

Projektowanie oprogramowania

1. Warunki wstępne

Warunkiem uczestnictwa w zajęciach jest zaliczenie przedmiotu: Podstawy inżynierii oprogramowania (ćwiczenia). Zajęcia składają się z 30 godzin wykładu i 30 godzin laboratorium.

2. Wykład – katalog kursów

2.1. Egzamin

Pisemny. Test wielokrotnego wyboru i zadania otwarte. Należy zdobyć co najmniej połowę punktów, aby zdać egzamin.

2.2. Wymagania wstępne do zaliczenia wykładu

Aby podejść do egzaminu student *nie musi zaliczyć* zajęć towarzyszących.

2.3. Zasady dyskwalifikacji

Student otrzymuje ocenę niedostateczną, jeżeli w czasie egzaminu zostanie przyłapany na korzystaniu ze ściąg lub pomocy kolegów.

3. Laboratorium

Celem laboratorium jest nabycie umiejętności systematycznej specyfikacji i dokumentacji oprogramowania z wykorzystaniem UML oraz zapoznanie się z narzędziami do modelowania.

Uwaga. Prowadzący laboratorium mogą ustalić dodatkowe wymagania, niesprzeczne z podanymi poniżej.

3.1. Wymagania wstępne do uzyskania zaliczenia laboratorium

Student może opuścić co najwyżej 2 laboratoria. W wyjątkowych sytuacjach, np. długotrwałej choroby – 3.

3.2. Realizacja laboratorium

Studenci są oceniani indywidualnie, lecz mogą pracować w parach. Studenci przychodzą przygotowani na laboratoria i pracują w trakcie zajęć oraz w domu z wykorzystaniem wersjonowanego repozytorium kodu (kont SVN). Konta SVN są przydzielne na pierwszych zajęciach. Studenci weryfikują dostęp do konta przed drugimi zajęciami umieszczając w repozytorium plik „!.txt” zawierający ich imiona, nazwiska i numery indeksów.

Aby sprawdzić przygotowanie studentów do zajęć prowadzący ma prawo przeprowadzić na początku zajęć niezapowiedzianą kartkówkę z podstawowych pojęć, których znajomość jest wymagana do realizacji danego etapu. Niezaliczenie kartkówki oznacza nieobecność na zajęciach (powyżej 3 nieobecności student nie otrzymuje zaliczenia). Plan prac do wykonania w ramach kolejnych laboratoriów, terminy oddawania wyników prac, punktacja wyników prac, narzędzia realizacji etc. są przedstawione w poniżej tabeli.

Lab#	Temat	Uwagi	Rezultaty	Narzędzia	Elementy podlegające sprawdzeniu
1	Zajęcia organizacyjne	Przedstawienie tematu wiodącego dla grupy laboratoryjnej. Omówienie sposobu organizacji zajęć. BHP			
L2 – L4 Opracowanie koncepcji systemu (razem: 20 p.); deadline oddania wyników prac L4					
2	Wizja. Słownik.	Studenci indywidualnie budują wizję i słownik dla tematu wiodącego.	Wizja (5 p). Słownik (5 p).	Word – zgodnie z szablonem (użytkownicy, cechy, etc.) Visual Paradigm	Spójność (wewnętrzna, zewnętrzna), niesprzeczność
3-4	Domenowy diagram klas + reguły biznesowe (min. 5).	Studenci dla tematu wiodącego na podstawie zdefiniowanego słownika budują diagram klas oraz definiują min. 10 reguł biznesowych (różnych typów) w języku naturalnym	Model domenowy: - diagram klas (5 p.) - reguły biznesowe (różnych typów) (5 p.)	Diagram klas Visual Paradigm Reguły biznesowe: Word	Diagram klas – zgodność z dziedziną i słownikiem; poprawność diagramu Poprawność reguł biznesowych
L5 – L7 Specyfikacja wymagań (razem 30p. + 5p*); deadline oddania kompletnej dokumentacji L7					
5	Specyfikacja wymagań funkcjonalnych i niefunkcjonalnych Model przypadków użycia	Studenci definiują na podstawie wizji wymagania funkcjonalne – opisy streszczające lub historie użytkownika (min. 10) i niefunkcjonalne (min. 3) Na podstawie zidentyfikowanych wymagań studenci budują model przypadków użycia i krótko je opisują w narzędziu.	Wymagania funkcjonalne i niefunkcjonalne (5 p.) Model PU (5 p.)	Word lub diagram wymagań Visual Paradigm (model + opisy w sekcji dokumentacji)	Model PU nie musi być kompletny, ale powinien być spójny. Należy zwrócić uwagę na poprawność śladowań.
6	Specyfikacja przypadków użycia	Dla dwóch ustalonych z prowadzącym PU studenci piszą w narzędziu scenariusze przypadków (główny i alternatywne)	Specyfikacja PU dla dwóch PU – wątek główny + alternatywne (5p. + 5p.)	Visual Paradigm – specyfikacja PU	Kompletność. Poprawność.
7	Prototyp interfejsu	Dla dwóch ustalonych na L6 przypadków użycia student buduje prototyp interfejsu (przynajmniej dla wątków głównych); mogą to być rysunki odręczne lub wykonane w dowolnym narzędziu, np. w NetBeans, Visio etc.	Prototyp interfejsu dla wątków głównych (3p. + 3p.); gdy prototypy zrobione w narzędziu (2p. + 2p.); gdy dodatkowo prototypy dla wątków alternatywnych (5p.*)	Dowolne dostępne narzędzie	Możliwość realizacji PU z wykorzystaniem zaproponowanego interfejsu.
L8 – L10 Projekt ogólny i szczegółowy (razem 25p. +10p.*); deadline oddania kompletnej dokumentacji L10					

