

Inżynieria oprogramowania

Instrukcja do laboratorium

rok akad. 2019/2020

Informacje podstawowe:

Celem laboratorium jest nabycie przez studentów praktycznej umiejętności wykonywania modeli analitycznych i projektowych systemu oraz posługiwania się narzędziami CASE (Computer Aided Software Engineering). Modele tworzone są w języku UML (Unified Modeling Language) za pomocą narzędzia **Enterprise Architect** (wersja 12).

Prowadzący zajęcia laboratoryjne:

- dr inż. Anna Bobkowska, annab@eti.pg.edu.pl, pok. 647.
- dr inż. Aleksander Jarzębowicz, olek@eti.pg.edu.pl, pok. 648.
- mgr inż. Magdalena Mazur-Milecka, magmilec@pg.edu.pl, pok. 106.
- mgr inż. Katarzyna Poniatowska, katarzynaponiatowska94@gmail.com, prac. zewn.
- mgr inż. Olga Springer, olga.springer@pg.edu.pl, pok. 648.
- dr inż. Andrzej Wardziński andrzej.wardzinski@eti.pg.edu.pl, pok. 646.

Materiały pomocnicze:

Podstawowym źródłem informacji jest treść wykładów przedmiotu oraz udostępniony przykładowy model UML. Dodatkowa literatura z zakresu laboratorium, pomocna w opanowaniu UML i modelowania, obejmuje następujące źródła:

1. Fowler M., Scott K., UML w kropelce (ang. *UML distilled*), LTP Oficyna Wydawnicza, 2005.
2. Booch G., Rumbaugh J, Jacobson I., UML Przewodnik użytkownika., WNT, 2002.
3. OMG Unified Modeling Language Specification, Version 2.5.
4. Maciaszek L., Requirements Analysis and System Design, Addison Wesley, 2007.
5. McLaughlin B., Pollice G., West D., Head First Object-Oriented Analysis and Design. Edycja polska (Rusz głową!), Helion, 2008.

Ramowy harmonogram zajęć w semestrze zimowym 2019/2020:

	Wstęp	Wizja systemu (6 pkt)		Przypadki użycia (6 pkt + 5 pkt)			Modelowanie obiektowe (6 pkt + 5 pkt)			Diagramy dynamiczne (6 pkt + 5 pkt)			Elementy projektowania (6 pkt + 5 pkt)				Rezerwa	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
PN		7-10	14-10	21-10	28-10	4-11	11-11	18-11	25-11	2-12	9-12	16-12	23-12	30-12	6-01	13-01	20-01	27-01
WT	1-10	8-10	15-10	22-10	29-10	5-11	12-11	19-11	26-11	3-12	10-12	17-12	24-12	31-12	7-01	14-01	21-01	28-01
ŚR	2-10	9-10	16-10	23-10	30-10	6-11	13-11	20-11	27-11	4-12	11-12	18-12	25-12	1-01	8-01	15-01	22-01	29-01
CZ	3-10	10-10	17-10	24-10	31-10	7-11	14-11	21-11	28-11	5-12	12-12	19-12	26-12	2-01	9-01	16-01	23-01	
PT	4-10	11-10	18-10	25-10	1-11	8-11	15-11	22-11	29-11	6-12	13-12	20-12	27-12	3-01	10-01	17-01	24-01	
	W	K	O	S W	K	O	S W	K	O	S W	K	O		S W	K	K	O	

S sprawdzian

W wprowadzenie do zadania

K konsultacja zadania

O ocena i omówienie zadania

kolorem czerwonym zaznaczono dni wolne

kolorem niebieskim zaznaczono dni ze zmienionym planem

Uwaga – powyższy harmonogram ma jedynie charakter ramowy, dokładne terminy sprawdzianów, terminów oddawania itp. ustalane są przez prowadzących poszczególne terminy zajęć laboratoryjnych.

