



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Inżynieria oprogramowania [S1Teleinf1>IO]

Przedmiot

Kierunek studiów

Teleinformatyka

Rok/Semestr

3/6

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

0

Inne

0

Ćwiczenia

0

Projekty/seminaria

30

Liczba punktów ECTS

5,00

Koordynatorzy

dr inż. Adam Wojciechowski

adam.wojciechowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Algorytmy i struktury danych I i II Języki programowania obiektowego

Cel przedmiotu

brak

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma ogólną wiedzę z zakresu inżynierii oprogramowania
 2. Ma podstawową wiedzę dotyczącą faz procesu produkcji oprogramowania
 3. Zna i rozumie zasady określania wymagań, modelowania systemów, projektowania oprogramowania, testowania i poprawiania wydajności oprogramowania, dokumentowania oprogramowania.
 4. Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu inżynierii oprogramowania.
 5. Ma wiedzę dotyczącą posługiwania się narzędziami CASE
 6. Ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w inżynierii oprogramowania
- Umiejętności:

1. określić i opisać wymagania wobec prostych systemów informatycznych
2. modelować systemy informatyczne (np. w notacji UML)
3. zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi
4. testować oprogramowanie i poprawiać jego wydajność
5. posługiwać się przynajmniej jednym z popularnych systemów zarządzania wersjami
6. dokumentować oprogramowanie

Kompetencje społeczne:

1. potrafi pracować w zespole i rozumie zasady współpracy społecznej

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

a) w zakresie wykładów:

- Egzamin pisemny

lub

- Ocena wiedzy i umiejętności oraz innowacyjności zawartych w pracach realizowanych w ramach laboratoriów i w formie pracy domowej

b) w zakresie ćwiczeń projektowych:

- Odbiór projektu na podstawie prezentacji członków zespołu projektowego

Treści programowe

Program przedmiotu obejmuje następujące zagadnienia:

- Tradycyjne modele cyklu życia i podstawowe rodzaje metodyk wytwarzania oprogramowania
- Faza określenia wymagań i standard IEEE 830.
- Analiza/modelowanie systemów z wykorzystaniem języka UML
- Projektowanie i implementacja oprogramowania
- Niezawodność oprogramowania – unikanie błędów, odporność na błędy, testowanie
- Ewolucja oprogramowania, modele komercjalizacji oprogramowania
- Narzędzia inżynierii oprogramowania. Zarządzanie zmianami i konfiguracją
- Zarządzanie projektem informatycznym
- Metodyki tworzenia oprogramowania
- Zarządzanie ryzykiem w projektach informatycznych
- Czynniki innowacji w budowie oprogramowania.

Program zajęć laboratoryjnych obejmuje wykonanie przez studentów projektu, w ramach którego praktycznie wykorzystają następujące zagadnienia:

- Modelowanie wymagań (opis problemu, specyfikacja wymagań funkcjonalnych i pozafunkcjonalnych)
- Modelowanie systemów w języku UML
- Projektowanie oprogramowania
- Implementacja kodu na podstawie diagramów UML
- Testowanie oprogramowania, tworzenie testów jednostkowych
- Dokumentacja kodu
- Narzędzia zarządzania wersjami: svn, git

Tematyka zajęć

brak

Metody dydaktyczne

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, studium przypadków,

Literatura

Podstawowa:

1. A. Jaskiewicz, Inżynieria oprogramowania, Helion, 1997.
2. G. Booch, J. Rumbaugh, I. Jacobson, UML przewodnik użytkownika, WNT, 2000.
3. M. Fowler, K. Scott, UML w kropelce, Oficyna Wydawnicza LTP, 2002

4. Sommerville Ian Inżynieria oprogramowania, WNT, 2003.
5. UML. Inżynieria oprogramowania. Wydanie II UML. Inżynieria oprogramowania. Wydanie II, P. Stevens, Helion, Gliwice, 2007
6. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Wzorce projektowe. Elementy oprogramowania obiektowego wie-lokrotnego użytku, WNT, 2008
7. Sacha, Inżynieria oprogramowania, PWN, 2010.

Uzupełniająca:

1. S. Maguire, Niezawodność oprogramowania, Helion, 2002
2. J. W. Cooper, Java. Wzorce projektowe, Helion, 2001
3. R. S. Pressman, Praktyczne podejście do inżynierii oprogramowania, WNT, 2004.
4. J. Warmer, A. Kleppe, Inżynieria oprogramowania OCL precyzyjne modelowanie w UML, WNT, 2003.
5. W. B. Kernighan, R. Pike, Inżynieria oprogramowania. Lekcja oprogramowania, WNT, 2002.
6. R.C. Martin, M. Martin, Agile, programowanie zwinne, Helion, 2008.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	120	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	64	3,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	56	2,00