnie i w małych zespołach nad realizacją zadania informatycznego	-	-	-	+	-	-	-	-	-	-	-

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach dydaktycznych/praktyka	24 godz
Przygotowanie do zajęć	16 godz
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	20 godz
Samodzielne studiowanie tematyki zajęć	20 godz
Egzamin lub kolokwium zaliczeniowe	2 godz
Sumaryczne obciążenie pracą studenta	82 godz
Punkty ECTS za moduł	3 ECTS

Pozostałe informacje

Szczegółowe treści kształcenia w ramach poszczególnych form zajęć (szczegółowy program wykładów i pozostałych zajęć)

Wykład

Wykład dotyczy zagadnień:

- inżynieria oprogramowania jako dziedzina
- proces inżynierii oprogramowania
- narzędzia CASE
- modelowanie systemów w języku UML wraz z dyskusją przykładu kompletnego systemu
- diagramy przypadków użycia
- diagramy klas i obiektów
- diagramy zachowania i stanu
- metodyki realizacji systemów
- obiektowe wzorce projektowe
- inżynieria wahadłowa z wykorzystaniem narzędzi
- metody testowania oprogramowania

Ćwiczenia laboratoryjne

Ćwiczenia laboratoryjne obejmują:

- zapoznanie się z wybranym systemem wspierającym modelowanie i testowanie systemów (IBM Rhapsody)
- indywidualna praca studentów nad wspólnie realizowanym przykładem modelu systemu informatycznego w języku UML, z wykorzystaniem diagramów przypadków użycia, sekwencji, klas i obiektów, maszyny stanowej i czynności
- uruchomienie i symulację modelu w środowisku IBM Rhapsody
- testowanie z użyciem logowania danych, asercji
- testowanie jednostek (klas) z użyciem kodu testującego wygenerowanego automatycznie (JUnit)
- automatyczne testowanie w środowisku IBM Rhapsody z użyciem diagramów sekwencji

Ćwiczenia projektowe

Ćwiczenia projektowe obejmują wykonanie indywidualnie lub w grupach 2-osobowych modelu UML prostego programu organizowanego obiektowo. Przykładowe tematy:

- Obsługa złożonego urządzenia (pralka, bankomat, odtwarzacz plików muzycznych, itp.)
- System rezerwacji połączeń lotniczych i sprzedaży biletów
- Komis samochodowy
- Wypożyczalnia (książek, filmów, itp.)

Po zakończeniu modelowania zadaniem jest uruchomienie i symulacja modelu w środowisku IBM Rhapsody, a następnie przygotowanie przypadków testowych i przetestowanie wybranych aspektów systemu.

Metody i techniki kształcenia:

Wykład: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.

Ćwiczenia laboratoryjne: W trakcie zajęć laboratoryjnych studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia. Prowadzący stymuluje grupę do refleksji nad problemem, tak by otrzymane wyniki miały wysoką wartość merytoryczną.

Ćwiczenia projektowe: Studenci wykonują zadany projekt samodzielnie, bez większej ingerencji prowadzącego. Ma to wykształcić poczucie odpowiedzialności za pracę w grupie oraz odpowiedzialności za podejmowane decyzje.

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:

Warunkiem zaliczenia ćwiczeń laboratoryjnych jest wykonanie zadań zleconych w trakcie zajęć i zaliczenie testu znajomości języka UML.

Warunkiem zaliczenia ćwiczeń projektowych jest wykonanie projektu i przedstawienie go do oceny, połączone z uzasadnieniem zastosowanych rozwiązań.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:

Wykład:

- Obecność obowiązkowa: Nie

- Zasady udziału w zajęciach: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne:

- Obecność obowiązkowa: Tak

- Zasady udziału w zajęciach: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

Ćwiczenia projektowe:

- Obecność obowiązkowa: Tak

- Zasady udziału w zajęciach: Studenci wykonują prace praktyczne mające na celu uzyskanie kompetencji zakładanych przez sylabus. Ocenie podlega sposób wykonania projektu oraz efekt końcowy.

Sposób obliczania oceny końcowej

Średnia ważona z oceny ćwiczeń laboratoryjnych (40%) i projektu (60%).

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:

Samodzielne wykonanie zaległych zadań i przedstawienie do oceny.

Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności modułów

Umiejętność sprawnego programowania obiektowego w języku C++

Zalecana literatura i pomoce naukowe

- Booch G., Rumbaugh J., Jacobson I.: UML – przewodnik użytkownika. WNT, Warszawa 2002
- Gamma E., Helms R., Johnson R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995
- Miękina L.: Inżynieria Oprogramowania, UWND AGH, Kraków, 2009, SU 1707
- Binder R.: Testowanie systemów obiektowych . Warszawa, WNT 2003
- Stroustrup B.: Język C++. Warszawa, WNT 2002
- Gregoire M., Professional C++, Third Edition, John Wiley & Sons, Inc., 2014

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Miękina L.: Inżynieria Oprogramowania, UWND AGH, Kraków, 2009, SU 1707

Informacje dodatkowe

1. Kolokwium zaliczeniowe odnosi się do zagadnień poruszanych na ćwiczeniach i ma wpływ na ocenę z ćwiczeń laboratoryjnych