**Nazwa przedmiotu:**

Inżynieria oprogramowania 1

**Koordynator przedmiotu:**

Dr inż. Krzysztof Kaczmarski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-IN000-ISP-0353

**Semestr nominalny:**

5 / rok ak. 2017/2018

**Liczba punktów ECTS:**

4

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

1. godziny kontaktowe – 45h; w tym
 a) obecność na wykładach – 30h
 b) obecność na ćwiczeniach – 15h
2. praca własna studenta – 55 h; 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, konsultacja – 20 h
Razem 100 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
Razem 45h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

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

1. obecność na ćwiczeniach – 15 h
2. przygotowanie do ćwiczeń – 15 h
3. przygotowanie do testów, rozwiązanie samodzielne zadań – 10h
4. napisanie projektu, konsultacja – 20 h
Razem 60h, co odpowiada 2 pkt. ECTS

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

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

**Wymagania wstępne:**

Programowanie 3 – zaawansowane, Programowanie 2 – obiektowe
Bazy danych
Systemy operacyjne

**Limit liczby studentów:**

Ćwiczenia – 30 os/grupa

**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, podsta-wowe potrzeby prowadzące do konieczności wykorzystania inżynierii oprogramowania, wprowadzenie pojęcia pracy grupowej i skalo-walności projektu
Pojęcia podstawowe obiektowego programowania: klasy, metody, dziedziczenie, polimorfizm, przeciążanie, uogólnianie, uszczegóła-wianie, 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 7 dwu godzinnych ćwiczeń. Podczas ćwiczeń przeprowadzane są 4 testy. Na końcu zajęć grupy studentów po cztery osoby przygotowują projekt zadanego systemu informa-tycznego. Końcowa ocena składa się w 50% z ocen z testów 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: Ele-ments 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:**

e.mini.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Zna język UML i sposoby stosowania go w praktyce.

Weryfikacja:

4 testy

**Powiązane efekty kierunkowe:** K\_W12

**Powiązane efekty obszarowe:** T1A\_W07

**Efekt W02:**

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

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane efekty kierunkowe:** K\_W07, K\_W09, K\_W12, K\_W14, K\_W15

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W06, T1A\_W07, T1A\_W08, T1A\_W08, T1A\_W11

**Efekt W03:**

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

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane efekty kierunkowe:** K\_W11, K\_W12

**Powiązane efekty obszarowe:** T1A\_W07, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

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

Weryfikacja:

4 testy

**Powiązane efekty kierunkowe:** K\_U01, K\_U10, K\_U13, K\_U28

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U10, T1A\_U16, T1A\_U14

**Efekt U02:**

Potrafi zaprojektować prosty system informatyczny

Weryfikacja:

Projekt

**Powiązane efekty kierunkowe:** K\_U23, K\_U26, K\_U28, K\_U30

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U16, T1A\_U12, T1A\_U14, T1A\_U16

**Efekt U03:**

Potrafi zastosować wybraną metodę oszacowania pracochłonności zadania

Weryfikacja:

ocena aktywności studenta

**Powiązane efekty kierunkowe:** K\_U26

**Powiązane efekty obszarowe:** T1A\_U12

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Rozumie zagadnienia związane z pracą grupową

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane efekty kierunkowe:** K\_K05

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04

**Efekt K02:**

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

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K02, T1A\_K05

**Efekt K03:**

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

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane efekty kierunkowe:** K\_K01, K\_K03, K\_K04

**Powiązane efekty obszarowe:** T1A\_K01, T1A\_K02, T1A\_K02, T1A\_K05