Zasady zaliczenia:

1. Z przedmiotu „Inżynieria Oprogramowania” wystawiana jest jedna łączna ocena. Do uzyskania jest 50 pkt z egzaminu i 50 pkt z laboratorium, suma wyników studenta z tych dwóch części decyduje o ocenie końcowej.
2. W ramach laboratorium punkty przyznawane są za zadania projektowe oraz sprawdziany.
3. Do zaliczenia laboratorium konieczne jest zaliczenie wszystkich zadań i zdobycie przynajmniej połowy spośród dostępnej puli punktów (25 pkt).
4. Zadania projektowe realizowane są w 3-osobowych zespołach (ewentualnie 2-os.). Treść poszczególnych zadań opisana jest poniżej, do części dostępne są dodatkowe instrukcje. Dokładne zasady realizacji i oddawania zadań są ustalane indywidualnie z prowadzącymi poszczególne terminy laboratorium.
5. Zadania powinny być oddawane terminowo wg harmonogramu ustalonego przez prowadzącego. Opóźnienie skutkuje punktami ujemnymi (-1 pkt za każdy rozpoczęty tydzień opóźnienia).
6. Zadanie uznaje się za zaliczone, jeżeli zespół uzyskał za nie minimum 50% dostępnej liczby punktów nie licząc opóźnienia (tzn. zadanie oddane 3 tyg. po terminie i ocenione na 5 pkt na 6 możliwych jest uznane za zaliczone mimo wyniku $5-3=2$, ale zadanie oddane terminowo i ocenione na 2 pkt wymaga poprawy).
7. Terminy i zakres sprawdzianów są ustalone (patrz harmonogram) i będą dodatkowo omawiane przez prowadzących. Sprawdziany odbywają się tylko raz, nie przewiduje się poprawek. W przypadku nieusprawiedliwionej nieobecności na sprawdzianie punkty te przepadają.
8. Obecność na zajęciach laboratoryjnych jest obowiązkowa. Dopuszczalne są bez konsekwencji 2 nieusprawiedliwione nieobecności. Przy 3 nieobecnościach student otrzymuje -3 pkt do swojego wyniku z laboratorium, przy 4 nieobecnościach -7 pkt. Przy 5 nieobecnościach następuje wykluczenie z laboratorium z oceną niedostateczną.
9. Zwolnienia lekarskie powinny być przedstawiane prowadzącemu na najbliższych zajęciach po powrocie na uczelnię. W przypadku nieobecności na sprawdzianie, zwolnienie uprawnia do pisania sprawdzianu w innym terminie uzgodnionym z prowadzącym.
10. Oceny punktowe za zadania przyznawane są wszystkim członkom zespołu. Jednakże w przypadku zauważenia znaczącej różnicy w pracy poszczególnych osób, prowadzący może te oceny zróżnicować.
11. W ramach oceniania danego etapu prowadzący sprawdza również czy wprowadzone zostały sugerowane poprawki do etapów poprzednich. W przypadku braku takowych może obniżyć ocenę.
12. Próby oddawania cudzych prac i inne przypadki „academic dishonesty” skutkują natychmiastową oceną niedostateczną z laboratorium.
13. Osoby powtarzające laboratorium zobowiązane są do wyboru nowego tematu, nie dopuszcza się ponownej realizacji tematu z poprzedniej edycji lub bardzo podobnego.
14. Laboratorium jest zamykane wraz z końcem semestru. Po tym czasie żadne zaległe zadania nie będą przyjmowane.

Zadania

Ogólny opis poszczególnych zadań projektowych – szczegóły będą przekazywane przez prowadzących w ramach wprowadzenia do danego zadania.

