**Nazwa przedmiotu:**

Inżynieria systemów informatycznych

**Koordynator przedmiotu:**

Dr inż. Krzysztof Kaczmarski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Inżynieria i Analiza Danych

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-DS000-ISP-0232

**Semestr nominalny:**

3 / rok ak. 2019/2020

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. godziny kontaktowe – 48h; w tym
a) obecność na wykładach – 30h
b) obecność na ćwiczeniach – 15h
c) konsultacje – 3h
2. praca własna studenta – 55h; w tym
a) zapoznanie się z literaturą – 10h
b) przygotowanie do zajęć ćwiczeniowych – 15h
c) przygotowanie do testów, rozwiązanie samodzielne zadań – 10h
d) napisanie projektu – 20 h
Razem 103 h, co odpowiada 4 pkt. ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1. obecność na wykładach – 30h
2. obecność na ćwiczeniach –15h
3. konsultacje – 3 h
Razem 48h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. napisanie projektu – 20 h
Razem 20h, co odpowiada 1 pkt. ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 15h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Architektura systemów informatycznych, Projekt indywidualny, Systemy operacyjne w inżynierii danych

**Limit liczby studentów:**

.

**Cel przedmiotu:**

Celem przedmiotu jest przekazanie wiedzy dotyczącej podstawowych zasad i technik inżynierii oprogramowania oraz wykształcenie umiejętności tworzenia prostych modeli systemów informatycznych.

**Treści kształcenia:**

Wykład:
Pojęcia wstępne: programowanie jako proces inżynieryjny, podstawowe potrzeby prowadzące do konieczności wykorzystania inżynierii oprogramowania, wprowadzenie pojęcia pracy grupowej i skalowalności projektu
Pojęcia podstawowe obiektowego programowania: klasy, metody, dziedziczenie, polimorfizm, przeciążanie, uogólnianie, uszczegóławianie, przykłady prostych schematów UML
Pojęcia zaawansowane obiektowego programowania: meta-klasy, wątki, niuanse schematów UML
Modele rozwoju oprogramowania: kaskadowy, spiralny, piramida, XP, Open-Source. Wymagania stawiane przed twórcą oprogramowania. Wymagania stawiane przed produktem końcowym.
Omówienie faz rozwoju projektu: planowanie, analiza (szacowanie złożoności oprogramowania), projektowanie (przygotowywanie pewnych dokumentów specyfikacji), implementowanie (wybór języka, zagadnienia dodatkowe), dokumentowanie (tworzenie dokumentacji), testowanie (automatyka i pomocne metody), instalowanie (przygotowanie pakietów dla klienta), konserwacja (przygotowywanie aplikacji do zmian i poprawek).
Zagadnienia pracy grupowej: obieg dokumentów, standaryzacja procesu wytwórczego, motywowanie pracowników.
Ćwiczenia:
Ćwiczenia obejmują dyskusje związane z modelowaniem w UML. Szacowanie pracochłonności zadania, planowania przedsięwzięcia informatycznego.

**Metody oceny:**

W czasie semestru odbywa się 14 godzinnych ćwiczeń obejmujących tworzenie modeli UML oraz modelowanie większych systemów. Na ostatnich zajęciach odbywa się kolokwium. Dodatkowo grupy studentów po cztery osoby przygotowują projekt zadanego systemu informatycznego. Końcowa ocena składa się w 50% z oceny z testu i 50% z oceny z wykonanej dokumentacji projektowej.

**Egzamin:**

nie

**Literatura:**

1. I. Sommerville Inżynieria Oprogramowania, 2003
2. G. Booch. Object-oriented analysis and design with applications., 1994.
3. F. P. Brooks. Mityczny osobomiesiąc. Wydawnictwa Naukowo Techniczne, 2000.
4. S. D. Conte, H. E. Dunsmore, V. Y. Shen. Software Engineering Metrics and Models. 1986.
5. R. Dumnicki, A. Kasprzyk, M. Kozłowski. Analiza i projektowanie obiektowe. Helion, 1998.
6. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
7. J. Górski. Inżynieria Oprogramowania w projekcie informatycznym. Mikom, 1999.
8. C. S. Horstmann. Mastering object-oriented design in C++. John Wiley, 1995.
9. Jaszkiewicz. Inżynieria oprogramowania. Helion, 1997.
10. W. C. Lim. Managing software reuse. Prentice-Hall, 1999.
11. J. Martin, J. J. Odell. Podstawy Metod Obiektowych. WNT, 1997.
12. R. J. Muller. Bazy Danych język UML w modelowaniu danych. Mikom, 2000.
13. J. Robertson, S. Robertson. Pełna analiza systemowa. WNT, 1999.
14. J. Rumbaugh, M. Blaha, W. Premerlani, W. Lorensen. Object-Oriented Modelling and Design. 1991.
15. K. Subieta. Obiektowość w projektowaniu i bazach danych. Akadem. Oficyna Wyd. PLJ, 1998.

**Witryna www przedmiotu:**

.

**Uwagi:**

.

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

Zna język UML i sposoby stosowania go w praktyce.

Weryfikacja:

Test pisemny na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_W14, DS\_W15

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka W02:**

Zna modele rozwoju oprogramowania, w tym modelu kaskadowego, spiralnego, odkrywczego wraz z poszczególnymi fazami oraz niekonwencjonalne metody wytwarzania oprogramowania: open-source, scrum, itd.

Weryfikacja:

Test pisemny na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_W15

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka W03:**

Posiada podstawową wiedzę dotyczącą środowisk wytwórczych, pracy w zespole oraz narzędzi do pracy zespołowej

Weryfikacja:

Test pisemny na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_W15

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi stworzyć model systemu w języku UML obejmujący wymagania użytkownika oraz projekt rozwiązania

Weryfikacja:

Test pisemny na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_U10, DS\_U21, DS\_U23, DS\_U26, DS\_U27

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UK, I.P6S\_UO

**Charakterystyka U02:**

Potrafi zaprojektować prosty system informatyczny

Weryfikacja:

Projekt oceniony na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_U23, DS\_U26, DS\_U27, DS\_U10, DS\_U21

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UO, I.P6S\_UK

**Charakterystyka U03:**

Potrafi zastosować wybraną metodę oszacowania pracochłonności zadania

Weryfikacja:

Test pisemny na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_U21, DS\_U23

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UK, I.P6S\_UO

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

Rozumie zagadnienia związane z pracą grupową

Weryfikacja:

Projekt oceniony na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_K02, DS\_K03, DS\_K04

**Powiązane charakterystyki obszarowe:** I.P6S\_KO, I.P6S\_KR

**Charakterystyka K02:**

Rozumie zasady negocjowania z klientem oraz prowadzenia wywiadu związanego z określeniem wymagań użytkownika

Weryfikacja:

Projekt oceniony na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_K03

**Powiązane charakterystyki obszarowe:** I.P6S\_KO

**Charakterystyka K03:**

Zna zagadnienia związane z jakością produktów informatycznych oraz konsekwencje szybkiego rozwoju nowych technologii w informatyce.

Weryfikacja:

Projekt oceniony na końcu zajęć

**Powiązane charakterystyki kierunkowe:** DS\_K01, DS\_K03

**Powiązane charakterystyki obszarowe:** I.P6S\_KK, I.P6S\_KO