8	Architektura systemu. Model danych.	<p>Student – na podstawie wymagań funkcjonalnych i нефункциональных proponuje ogólną logiczną architekturę systemu i dokumentuje ją diagramem pakietów lub komponentów.</p> <p>Na tym etapie powstaje model danych (diagram klas), obejmujący co najmniej dane potrzebne do realizacji PU (projekt szczegółowy; atrybuty, liczności, nazwy ról etc.)</p>	<p>Architektura logiczna (5 p.)</p> <p>Model danych – diagram klas (5 p.)</p>	<p>Visual Paradigm</p> <p>Visual Paradigm</p>	<p>Sprawdzenie pokrycia wymagań.</p> <p>Kompletność. Zgodność z interfejsem. Spójność.</p>
9	Realizacja PU.	<p>Student, dla dwóch wskazanych PU, projektuje ich realizację, definiując diagramy klas i sekwencji. Realizacja obejmuje co najmniej wątki główne. Klasy muszą zostać umieszczone w architekturze.</p> <p>Dodatkowo, student może udokumentować mechanizm architektoniczny, np. sposób utrwalania danych, w tym wykorzystanie wzorców projektowych.</p>	<p>Realizacja PU – wątki główne (5 p. + 5 p.)</p> <p>Wątki alternatywne (5 p.*)</p> <p>Mechanizm architektoniczny (5 p.*)</p>	<p>Visual Paradigm – diagram klas, diagram sekwencji</p>	
10	Szczegóły klas.	<p>Dla klas sterujących, zidentyfikowanych w L9, student dodaje szczegóły (atrybuty, nazwy operacji)</p>	<p>Szczegóły klas sterujących (5 p.)</p>	<p>Visual Paradigm – definicje klas</p>	<p>Ocena szczegółów klas „odroczone” do fazy implementacji. Tam oceniana spójność.</p>
L11 – L14 Implementacja + testy + ocena jakości (razem 30p. + 15p.*); deadline oddania kompletnej dokumentacji L14					
11	Implementacja interfejsu zgodnie z projektem	<p>Na podstawie prototypu interfejsu student buduje interfejs w Javie. W przypadku wykrycia rozbieżności pomiędzy prototypem i implementacją – student dokumentuje różnice.</p>	<p>Kod źródłowy w Java (za zgodą prowadzącego dopuszczalny inny język) dla każdego PU (7.5 p. + 7.5 p.), który da się skompilować i uruchomić + max. 5 p.* w przypadku systemów rozproszonych</p>	<p>Eclipse lub NetBeans</p>	<p>Odpowiedź na pytania: - czy działa zgodnie ze specyfikacją (przypadki użycia, prototyp interfejsu) - czy kod źródłowy zgodny z projektem</p>
12	Podłączenie logiki aplikacji do interfejsu	<p>Implementacja logiki aplikacji (zgodnie z projektem) – wątki główne + alternatywne</p> <p>Uwaga. Implementacja może być uproszczona, tzn. nie musi działać w środowisku rozproszonym, może korzystać z mechanizmów serializacji do utrwalania danych. W przypadku implementacji</p>			

