

KARTA PRZEDMIOTU

Nazwa przedmiotu: Inżynieria oprogramowania (InfAu>SI5-IO-19)

Nazwa w języku polskim:

Nazwa w jęz. angielskim: Software engineering

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

ZAL

Język wykładowy:

polski

Strona WWW:

<https://platforma2.polsl.pl/rau2/course/view.php?id=878>

Skrócony opis:

Przedmiot ma formę projektu. Jego zasadniczym elementem jest opracowanie specyfikacji systemu informatycznego. Na specyfikację składają się: specyfikacja wstępna, specyfikacja wymagań, model przypadków użycia, scenariusze, model klas, szkic architektury, oraz model zachowania. Podstawowym celem dydaktycznym przedmiotu jest nauka modelowania SI z zastosowaniem języka UML, a także nauka posługiwania się narzędziem CASE.

Dodatkowym elementem projektu jest implementacja zaprojektowanej wcześniej aplikacji. Celem tego etapu jest weryfikacja modelu oraz ćwiczenie umiejętności metodycznego testowania aplikacji.

Opis:

ECTS : 3

suma godzin: 80 (kontaktowo 30h, praca własna 50h)

projekt - 30h,

praca własna - 50h

w tym:

oprac. tzw. artefaktów - dokumentów stanowiących część projektu: 20h

implementacja programu: 30h.

Zajęcia mają formę projektu w wymiarze 2 godz. lekcyjne tygodniowo.

Student otrzymuje 3 pkt ECTS za zrealizowanie następujących zadań projektowych:

1 - przygotowanie wstępnej specyfikacji systemu. (6 godz. lek.)

2 - opracowanie modelu przypadków użycia. (6 godz. lek.)

3 - opracowanie modelu klas (+prototyp #1) (6 godz. lek.)

4 - przygotowanie modelu zachowania (+prototyp #2) (6 godz. lek.)

5 - przeprowadzenie implementacji i testów (6 godz. lek.)

Szczegółowe cele dydaktyczne to:

1. ćwiczenie umiejętności pozyskiwania informacji i formułowania wymagań;

2. nauka praktycznego posługiwania się notacjami, głównie językiem UML;

3. nauka posługiwania się narzędziami CASE w procesie tworzenia oprogramowania;

4. wyrobienie nawyku sporządzania specyfikacji programów przed rozpoczęciem implementacji;

5. zapoznanie studentów z metodyką projektowania oprogramowania (w szczególności wyrobienie umiejętności przechodzenia pomiędzy kolejnymi artefaktami od opisu słownego, poprzez wymagania, itd. – do implementacji);

6. ćwiczenie umiejętności implementacji programów na podstawie dokumentacji;

7. ćwiczenie umiejętności pracy zespołowej z podziałem na role projektowe;

Literatura:

• Perdita Stevens, "UML. Inżynieria oprogramowania", Helion, 2021.

• Grady Booch, Ivar Jacobson, James Rumbaugh, "The Unified Modeling Language User Guide", Addison Wesley, first edition 1999.

• Koszłajda Adam, "Zarządzanie projektami IT, przewodnik po metodykach", HELION, Gliwice, 2010.

• Pichler Roman, "Zarządzanie projektami ze Scrumem", edycja polska, HELION, Gliwice, 2010

• Kshirasagar Nike, Priyadarshi Tripathy, "Software testing and Quality Assurance", Wiley, 2008, United States.

• Szyjewski Zdzisław, "Metodyki zarządzania projektami informatycznymi", Wydawnictwo PLACET, Warszawa, 2004.

• Kruchten Philippe, "The rational unified process an introduction", Parson Education Inc., Upper Saddle River, 2004.

• Sommerville I.: Software Engineering. Addison Wesley Longman, Harlow 2004.

• Alistair Cockburn, "Writing Effective Usecases", Addison-Wesley, c. 2001.

• Larman C.: Applying UML and patterns: An introduction to object-oriented analysis and design and the Unified Process. Prentice Hall, Upper Saddle River 2001.

• UML Specifications, About the Unified Modeling Language Specification Version 2.5.1 (omg.org), up-to-date version, 2021.

Efekty uczenia się:

1. Student potrafi rozwiązywać problemy z zakresu oprogramowania pracując w zespole - K1_U29 (ocena końcowa z projektu, oceny częściowe z artefaktów dotyczących ćw. 1-5).

2. Student posiada praktyczną wiedzę na temat specyfikowania, modelowania i projektowania oprogramowania K1A_W11, K1_U29 (ocena końcowa z projektu, oceny częściowe z artefaktów dotyczących ćw. 1-4).

3. Student posiada wiedzę z zakresu stosowania notacji UML (diagramy klas, diagramy przypadków użycia, diagramy komponentów) K1A_W11 (ocena końcowa z projektu, w tym oceny częściowe z artefaktów dotyczących ćw. 2-4).

4. Student posiada wiedzę na temat nowoczesnego oprogramowania CASE, jego funkcji i zastosowania - K1A_U21 (ocena końcowa z projektu, w tym oceny częściowe z artefaktów dotyczących ćw. 2-4).

5. Student potrafi opracować wymagania dotyczące oprogramowania na podstawie artefaktów tekstowych opisujących przykładowy system K1A_U21 (ocena końcowa z projektu, w tym ocena częściowa z artefaktów dotyczących ćw. 1).

6. Student potrafi używać nowoczesnego oprogramowania CASE i stosować je w praktyce na etapie formułowania wymagań, modelowania

przypadków użycia, modelowania struktury i zachowania - K1A_U21 (ocena końcowa z projektu, w tym oceny cząstkowe z artefaktów dotyczących ćw. 2-4).

7. Student posiada umiejętność weryfikacji i walidacji oprogramowania - K1A_U15, K1A_U19 (ocena końcowa z projektu, w tym ocena cząstkowa z ćw. 5 - ocena zaimplementowanej aplikacji).

Metody i kryteria oceniania:

Studenci realizują tematy projektowe (specyfikacja i implementacja przykładowego systemu). Tematy przydziela prowadzący.

Grupy dzielą się na dwie sekcje (automatyczny zapis do grupy na podst. hasła na PZE). Wewnątrz sekcji studenci dzielą się na zespoły. Każdy zespół realizuje jeden temat. Projekt składa się z pięciu części, każda z nich jest oceniana w skali 2-5 co 0,5. Końcowa ocena zaliczeniowa, to średnia arytmetyczna ocen cząstkowych zaokrąglona do dwóch cyfr po przecinku, przy czym X.25, to ocena równa X, a X.75, to ocena X+1.

Sylabus obowiązuje od zimowego semestru / roku akademickiego 2025/2026, a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Punkty przedmiotu w cyklach:

<bez przypisanego programu>			
Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	3	2021/2022-Z	