Z1 - Wizja systemu

Cele: Wybór tematu, zdefiniowanie zakresu dalszych prac, opracowanie celów, wymagań i ograniczeń rozważanego systemu

Zadania:

1. Ustalenie tematu z prowadzącym zajęcia.
2. Wyznaczenie granicy projektowanego systemu.
3. Przedstawienie prowadzącemu wstępnych założeń dot. wizji systemu.
4. Wykonanie opisu zadanego systemu według szablonu (plik IO_WizjaSystemu_2019.doc).
5. Zweryfikowanie z prowadzącym przyjętych założeń dotyczących wizji systemu (po sprawdzeniu wyników tego etapu)

Produkty: Dokument wizji systemu.

Z2 - Przypadki użycia systemu

Cele: Identyfikacja wymagań funkcjonalnych systemu i przedstawienie ich za pomocą przypadków użycia

Zadania:

1. Identyfikacja i opis aktorów.
2. Identyfikacja systemowych (funkcjonalnych) przypadków użycia.
3. Ustrukturyzowane opisy przypadków użycia
4. Identyfikacja powiązań pomiędzy aktorami a przypadkami użycia ("comunicate") oraz pomiędzy przypadkami użycia ("include", "extend") i przedstawienie tego na diagramie przypadków użycia.

Produkty: Diagram przypadków użycia z opisami w Enterprise Architect.

Z3 - Diagram klas

Cele: Stworzenie modelu klas pokrywającego wszystkie istotne z punktu widzenia projektu klasy i powiązania pomiędzy nimi. Po wstępnej identyfikacji klas w systemie poprzez kolejne iteracje należy dążyć do stworzenia modelu pokrywającego pełny zakres funkcjonalny systemu określony w przypadkach użycia

Zadania:

1. Identyfikacja klas potrzebnych do funkcjonowania systemu.
2. Identyfikacja związków między klasami i określenie ich semantyki (nazwa, liczność).
3. Identyfikacja atrybutów klas oraz w możliwym zakresie usług klas (operacje).
4. Opis klas oraz ich atrybutów i operacji (w zakresie niezbędnym do zrozumienia).

Produkty: Diagram klas w Enterprise Architect.

Z4 - Modelowanie dynamiki systemu

Cele: Identyfikacja i przedstawienie tego, w jaki sposób obiekty klas składowych systemu poprzez współpracę i interakcje pozwalają na zrealizowanie wymaganej funkcjonalności systemu. Identyfikacja i przedstawienie złożonego zachowania obiektów.

Zadania:

1. Identyfikacja scenariuszy interakcji na podstawie scenariuszy (podstawowych i alternatywnych) z przypadków użycia.
2. Zidentyfikowanie zdarzeń (komunikatów) w systemie, ich parametrów oraz obiektów będących nadawcą i odbiorcą komunikatu.
3. Przygotowanie diagramów sekwencji, komunikacji i czynności.
4. Przygotowanie diagramu stanów dla wybranej klasy o skomplikowanym zachowaniu.

Produkty: Diagramy sekwencji (trzy, dla wybranych przypadków użycia), diagram komunikacji (jeden, dla wybranego przypadku użycia), diagram czynności (jeden, do wyboru - dla procesu biznesowego albo dla wybranego przypadku użycia), diagram stanów (dla klasy o najbardziej skomplikowanym zachowaniu)

Z5 - Elementy projektowania

Cele: Przejście od ogólniejszego modelu analitycznego do modelu projektowego ukierunkowanego na implementację w wybranych technologiach

Zadania:

1. Podział systemu na warstwy i podsystemy-partycje.
2. Określenie rozwiązań technologicznych do implementacji systemu.
3. Określenie architektury fizycznej i przygotowanie diagramu rozmieszczenia/wdrożenia.
4. Projekt interfejsu użytkownika dla wybranego podsystemu.
5. Uszczegółowienie klas i wygenerowanie szkieletu kodu dla wybranego podsystemu.
6. Projekt bazy danych.

Produkty: Raport przygotowany na podstawie udostępnionego szablonu (plik IO_Elementy_Projektowania_2019.doc), zawierający wyniki ww. zadań.