		przekraczających te wymagania, np. komunikacja z serwerem bazy danych, student może zdobyć dodatkowe punkty.			
13	Testy jednostkowe dla klas sterujących (testy dla min. 2 metod)	Definicja testów jednostkowych dla wybranych metod klas	Kod z testami (5 p.)	Eclipse lub NetBeans	Czy testy obejmują sytuacje typowe, brzegowe.
14	Przypadki testowe dla PU	Przypadki testowe (zapisane w postaci tekstowej) dla zrealizowanych PU	Tekst z przypadkami testowymi (5p. + 5p.)	Word	Odpowiedź na pytania: - czy przypadek testowy zgodny z interfejsem - czy system zachowuje się zgodnie z przypadkami testowymi
	Automatyzacja testów funkcjonalnych	Automatyzacja testów dla jednego przypadku użycia	Demonstracja + skrypt testowy (5 p.*)	Np. Selenium lub Functional Tester	
	Badanie jakości projektu	Wyliczenie i ocena metryk zaimplementowanej aplikacji	Raport z wynikami metryk (5 p.*)	JJDepend	Czy dokonano oceny jakości kodu i wyciągnięto wnioski
L15 Rezerwa + oceny					

Każdy milestone (1-4) umieszczony jest w oddzielnym katalogu na SVN („milestone1”... „milestone4”) i zawiera plik „odpowiedzialnosc.txt” zawierający dokładny opis, kto, za co odpowiadał i co zrobił. Artefakty umieszczane są w odpowiednich podkatalogach np. wizja i słownik w podkatalogu \milestone1\lab2 a prototyp interfejsu w podkatalogu \milestone2\lab7.

Ze względu na możliwe problemy z kompatybilnością, artefakty przygotowane w Visual Paradigm proszę umieszczać na SVN również w formacie graficznym (png, jpg, pdf). Artefakty powinny być czytelne i możliwe do obejrzenia w całości, lub niemal w całości, na monitorze (12 cale/22 cale).

3.3. Zasady dyskwalifikacji

Student otrzyma ocenę niedostateczną z laboratorium, jeżeli przedłoży prowadzącemu rozwiązanie, którego nie jest autorem lub będące plagiatem lub kopią całości lub pewnej fazy innego rozwiązania.

3.4. Sposób oceniania

Studenci za wykonane prace otrzymują punkty. Maksymalna liczba punktów za dany artefakt jest podana w tabeli (patrz punkt 3.2). Prowadzący powinien przedstawić ocenę prac najpóźniej dwa tygodnie po oddaniu prac.

Gwiazdką oznaczono punkty za zadania dodatkowe (student decyduje o tym, czy je wykonać).

Ostateczne terminy oddawania prac są określone w tabeli.

Za oddanie prac po terminie student otrzymuje -5p. za każdy rozpoczęty tydzień spóźnienia.

Oddanej i ocenionej pracy nie można poprawiać w celu podniesienia otrzymanej zań punktacji. Terminy oddawania prac są ustalone tak, aby student przed oddaniem mógł skonsultować z prowadzącym jakość oddawanego artefaktu.

Jeżeli już oddana i oceniona praca, ze względu na popełnione błędy, może wpłynąć negatywnie na ocenę oddawanego dokumentu, student ma możliwość dokonania poprawek w oddanej pracy i przedłożenia obu prac (oddanej-poprawionej i bieżącej).

Studenci mogą zdobywać dodatkowe punkty za przygotowanie materiałów pomocniczych do poszczególnych laboratoriów, np. jak zdefiniować i uruchomić w Eclipse testy jednostkowe (max. 5p.* za opracowanie) – do uzgodnienia z prowadzącym.

Za wykonanie ćwiczeń student może zdobyć maksymalnie: 105p. + 30p.* + punkty za materiały pomocnicze* (max. 10 p.) – razem 145 p.

Proponowana skala ocen:

<0, 60) → ndst

[60, 70) → dst

[70, 80) → dst+

[80, 90) → db

[90, 100) → db+

[100, 110) → bdb

>= 111 → cel

Ocena celująca jest podstawą zwolnienia z egzaminu z oceną 5.0.

4. Tematy wiodące

Zostaną zaproponowane przez prowadzących laboratorium.